Mathematical Statistics
Stockholm University

Towards personalized treatment of cancer

Tomas Jacobsson

Examensarbete 2012:3

Postal address:
Mathematical Statistics
Dept. of Mathematics
Stockholm University
SE-106 91 Stockholm
Sweden

Internet:
http://www.math.su.se/matstat

Mathematical Statistics
Stockholm University
Examensarbete **2012:3**,
http://www.math.su.se/matstat

Towards personalized treatment of cancer

Tomas Jacobsson*

May 2012

Abstract

Every year about 55,000 people are diagnosed with cancer and about 20,000 people die from the disease in Sweden. Statistically, one of three Swedes will suffer from cancer at some time in their life. The significance of reducing cancer mortality can hardly be overstated. Most patients can be cured by cancer surgery and radiation of the primary tumor if no metastasis have occurred. It is therefore very important to reduce the tumor spreading at an early stage. To reduce the tumor spreading, chemotherapy is normally used to inhibit the production of new cancer cells.

During the last decade it has been realized that cancer is a heterogeneous disease. This suggests that we need better methods to match molecular tumor characteristics with an optimal drug combination for each patient. To study the potential of developing such methods, data from cancer cell lines was used to test two cases: Case 1) To optimize the treatment based on the patient's molecular tumor profile. Case 2) To develop accurate prediction models for drug screening to help make cancer drug discovery more efficient.

By using different techniques within the field of chemometrics (the intersection of chemistry and statistics) it was possible to integrate gene expression data (describing the characteristics of a cancer tumor) and chemical data (describing the properties of a chemical compound) to predict the concentration level needed for a chemical compound to inhibit the cell line growth with 50%. Such a prediction model is of direct use in Case 1 and Case 2, where it can be used for predicting the optimal drug treatment based on the patient's molecular tumor profile and for predicting the effect of a new drug candidate, respectively.

In both cases, the best model for predicting the concentration level needed for a chemical compound to inhibit the cell line growth with 50% was achieved with random forest. The variables describing the chemical compounds were of high importance when predictions were made. Most importantly, it was also found that the gene expression data, describing the cancer cell line, adds significant information, indicating that cancer treatment should be personalized.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: jacobsson_tomas.hotmail.com. Supervisor: Jan-Olov Persson.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Martin Eklund at the department of Medical Epidemiology and Biostatistics at Karolinska Institutet. Thank you, for formalizing this project, for contributing with your wide knowledge, programming skills, inspiration and fruitful discussions.

I would also like to thank Professor Anders Jacobsson, my father, for valuable comments on the introduction to molecular biology section and my supervisor at Stockholm University, Jan-Olov Persson for valuable comments and discussions.

Contents

1 Introduction

1.1 Background

Every year about 55,000 people are diagnosed with cancer and about 20,000 people die from the disease in Sweden alone. Statistically one of three Swedes will suffer from cancer at some time. There are around 200 different classified types of cancer. For men the most common cancer type is prostate cancer with 10,000 diagnoses per year in Sweden. For women the most common cancer type is breast cancer with 7000 diagnoses per year in Sweden (Cancerfonden, 2012).

Depending on which organ/organs the cancer developed, how far the cancer has spread and age and health of the patient the treatment options are surgery, radiotherapy, hormone treatment, and chemotherapy.

The cause of cancer is divided into two groups, environmental cause, which is estimated to cause 90-95% of the cases and those with a hereditary genetic cause which is estimated to cause 5-10% of the cases. Common environmental factors that causes cancer are poor diet (30-35%), smoking (25-30%), infection (15-20%) and radiation and stress, etc. (remaining percentage) (Anand et al., 2008).

Regardless the cause of cancer, all types of cancer appears by genetic changes in the DNA. Many of the changed genes are involved with the reparation of the DNA. During cell division the DNA is replicated in the cell and when something goes wrong repairing factors (proteins) correct the error. When a repairing gene stops working the cell can undergo uncontrolled growth and division, which may destroy the surrounding tissue. After multiple divisions the cancer cells turn into a small lump called a tumor. Mutations lead to tumor growth and eventually the tumor will break through the basal membrane (a sheet of thin tissue that forms the border with the underlying connective tissue). Some cancer cells take the opportunity to circulate through the bloodstream and spread to other organs in the body, called metastasis. Metastasis are in fact the reason to almost 90% of cancer-related deaths (Hejmadi, 2010). Most patients can be cured by cancer surgery and radiation of the primary tumor if no metastasis occur. It is therefore very important to reduce the tumor spreading at an early stage, which is done with chemotherapy. Chemotherapy is a drug therapy that is used to inhibit the production of the cancer cells by killing the cells that divide rapidly. Unfortunately chemotherapy may also harm the healthy cells that divide rapidly which can lead to side effects like decreased production of blood cells, hair loss etc. Therefore the level of concentration that is given to a patient have to be restricted, since at some concentration level everything will eventually die.

Under the last two decades the revolution of genomics has lead to an enormous accumulation of biological data from gene sequencing and other

techniques. We can generate thousands, sometimes millions of measurements on a single individual.

Although there are about 200 different classified cancer types it has under the last decade been realized that most cancer are heterogeneous diseases. This indicates that we need to abandoning the traditional way of classifying cancer tumors into discrete subtypes. In order to achieve the best results for a patient we need to provide personalized treatments that are tailored to an individual patient, based on his or her particular molecular tumor profile.

1.2 Long term aims

The long term aim for researchers at the department of Medical Epidemiology and Biostatistics (MEB) at Karolinska Institutet is to reduce the mortality of cancer. As a step to reduce the mortality of cancer we focus on two cases:

Case 1) **Personalized treatment optimization.** To reduce the mortality of cancer by personalizing the treatments. Based on the patient's molecular tumor profile the hope is to optimize the treatment of cancer.

Case 2) **Drug screening.** To make cancer therapy drug discovery faster and cheaper by developing accurate prediction models for drug candidates.

1.3 Aims for this project

This project will take an important step to fulfill these long term aims. To simulate these two cases, data from 60 human tumor cell lines, called NCI-60, is used. The data contains information about gene expressions, describing the characteristics of the cell lines. The concentration level needed to inhibit the tumor growth with 50% for 110 different chemical compounds has been measured by the US National Cancer Institute and is used as response data. To describe the chemical structure of a compound a vector with 14 variables is constructed.

The specific aims for this project are:

- to connect and integrate gene expression data, describing the characteristics of a cell line, and chemical data, describing the structure of a chemical compound.
- for the personalized treatment case and the drug screening case, respectively, construct statistical models for predicting the concentration level needed for a chemical compound to inhibit the cell line growth with 50%.
- to investigate if gene expression data and chemical data, respectively, are important for the prediction of the growth inhibition of cancer

cells. This gives an indication of the possibility to match molecular tumor characteristics to the optimal drug combination and predict how a promising drug candidate should perform by using a chemical description of the compound.

1.4 Main results

By using different techniques within the field of chemistry where multivariate mathematical methods are used to extract chemically relevant information, it was possible to connect and integrate the gene expression data (describing the characteristics of a cell line) and the chemical data (describing the properties of a chemical compound).

This report shows that the variables describing the chemical compounds, such as number of bonds and molecular weight etc., were of high importance when the concentration level needed for 50% growth inhibition was predicted. It was also found that the gene expression variables provide significant information for the prediction, indicating that cancer treatment should be personalized.

Methods as multiple linear regression, Section 3.3.3, lasso, Section 3.3.5.1, and random forest, Section 3.3.7, were used to construct models for predicting the concentration level needed for a chemical compound to inhibit the cell line growth with 50%. To minimize the prediction error of such a model variable selection methods as lasso and NetPath, Section 3.3.5.2, were used.

In both the personalized treatment case and the drug screening case, the random forest method generated the best prediction among the methods used. Also, by preselecting variables the computation time was significantly reduced.

1.5 Report structure

In order to simplify for the reader and to gain insight into basic molecular biology, the freestanding Section 6.1 in the Appendix provide an introduction to this field. In Section 2, the two cases, Case 1: Personalized treatment optimization and Case 2: Drug screening, are explained. In Section 3, the data is introduced together with statistical methods and software used. How the description of a disease state is connected and integrated with a description of a compound is also presented in this section. The results are represented in Section 4 followed by a discussion section, Section 5.

2 The two cases: Personalized treatment optimization and Drug screening

The long term aim is to reduce the mortality of cancer. As a step to reduce the mortality of cancer we focus on two cases:

Case 1) **Personalized treatment optimization.** To reduce the mortality of cancer by personalizing the treatments. Based on the patient's molecular tumor profile the hope is to optimize the treatment of cancer.

Case 2) **Drug screening.** To make cancer therapy drug discovery faster and cheaper by developing accurate prediction models for drug candidates.

The two cases can of course also relate to each other. For example, to optimize the personal treatment we may have to identify a new drug.

To simulate these two cases and construct prediction models, data from 60 human tumor cell lines, called NCI-60, is used. The data contain information about gene expressions, describing the characteristics of the cell lines. The concentration needed for 110 different chemical compounds to inhibit the tumor growth with 50% have been measured by the US National Cancer Institute and is used as response data. The data is further explained in Section 3.1.

2.1 Case 1: Personalized treatment optimization

In the personalized treatment case, the aim is to optimize the cancer drug treatment, based on the patient's tumor profile. The optimal drug treatment may be thought of as the concentration level of a chemical compound that inhibit the tumor growth the most. However, at some high concentration level everything will eventually die and a low concentration level is preferable. Therefore, the optimal compound is interpreted as the compound that needs the lowest concentration level to inhibit the cell line growth with 50%.

In a real situation, in order to select the optimal drug treatment for a patient's cancer tumor we have no information about the effect for any drugs tested on the patient's particular tumor. For this reason, data is not used from the cell line we want to predict the growth inhibition for. Thus, before a model is constructed, the data is divided by cell lines.

2.2 Case 2: Drug screening

In the drug screening case, the aim is to predict the effect of a promising drug candidate. How a drug affects a tumor is simulated with the 110 chemical compounds, by predicting the concentration level needed for a chemical compound to inhibit the cell line growth with 50%. A low concentration value needed for 50% growth inhibition indicate a promising drug candidate.

In a real situation, we have no information about the effect of a new drug. For this reason, data is not used from the chemical compound we want to predict the concentration level needed for 50% growth inhibition. Thus, before a model is constructed, the data is divided by chemical compounds.

3 Materials and methods

3.1 Data

In this project data from 60 human tumor cell lines is used, called the NCI-60 panel, each containing information about gene expression and growth inhibition of 110 different chemical compounds used. The NCI-60 panel contain the most extensively characterized human cell lines in broad laboratory use and have been used by the US National Cancer Institute to screen over 100000 compounds to receive information about their effect on the respective cell lines.

The cell lines were obtained from cancer tissue and are grown in plastic bottles or plastic cups containing culture medium mimicking a "normal" environment of the organism. The NCI-60 cell lines are used in this project because they have the advantage of having abundant publicly available data. Another advantage is the possibility of rapid testing new chemical compounds at a relatively low cost without ethical problems. The specific data of the NCI-60 panel and the drug screening is explained in detail below.

3.1.1 NCI-60

Between 1985-1990, the US National Cancer Institute (NCI) developed an *in vitro* (Latin: within glass) primary screen based on a panel of 60 human tumor cell lines, representing cells from:

- Leukemia (blod) (6)
- Lung (9)
- Colon (bowel) (7)
- CNS (central nervous system) (6)
- Melanoma (skin) (10)
- Ovarian (7)
- Renal (kidney) (8)
- Prostate (2)
- Breast (5)

In parentheses is the number of cell lines for each cancer subtype. As biological properties of the cell lines the mRNA expression data from the Affymetrix Human Gene U133A chip is used. Affymetrix Human Gene U133A chip is a microarray that was used by NCI to measure the production of mRNA of thousands of genes simultaneously during a specific

condition. Microarrays are further explained in Section 6.1. For each cell line we have the mRNA value of 22283 probes (fragment of a gene), representing 13032 unique genes (Shankavaram et al., 2007). A higher mRNA level indicates a more active gene, which in a disease state might indicate that the gene could serve as a drug target. The NCI-60 data are publicly available at `http://dtp.cancer.gov/mtargets/mt_index.html`

3.1.2 Chemical compound data

More than 100000 chemical compounds have been tested against the 60 cell lines with a defined range of concentrations to determine the growth inhibition. In this analysis a panel of 110 molecules are used, called A118. The A118 are used because they have known mechanism of action and have been experimentally tested at least 4 times on the NCI-60 cell lines (Bussey et al., 2006). To be able to use mathematical methods the chemical compounds has to be described numerically. How this was done is explained in Section 3.2.1. The A118 data set was downloaded from `http://discover.nci.nih.gov/cellminer/`.

3.1.3 GI_{50}

After 48 hours of drug treatment the growth inhibition for each compound tested on each cell line has been determined by the US National Cancer Institute (Boyd and Paull, 1995). For each chemical compound the concentration that caused 50% growth inhibition (GI_{50}) in the unit M (mol/l) was determined for each cell line. GI_{50} values was obtained between 10^{-11} to 1 M, where a low concentration value is preferable and indicates higher efficacy of the chemical compound. Chemotherapy will also harm the healthy cells that divide rapidly. Therefore the level of concentration that are given to a patient have to be restricted, since at some concentration level everything will eventually die.

The response variable, $y = -log_{10}(GI_{50})$ is chosen by NCI. Since

$$GI_{50} = 10^{-y} \Leftrightarrow -log_{10}(GI_{50}) = y, \tag{1}$$

y obtain values from 0 to 11. As mentioned, a low concentration value to inhibit the growth with 50% is preferable, which results in a high value on the response variable y.

3.1.4 Missing or incomplete data

In statistical modeling and inference a common problem is how to handle missing data. By using probability models, one approach, called imputing, is to fill in the missing data. A problem with this imputation is that the missing values will be treated as they were known. Another approach is to

simply discard the observations that are incomplete, this however can lead to reduced power and selection bias.

In our data we have some mRNA values missing for one lung cancer cell line (LC.NCI_H23) and we have missing values for some properties for one of the 110 chemical compounds (609395). In this case we discarded the cell line and the chemical compound with the missing values. This can be considered as we only had 59 cell lines and 109 chemical compounds from the beginning. Thus, the outcome of this is that we now have 109 chemical compounds that have been tested on 59 cancer cell lines, which gives us 6431 observations in total.

In the gene expression data we are missing the IDs (names) of 1214 genes, which we have been named NA (Not Available) and the number of its position within the data set (ex: NA22238, for a missing gene ID on the gene positioned at 22238 in the data).

3.2 Chemometrics - chemistry meets statistics

To improve the understanding of chemical information, mathematical, statistical, graphical or symbolic methods are used. This area is called chemometrics. One of the founders, Svante Wold explains chemometrics as:

"How to get chemically relevant information out of measured chemical data, how to represent and display this information, and how to get such information into data" (Wikberg et al., 2010).

A major issue within this field is how to connect and integrate a biological and chemical description of a disease state with the chemical description of a compound.In this project, one aim was to describe the 109 chemical compounds and the 59 cell lines numerically. How to solve these issues are explained below.

3.2.1 Descriptors

As the name indicates, descriptors describe a molecule's structure. By studying the chemical structures the expected properties can be described. Descriptors vary in complexity and dimensionality. Bioclipse is a free and open source workbench, an integration platform for chemo- and bioinformatics (Spjuth et al., 2009; bioclipse.net, 2007). We constructed our descriptors in Bioclipse with the Chemistry Development Kit (CDK), which is an open source Java based library for structural chemo- and bioinformatics (Steinbeck et al., 2006; cdk.sf.net, 2003). We selected a 2D description of the chemical structure, which can be explained as what we can write on a piece of paper. We obtained a descriptor vector with 14 variables for each chemical compound, e.g. number of carbons (nC), molecular weight (MW) and

number of bonds (nB). These descriptors describe the molecules structure in a numerical representation of the 109 chemical compounds.

3.2.2 Quantitative structure-activity relationship (QSAR)

By using the chemical descriptors that describe a compound we would like to predict its effect on a cell line. Quantitative structure-activity relationship (QSAR) is a method that use numerical properties of a compound (descriptors) to form a mathematical relationship with a biological activity (Wikberg et al., 2010). One of the first historical QSAR application was to predict boiling points. In our case the biological activity is $-log_{10}$ of the concentration of a substance required to give a 50% growth inhibition. We can express the QSAR model as a regression model:

$$y_i = f(d_{C,i}), \qquad i = 1, ..., 109 \tag{2}$$

where f is an unknown function and $d_{C,i}$ is a vector with chemical descriptions for compound i. The aim is to find an empirical equation that can predict the biological activity for other chemical compounds. However, since we also have biological properties about the tumor cell lines and QSAR models only deals with the chemical space, we wanted to use a technology that also can merge tumor biology with chemistry.

3.2.3 Proteochemometrics (PCM)

As the development within genomics (the study of the genome) "exploded", a technology called proteochemometrics (PCM) was born. Proteochemometrics connects and integrates biological and chemical data to construct mathematical models for prediction of properties of chemical compounds. Proteochemometrics is a generalization of QSAR that includes multiple protein targets and was developed by Wikberg et al. in 2001 (Wikberg et al., 2001).

In the same way as we describe a compound using chemical descriptors we need to express the properties of a cancer tumor. To get a numerical representation capturing the biological properties of the cancer cell lines the gene expression data (mRNA levels) was used. mRNA is explained in the introduction to molecular biology section, Section 6.1, and can be interpreted as the activity of a gene.

The numerical quantifications permits mathematical treatment, thereby merging tumor biology with the chemistry to predict treatment efficacy. Given data from the tumor biological space and data from the therapy chemical space, we can expand equation (2) with proteochemometrics:

$$y_{i,j} = f(d_{C,i}, d_{T,j}) \qquad i = 1, ..., 109, j = 1, ..., 59. \tag{3}$$

In our case $y_{i,j} = -log_{10}(GI_{50i,j})$ is the treatment efficacy of chemical compound i and tumor cell line j, $d_{C,i}$ is a chemical description (a vector with 14 variables) of compound i and $d_{T,j}$ is a biological description (a vector with 22283 variables) of tumor cell line j. f is an unknown function that we want to estimate by using the observed growth inhibition for each compound tested on each cell line.

3.2.4 Merged data

Our aim is to predict the concentration level needed of a chemical compound to inhibit the growth with 50%, given a molecular profile of a cell line and a numerical description of a chemical compound. By using proteochemometrics that merge tumor biology with chemistry we can predict the $-log_{10}(GI_{50})$ for each chemical compound tested on each cell line. The data was merged as in (4) and the aim is now to deduce a function, f, in equation (3) from the $n \times (p+1)$ data matrix:

$$(\mathbf{X}, \mathbf{y}) = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} & y_1 \\ x_{21} & x_{22} & \dots & x_{2p} & y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} & y_n \end{bmatrix} =$$

$$= \begin{bmatrix} \overbrace{x_{1,1}}^{geneexpressions} & x_{1,2} & \dots & x_{1,22283} & \overbrace{x_{1,22284}}^{descriptors} & \dots & x_{1,22297} & \overbrace{y_1}^{response} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{59,1} & x_{59,2} & \dots & x_{59,22283} & x_{59,22284} & \dots & x_{59,22297} & y_{59} \\ x_{60,1} & x_{60,2} & \dots & x_{60,22283} & x_{60,22284} & \dots & x_{60,22297} & y_{60} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{6431,1} & x_{6431,2} & \dots & x_{6431,22283} & x_{6431,22284} & \dots & x_{6431,22297} & y_{6431} \end{bmatrix} =$$

$$= \begin{bmatrix} \overbrace{9.52}^{geneexpressions} & 7.67 & \dots & 3.67 & \overbrace{160}^{descriptors} & \dots & 55.98 & \overbrace{7.35}^{response} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 6.98 & 7.65 & \dots & 2.97 & 160 & \dots & 55.98 & 4.67 \\ 9.52 & 7.67 & \dots & 3.67 & 196 & \dots & 126.48 & 6.82 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 6.98 & 7.65 & \dots & 2.97 & 228 & \dots & 107.53 & 4.34 \end{bmatrix}, \quad (4)$$

where n is the number of observations and p is the number of independent variables. The data was ordered so that each row in (4) represents one observation. For each observation, the first 22283 columns are gene expression data (mRNA values) for the specific cancer cell line. The following 14 columns (22284 to 22297) are a description of the chemical compound that has been tested on the cell line. The last column is the response, $-log_{10}$

of the concentration level needed for the chemical compound to inhibit the growth of the cancer cell spread with 50%.

The data have been merged in such a way that the first 59 rows are the observations from the first chemical compound tested on the 59 cell lines. The next 59 rows (60 to 118) are the observations from the second chemical compound tested on the 59 cell lines, and so on.

3.3 Statistical methods

To fit a function, f, a standard procedure is to fit a linear regression model to the data (Section 3.3.3). In real life however, effects are often not linear. Working with genetic data and chemical data, we often have a large number of independent variables, which are often very strongly correlated. When we have a much larger number of independent variables p than number of observations N, as in this case $p = 22297$ and $N = 6431$, a problem is often high variance and overfitting (Section 3.3.2). To reduce the number of independent variables methods as Lasso and NetPath (Section 3.3.5) were used. With strongly correlated independent variables it is difficult to attribute changes in the dependent variable to one of the independent variables rather than another. When independent variables are strongly correlated, linear regression can perform poorly because of high variance.

An algorithm which tends to work better than linear methods in these situations are Regression Trees (Section 3.3.6.1). The main idea behind regression trees is to use data to recursively partition the sample space into smaller and smaller regions. To construct a statistical model, besides using linear regression methods, a method called random forest was used. The Random Forest method (Section 3.3.7) uses unpruned trees with a randomized selection of independent variables at each split to reduce the correlation between trees, which in turn reduce the variance when trees are correlated (Section 3.3.6.3). Before the random forest method is introduced, the methods used in the random forest need to be introduced (Section 3.3.6).

3.3.1 Loss function

To determine and compare how well the predicted function $\hat{f}(X)$ fits the available data a "loss function" $L(Y, \hat{f}(X))$ is introduced. The most convenient and commonly used loss function measures the squared error between Y and the predicted function $\hat{f}(X)$,

$$L(Y, \hat{f}(X)) = (Y - \hat{f}(X))^2. \tag{5}$$

3.3.2 Model complexity bias-variance tradeoff

Before we compare and select our models there are some issues that we need to consider. Assume that we observe our data matrix $(\mathbf{X}, \mathbf{y})$ from a

statistical model:

$$Y = f(X) + \varepsilon, \tag{6}$$

where $E[\varepsilon] = 0$ and $Var(\varepsilon) = \sigma_\varepsilon^2$. With the input vector X we want to develop a function $\hat{f}(X)$ which we can use for future predictions of Y.

One natural way to estimate the prediction error is the average loss over the data sample used to construct the model, called the training error (R_t),

$$R_t = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(\mathbf{x}_i)) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{7}$$

when the loss function is the squared error and $\mathbf{x}_i = (x_{i,1}, ..., x_{i,p})^T$. Unfortunately the training error (7) is not a good estimate because we can always make the training error arbitrarily small by selecting our model $\hat{f}(X)$ complex enough (Figure 1). If we select a very complex model, the model fitted to the data has adapted to the random noise in the data and will predict very poorly to new observations.

The expected prediction error, also called test or generalization error (R_g), at $X = \mathbf{x}$ where $\mathbf{x}$ is the observed vector, can under squared error loss be expressed as

$$\begin{aligned}
R_g(\mathbf{x}) &= E[(Y - \hat{f}(\mathbf{x}))^2 \mid X = \mathbf{x}] \\
&= E[(\hat{f}(\mathbf{x}) - E[\hat{f}(\mathbf{x})] + E[\hat{f}(\mathbf{x})] - Y)^2 \mid X = \mathbf{x}] \\
&= \sigma_\varepsilon^2 + E[(\hat{f}(\mathbf{x}) - E[\hat{f}(\mathbf{x})])^2 + 2(\hat{f}(\mathbf{x}) - E[\hat{f}(\mathbf{x})])(E[\hat{f}(\mathbf{x})] - f(\mathbf{x})) \\
&\qquad + (E[\hat{f}(\mathbf{x})] - f(\mathbf{x}))^2] \\
&= \sigma_\varepsilon^2 + E[\hat{f}(\mathbf{x}) - E[\hat{f}(\mathbf{x})]]^2 + (E[\hat{f}(\mathbf{x})] - f(\mathbf{x}))^2 \\
&= \sigma_\varepsilon^2 + Var(\hat{f}(\mathbf{x})) + bias^2(\hat{f}(\mathbf{x})).
\end{aligned} \tag{8}$$

The first term σ_ε^2 is the irreducible error, the variance of the error term ε, and can not be avoided no matter how well we estimate $f(\mathbf{x})$. The second term is the variance, the expected squared deviation of $\hat{f}(\mathbf{x})$ around its mean. The last term is the squared bias, the amount by which the average of our estimate differs from the true mean (Hastie et al., 2009).

Generally, the more complex we select the model $\hat{f}(X)$, the more able it is to adapt to a complex relationship between X and Y, and the lower the bias but the higher the variance. Since, too much fitting will adapt the model to closely to the data used to fit the model and will not generalize well, i.e. have large test error. If the model is not complex enough, it will underfit and may have large bias. Thus, in between there is an optimal model $\hat{f}(\mathbf{x})$ that best balances the bias and the variance and gives a minimal test error (Hastie et al., 2009). This is the model we want to select for future prediction. The bias-variance tradeoff is described graphically in Figure 1.

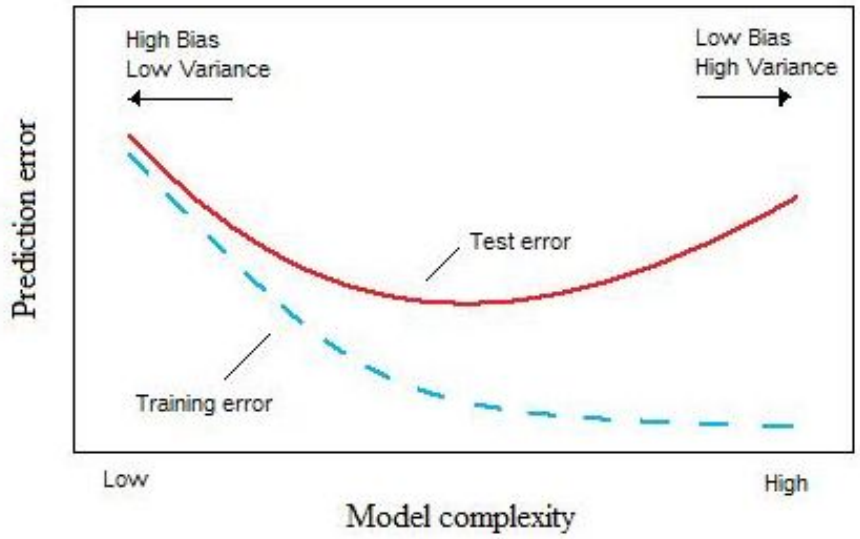

Figure 1: *The bias-variance tradeoff. Test error and training error as a function of model complexity. The training error is decreasing when the model complexity is increasing, whereas the test error has a minimum because of the bias-variance tradeoff (figure adapted from Elements of Statistical Learnings, p.38 (Hastie et al., 2009)).*

3.3.3 Linear regression models

To find a function $f(X)$ a standard procedure is to fit a linear regression model of the form

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j. \tag{9}$$

This model assumes that the regression function $E(Y \mid X)$ is linear or can be approximated by a linear function.

To estimate the beta parameters the most common method is *least squares*, where the coefficients $\beta = (\beta_0, \beta_1, ..., \beta_p)^T$ are chosen to minimize the residual sum of squares (RSS):

$$RSS(\beta) = \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2. \tag{10}$$

The β estimates can be written as

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{11}$$

and are called *ordinary least squares*. However, the parameter vector β can be difficult to compute if $p > N$, since the ordinary least squares solution (11) requires that the inverse of $\mathbf{X}^T\mathbf{X}$ exist, i.e. that $rank(\mathbf{X}^T\mathbf{X}) < p$. This can be done using pseudo-inverses but that is not addressed in this report.

3.3.4 High-dimensional problems, $p >> N$

During the last two decades the revolution of genomics has lead to an enormous accumulation of biological data from gene sequencing and other techniques. However, a large amount of data is not the same as a large amount of concrete information. With gene data the case is often that we have a much larger number of independent variables p than number of observations N, $p >> N$. This is also the situation in our case, $p = 22297$ and $N = 6431$, which complicates the calculations. When we have a $p >> N$ situation, signals can drown in noise and spurious correlations can occur. We also have a computational challenge with big matrices. In a linear regression model there are p parameters but the $\mathbf{X}$-matrix only has rank N, as mentioned we are not able to estimate all parameters when we use the least square method (11).

3.3.5 Variable selection and shrinkage methods

To control the model complexity, traditional methods as forward- and backward stepwise selection could not be used. Backward selection can only be used when $N > p$ while forward selection was computationally not possible because the large amount of data. Instead the shrinkage method, Lasso (Section 3.3.5.1), was used.

Generally, challenges with variable selection in life science applications is often that: we have more parameters than observations, a nonlinear relationship between input and output, errors in variables, outliers and clusters in the data. With variable selection the variables are either included or excluded in the model and this could generate high variance. Shrinkage methods are smoother and will reduce the variance by penalizing the size of the regression coefficients, which is further described below.

To select variables, prior knowledge as the NetPath genes was also used (Section 3.3.5.2).

3.3.5.1 Lasso

Lasso stands for Least Absolute Shrinkage and Selection Operator and is a shrinkage method that also can be used for variable selection. The linear regression model have to pay some cost, λ, for including a non-zero parameter in the model. The lasso solution is the coefficients that are obtained by

minimization of (12).

$$\hat{\beta}^{lasso} = argmin_{\beta}\{\frac{1}{2}\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\}. \quad (12)$$

A constraint $\sum_{j=1}^{p}|\beta_j| \leq t$ is thus used. When t is large, the constraint has no effect and the solution is the multiple linear least squares regression. However, making t sufficiently small will cause some of the coefficients to be exactly zero.

Coefficient paths are fitted by optimizing each parameter separately, holding the others fixed, as λ varies (Figure in Section 4.4.1).

In this project, lasso was used both as a prediction method and for variable selection. The *glmnet* package in R was used to represent the lasso method (Friedman et al., 2010).

3.3.5.2 NetPath

Another way to reduce dimensionality is to use prior knowledge about the gene expression variables. Netpath is a database that includes 10 cancer signaling pathways that provides a list of the genes that are up- or down- regulated at the level of mRNA expression in a cancer condition (netpath.org, 2005). The reactions in NetPath are compiled from experimental evidence by PhD level scientists (Kandasamy et al., 2010). The 708 listed NetPath cancer genes were used to select the same genes in our dataset.

3.3.6 Methods used in Random Forest

As mentioned, a linear regression model may not be the best model to predict y, the concentration needed for a chemical compound to inhibit the cell line growth with 50%.

Generally, when choosing and applying a method there are also other factors than dimensional problems to consider:

- many algorithms requires that the independent variables are numerical and scaled to similar ranges.

- when the independent variables are highly correlated, linear regression can perform poorly because of high variance.

- if there are complex interactions among the dependent variables this must be specified manually in linear methods.

In real life, effects are often not linear. In genetic data we often have a large number of independent variables, which are often very strongly correlated. It is also common that molecular descriptors are strongly correlated, because they are different reflections of the same underlying molecular property.

All these factors suggest that traditionally used multiple linear regression, which requires exact data and few non-correlated independent variables will perform badly for modeling y.

To construct a statistical model beside using multiple linear regression, a method called random forest was used. The Random Forest method (Section 3.3.7) uses unpruned regression trees (Section 3.3.6.1) with a randomized selection of independent variables at each split to reduce the correlation between trees. By reducing the correlation between the trees the variance is reduced (Section 3.3.6.3). Before the random forest method is introduced, an introduction to regression trees (Section 3.3.6.1), bootstrapping and bagging (Section 3.3.6.2) is needed, because these methods are used in the random forest method.

3.3.6.1 Regression trees

An algorithm such as trees tend to work better than linear methods if there are complex interactions among variables. The main idea behind tree-based methods is to use the data to recursively partition the sample space into smaller and smaller regions. By splitting the space into two regions by an optimal split s that is found over all independent variables p. Where the optimal split is the split that reduce the residual sum of squares, $RSS = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$, the most. This is repeated in a recursive form to build a regression tree until some stopping rule is applied. Figure 2 illustrates an example of a regression tree. The recursively partitions results in a model of the form

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m), \tag{13}$$

where c_m is the constant term for the m-th region which is estimated as the mean of y_i for the observations in the region R_m.

Growing a tree gives a bias-variance tradeoff situation (Section 3.3.2). A larger tree will have smaller regions and result in overfitting. A small tree might not capture important relationships among the variables. How large a tree was grown is discussed in the random forest section, Section 3.3.7.

A problem with trees is that they have high variance. Small differences in the data can result in a totally different tree, since a different split at the top of the tree will affect the splits below it and therefore also the whole tree. The idea of bagging, which is introduced in the next section, is to average many noisy but approximately unbiased trees to reduce the variance.

3.3.6.2 Bootstrapping and Bagging

Bootstrapping is a technique where resampling is used to obtain estimates of summary statistics. The idea is to randomly draw datasets $\mathbf{Z}^{*1}, \mathbf{Z}^{*2}, ..., \mathbf{Z}^{*B}$ of size N with replacement from the training set $\mathbf{Z} = (z_1, z_2, ..., z_n)$ where $z_i = (x_i, y_i)$ is the i-th observation. Each bootstrap sample leaves out

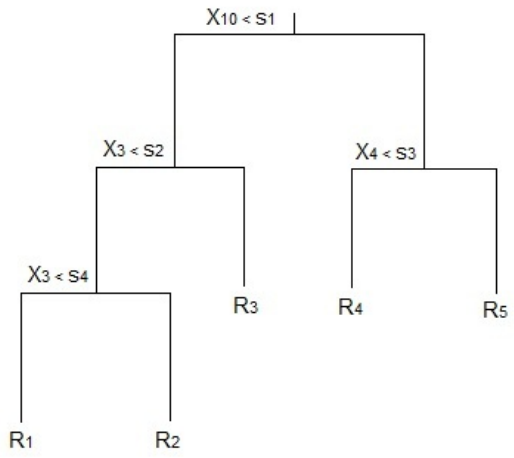

Figure 2: *An example of a regression tree. In the first split, variable X_{10} and the best split point s_1 are used to split the observations into two regions. The observations with a X_{10} value lower than s_1 are divided into two regions by variable X_3 and split point s_2. This process is repeated until some stopping rule is applied.*

roughly 37% of the observations in the training set $\mathbf{Z}$. Since, the probability that the i-th observation z_i is drawn among N observations is $\frac{1}{N}$ and consequently the probability that it will not be drawn is $1 - \frac{1}{N}$. If we randomly draw N observations, the probability that z_i not will be drawn is $\lim_{N\to\infty}(1 - \frac{1}{N})^N = e^{-1} \approx 0.37$ when N is large. The observations that are left out, called *out-of-bag* (OOB), can be used to form estimates of the prediction error. More about this in the random forest section, Section 3.3.7. In bootstrapping the model is fitted for each bootstrap sample $\mathbf{Z}^{*b}$, $b = 1, 2, ..., B$, which gives the prediction $\hat{f}^{*b}(x)$.

Bagging (Breiman, 1996), also called Bootstrap aggregating, averages the prediction of the bootstrap samples to reduce the variance of an estimated prediction function and is defined by

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x). \tag{14}$$

Since the idea in bagging is to average many noisy but unbiased models, trees are ideal candidates. As discussed earlier, trees can capture complex interaction structures in the data, which can result in high variance but low bias if the trees are grown sufficiently deep.

3.3.6.3 The variance of the average when variables are correlated

Generally, an average of n independent and identical distributed random variables $X_1, ..., X_n$ each with $Var(X) = \sigma^2$ has the variance

$$Var(\bar{X}) = Var(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{1}{n^2}\sum_{i=1}^{n} Var(X_i) = \frac{\sigma^2}{n}. \tag{15}$$

Bagging generates identically distributed trees, but not necessarily independent. The variance of the average when we have correlated variables is then

$$\begin{aligned} Var(\bar{X}) &= Var(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} Cov(X_i, X_j) \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \begin{pmatrix} \sigma^2 & \rho\sigma^2 & \dots & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 & \dots & \rho\sigma^2 \\ \vdots & \vdots & \ddots & \vdots \\ \rho\sigma^2 & \rho\sigma^2 & \dots & \sigma^2 \end{pmatrix} = \frac{1}{n^2}\left(n\left(\sigma^2 + (n-1)\rho\sigma^2\right)\right) \\ &= \frac{\sigma^2}{n}\left(1 + n\rho - \rho\right) = \rho\sigma^2 + \frac{1-\rho}{n}\sigma^2, \end{aligned} \tag{16}$$

where ρ is the pairwise correlation.

The second term in equation (16), $\frac{1-\rho}{n}\sigma^2$, disappears when n increases, but the first term $\rho\sigma^2$ remains. Thus, to reduce the variance we need to reduce the correlation. The idea in random forest is to improve the variance reduction of bagging by reducing the correlation, ρ, between the trees.

3.3.7 Random Forest

As an alternative method to linear methods, regression trees were introduced. However, a problem with trees is that they have high variance. To reduce the variance of an estimated prediction function $\hat{f}$, bootstrapping and bagging can be used. By reducing the correlation between trees to further improve the variance reduction, random forest was used.

Random forest is a modification of bagging, that uses unpruned regression trees with a randomized selection of independent variables at each split (Breiman, 2001, 2002). In standard trees, each node is split using the best split among all independent variables. In a random forest, each node is split using the best split, the split that reduce the residual sum of squares the most, among a subset of independent variables m randomly selected at that node. Intuitively, reducing m will reduce the correlation between trees and therefore reduce the variance of the average in (16).

The random forests algorithm for regression:

1) Draw n bootstrap samples $(\mathbf{Z}^{*1}, \mathbf{Z}^{*2}, ..., \mathbf{Z}^{*n})$ of size N from the training data.

2) For each of the bootstrap samples, grow an unpruned regression tree T_b, by recursively repeating the following steps until the minimum node size n_{min} is reached:

   (a) At each node, rather than selecting the best split among all independent variables (as in bagging), randomly select m_{try} variables of the p independent variables.

   (b) Among the m_{try} variables, select the best variable/split-point.

   (c) Split the node into two daughter nodes.

3) After n trees, the random forest predictor

$$\hat{f}_{rf}^{n}(x) = \frac{1}{n}\sum_{b=1}^{n} T(x; \Theta_b) \tag{17}$$

   is an average used to make a prediction for a new observation x. Θ_b characterizes the b-th random forest tree in terms of split variables, cutpoints at each node and terminal-node values.

   The *randomForest* package in R (Breiman, 2001, 2002) was used to generate a random forest. The default value for m_{try} is $p/3$, which also was used in this report. The default node size, n_{min}, is 5 in the *randomForest* package and is also used in this report. Thus, in the tree algorithm nodes with fewer observations than 5 were not splitted.

The number of trees necessary for good performance grows with the number of independent variables. The default value in the randomForest package in R is $ntrees = 500$, which proved to be sufficient even for me.

In the random forest, an estimate of the error rate is obtained by using the out-of-bag data. Out-of-bag was introduced in the previous Section 3.3.6.2 as the observations that are left out in a bootstrap sample. By using the tree grown with the bootstrap sample the observations not included (out-of-bag) in the bootstrap sample is predicted. As explained earlier, when using bootstrapping, each observation will be out-of-bag around 37% of the times. Random forest average the OOB predictions for the i-th observation and calls it $\hat{y}_i^{OOB}$. The "mean of squared residuals" is calculated in the randomForest package in R as

$$MSE_{OOB} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{OOB})^2. \tag{18}$$

The randomForest package also produces two measurements of importance for the independent variables:

- %IncMSE: The average increase in squared out-of-bag residuals when the variable is permuted. For each tree, the mean squared error is recorded on the out-of-bag data and then the same is done after permuting each independent variable, $MSE_{OOB}^{permuted}$. The difference between MSE_{OOB} and $MSE_{OOB}^{permuted}$ is then averaged over all trees and normalized by the standard deviation of the difference. The higher %IncMSE value the higher variable importance.

- IncNodePurity: The total decrease in node impurity (residual sum of squares) from splitting on the variable, averaged over all trees. Higher IncNodePurity value represents a higher variable importance, i.e. nodes are much 'purer'.

Both measurements were used to determine the importance for the independent variables.

3.3.8 Cross-validation

As the statistical methods used in the report have been introduced, the next step is to introduce the validation method used to validate the models. In this section, cross-validation is first introduced in a general way and then further described for the two cases, Case 1: Personalized treatment optimization and Case 2: Drug screening.

In an ideal world we have a data rich situation and we randomly divide the dataset, before doing anything else, into three subsets (Figure 3). A

Figure 3: *The data set divided into three parts: a training set, a validation set, and a test set.*

training set to fit models, a validation set used to estimate prediction error for model selection and a test set used for assessment of the test error, equation (8), of the final model.

In a situation where data is scarce, which almost always is the situation, the cross-validation method can be used to circumvent the problem of scarce data. For K-fold cross-validation the N observations in the data set are randomly allocate to K roughly equal-sized subsets. For the k-th subset (the validation set) the K-1 other subsets (the training sets) are used to fit a model. The model fitted by the training data is then used for predicting

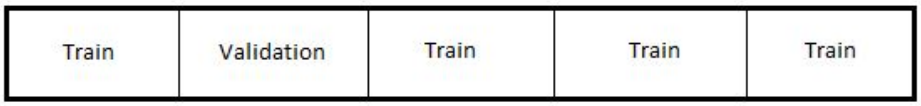

Figure 4: *5-fold cross-validation. The N observations are randomly divided into five subsets. To fit a model for the validation subset the four training subsets are used. This is done for all five subsets.*

the k-th subset of the data. This is done for every subset. Figure 4 shows when $K = 5$, where the second subset is the validation set and the other four subsets are used as training data. The cross-validation estimate of the prediction error is

$$CV(\hat{f}(\mathbf{x})) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-k}(\mathbf{x}_i^T)) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}^{-k}(\mathbf{x}_i^T))^2, \tag{19}$$

where $\hat{f}^{-k}(\mathbf{x})$ is the fitted function, computed with the k-th part of the data removed.

The optimal choice of K is complex (Breiman and Spector, 1992), since we have a bias-variance tradeoff situation (Section 3.3.2). $K = N$ is called *leave-one-out* and the prediction error has a low bias but could have a high variance and are computational difficult. Other common choices are $K = 5$ (5-fold) and $K = 10$ (10-fold), where instead bias could be a problem but the variance is low. I have used 5-fold and 10-fold cross-validation as recommended in Breiman and Spector in 1992 (Breiman and Spector, 1992), and because cross-validation with leave-one-out was computational impossible due to the large amount of data.

3.3.8.1 Cross-validation for Case 1: Personalized treatment optimization

In order to select the optimal drug treatment for a patient's cancer tumor we have no information about the effect for any drugs tested on the patient's particular tumor. For this reason, data was not used from the cell line I wanted to predict the concentration level needed to inhibit the cell line growth with 50%. Thus, before a model was constructed, the data was divided by cell lines.

Before anything else, the 59 cell lines were randomly divided into 5 (or 10) roughly equal-sized subsets. This is illustrated with an example (Figure 4):

> The 59 cell lines are randomly divided into 5 roughly equal-sized subsets. The data from the first cell line are divided into the third subset,

i.e. the 109 observations obtained when the 109 chemical compounds were tested on the first cell line. The data from the second cell line are divided into the second subset, i.e. the 109 observations obtained when the 109 chemical compounds were tested on the second cell line. This is done for all the 59 cell lines. To fit a model for the 109 observations from the first cell line the four other subsets are used, i.e. the third subset is not used to construct a model. The model is then used to predict y_i for the observations in the third subset. To fit a model for the 109 observations from the second cell line the four other subsets are used, i.e. the second subset is not used to construct a model. The model is then used to predict y_i for the observations in the second subset. This is done for each of the 5 subsets.

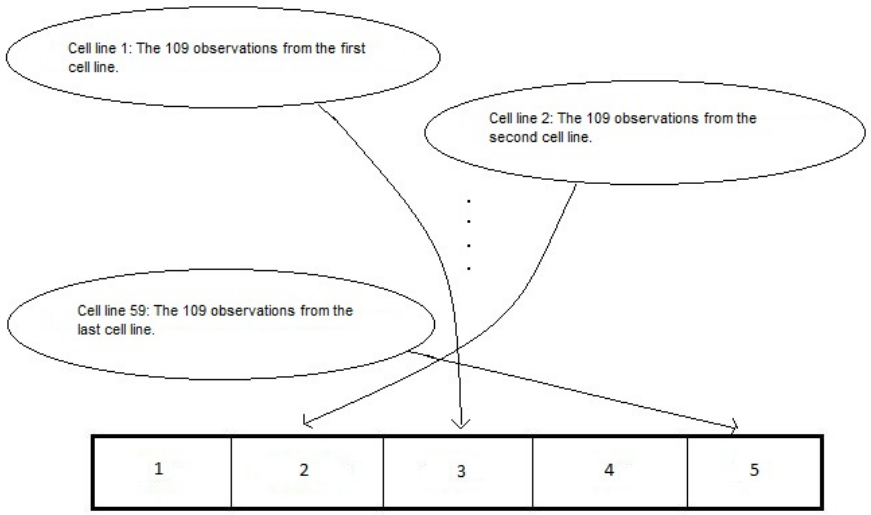

Figure 5: *An example for the personalized treatment case, when 5-fold cross-validation is used. The 59 cell lines are randomly divided into 5 roughly equal-sized subsets. In this example the 109 observations from the first cell line are divided into the third subset, the 109 observations from the second cell line are divided into the second subset and so on. To fit a model for the 109 observations from the first cell line the four other subsets are used, i.e. the third subset is not used to construct a model. This model is used to predict y for the observations in the third subset.*

3.3.8.2 Cross-validation for Case 2: Drug screening

In order to predict the effect of a promising drug we may not have any

information about the effect of the drug on other tumors. For this reason, data was not used from the chemical compound I wanted to predict y for. Thus, before a model was constructed, the data was divided by chemical compounds.

Before anything else, the 109 chemical compounds were randomly divided into 5 (or 10) roughly equal-sized subsets and the construction of models and validation of the models were done in the same way as in Case 1.

3.3.9 Construction of models and prediction of y_i in Case 1: Personalized treatment optimization

The predictions of y_i in the personalized treatment case were made as follow:

1) The 59 cell lines were randomly divided into K (5 or 10) roughly equal-sized subsets (folds), see Figure 5.

2) Variable selection method was chosen to reduce the number of independent variables. One of the three methods random forest, lasso or linear regression was chosen to construct a model for prediction of y_i.

   For each of the K subset samples:

   a) When variable selection was used: the K-1 other subsets were used to preselect independent variables that were used by the method to fit a model.

   b) The K-1 other subsets were used to fit a model with the chosen method.

   c) The model was then used to predict y_i for the K-th subset.

   This procedure was made for each of the K subsets.

3) As predictions of y_i were made for every subset, i.e. all 6431 observations, the prediction error was calculated as

$$MSE = \frac{1}{6431} \sum_{i=1}^{6431} (y_i - \hat{y}_i)^2, \tag{20}$$

   where y_i is the observed response value for the i-th observation and $\hat{y}_i$ is the predicted response value for the i-th observation.

As cross-validation is used, a model is constructed for each subset. Therefore, the prediction error represent the MSE when a specific method is used.

3.3.10 Construction of models and prediction of y_i in Case 2: Drug screening

The predictions of y_i in the drug screening case were made in the same way as in the personalized treatment case (Section 3.3.9), except for the first step. The 109 chemical compounds were randomly divided into K (5 or 10) roughly equal-sized subsets (folds). The predictions were then calculated in the same way as in 2) and 3) in the personalized treatment case.

3.3.11 Permutation test

One of the aims within the project was to do initial research that could give an indication to answer the question: Should we personalize the treatment for a cancer patient or should we give the same treatment to all patients?

To answer the question, the data represented the gene expression variables was randomly shuffled before the following algorithm was used:

1) Between each observation the gene expression values were randomly shuffled and the descriptors were kept fixed.

   An example of how a dataset could look like when gene expression data are permuted:

$$\begin{bmatrix} \overbrace{x_{5,1} & x_{5,2} & \cdots & x_{5,22283}}^{geneexpressions} & \overbrace{x_{1,22284} & \cdots & x_{1,22297}}^{descriptors} & \overbrace{y_1}^{response} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{1201,1} & x_{1201,2} & \cdots & x_{1201,22283} & x_{59,22284} & \cdots & x_{59,22297} & y_{59} \\ x_{436,1} & x_{436,2} & \cdots & x_{436,22283} & x_{60,22284} & \cdots & x_{60,22297} & y_{60} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{3,1} & x_{3,2} & \cdots & x_{3,22283} & x_{6431,22284} & \cdots & x_{6431,22297} & y_{6431} \end{bmatrix}$$

   The descriptors and the response values are kept fixed as the gene expressions are shuffled between the observations. For example, the gene expressions from the fifth observation now represent the first observation.

2) For computational reasons 5-fold cross-validation was used.

3) For each of the 5 subset samples:

   (a) Lasso used the 4 training subsets to select which independent variables that were used in the random forest method.

   (b) The 4 training subsets was used to fit a model with random forest.

   (c) The model constructed in (b) was used to predict the observations in the validation subset.

4) As the predictions y_i were calculated for every subset, i.e. all 6431 observations, the prediction error was calculated as

$$RSS_{permuted} = \sum_{i=1}^{6431} (y_i - \hat{y}_i^{permuted})^2, \tag{21}$$

where y_i is the observed response value for the i-th observation and $\hat{y}_i^{permuted}$ is the predicted response value for the i-th observation in the permuted dataset.

By randomly shuffling the data it was possible to generate as many permuted data sets as liked. To estimate the sampling distribution of $RSS_{premuted}$ the algorithm was repeated 100 times, when the gene expression data was permuted. In order to test if the gene expression variables adds important information the residual sum of squares obtained by using the original data set, RSS_{org}, was ranked among the 100 residual sum of squares from the permuted data, $RSS_{permuted}$.

The test was done analogue for the descriptors to test if the they significantly improved the prediction models.

3.4 Software

To construct the descriptors Bioclipse (Spjuth et al., 2009; bioclipse.net, 2007) was used with the Chemistry Development Kit (CDK). CDK which is an open source Java based library for structural chemo- and bioinformatics (Steinbeck et al., 2006; cdk.sf.net, 2003). R version 2.12.2 and version 2.14.1 were used for statistical analysis. Packages used in R were glmnet (Friedman et al., 2010) and randomForest (Breiman, 2001, 2002). To be able to handle the large dataset, Kalkyl, a high performance computer cluster at UPPMAX (UPPMAX, 2003), was used.

4 Results

4.1 The effect of a chemical compound is similar regardless of cell line

In Section 3.1.3 the GI_{50} was defined as the concentration level of a chemical compound that causes 50% growth inhibition. The response variable, y, is $-log_{10}$ of the GI_{50} and obtained values between 0 and 11. Since a low concentration value is preferable and indicates high efficacy of the drug, a high value on the response y is preferable.

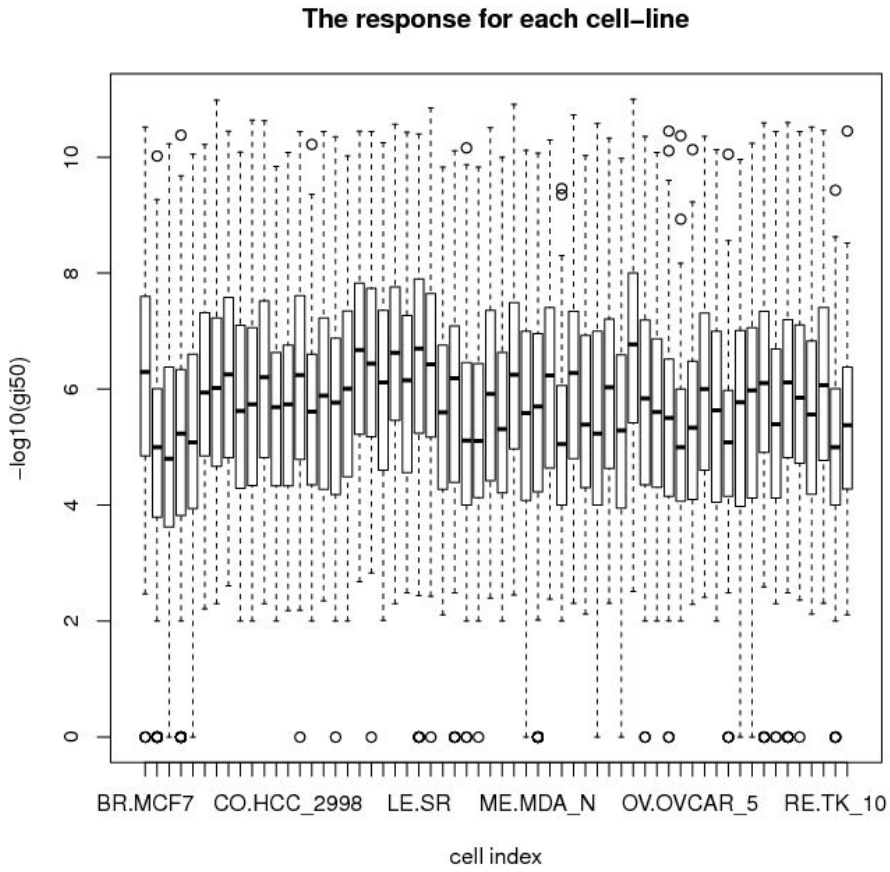

Figure 6: *The response values, grouped by the 59 cell lines, and plotted as boxplots.*

Among the 6431 response values, 109 chemical compounds tested on 59 cell lines, 106 response values (less than 2%) are equal to 0. The rest of the 6325 response values (more than 98%) are values between 2 and 11. The

interpretation of these $y = 0$ values were that concentration values tested on the cell line up to the max dose, 0.01 Molar ($y = 2$), could not cause an growth inhibition of at least 50% and were therefore $y = 0$. 27 cell lines contains at least one response value equal to 0. These values were not treated as missing values and were included in the calculations.

In Figure 6 the response values are grouped by cell lines and illustrated as boxplots, one boxplot for each cell line. The response values are spread out similarly for each cell line, according to median and quartiles.

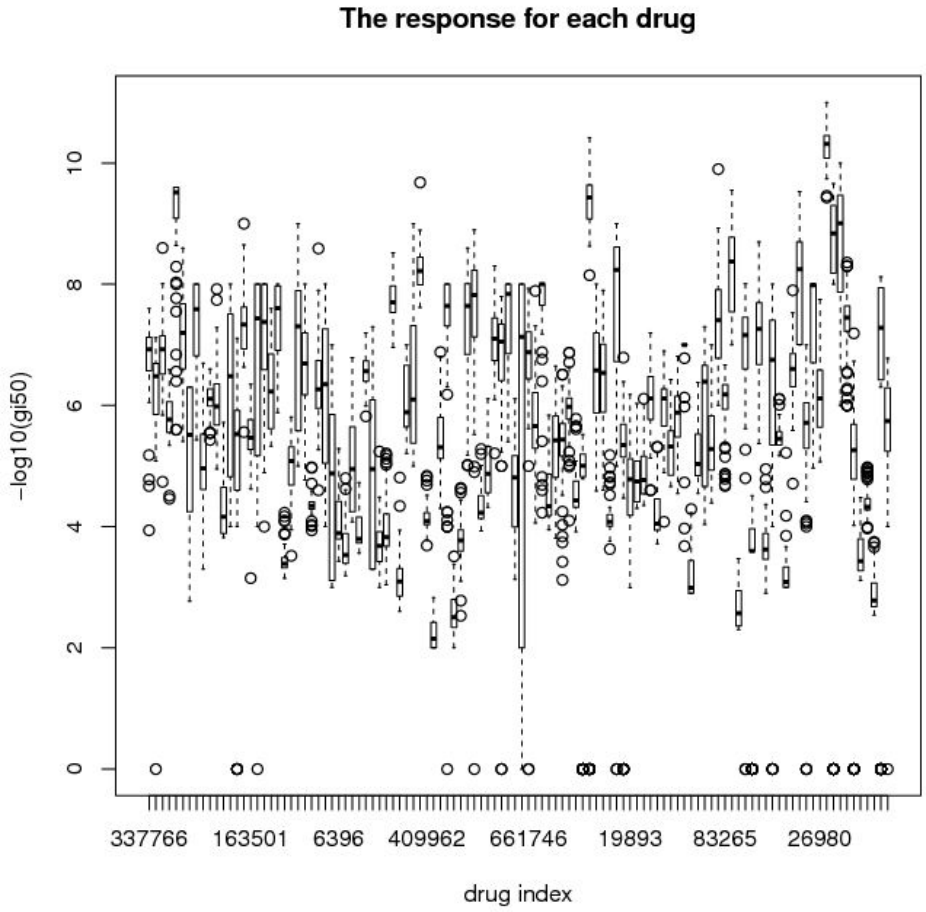

Figure 7: *The response values, grouped by the 109 chemical compounds, and plotted as boxplots.*

In Figure 7 we can observe that the response value range grouped by chemical compounds are much smaller compared to the response value range grouped by cell lines (Figure 6). Thus, among each chemical compound the concentration level that was needed to inhibit the growth with 50% was

similar regardless of which of the 59 cell lines the compound was tested on.

4.2 Correlations between response variables and cancer types

Although it has been shown that cancer are heterogeneous diseases, it seems reasonable that two cancer tumors classified as the same cancer type are more similar to each other than two tumors classified as two different cancer types. If this is the case, a chemical compound should effect cancer tumors classified as the same cancer type similarly. The question then is if we only should use the cancer cell lines from the same cancer type when predicting the response value y, or is it justified to also include cell lines from other types of cancer?

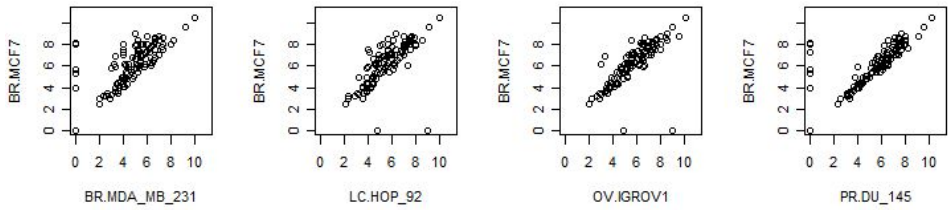

Figure 8: *The response values, y_i, for the breast cancer cell line BR.MCF7 plotted against the response values for four different cancer cell lines. The four other cell lines are categorized as breast cancer, lung cancer, ovarian cancer and prostate cancer.*

In Figure 7 we could observe that a chemical compound gives similar response values regardless of which cell line the compound is tested on. To investigate the correlation between the cell lines, the correlation between y_i from the cell lines of the same type were compared with the correlation between y_i from the cell lines of different cancer types. It turned out that a stronger correlation between cell lines from the same cancer type was not the case for this data. As an example, Figure 8 illustrates the correlation for the response values between a cell line categorized as breast cancer (BR.MCF7) and four other cell lines categorized as breast cancer (BR.MDA_MB_231), lung cancer (LC.HOP_92), ovarian cancer (OV.IGROV1) and prostate cancer (PR.DU_145). We can observe that the response values are approximately equally correlated. In fact the correlation is slightly stronger between the breast cancer cell line and the three cell lines not categorized as breast cancer than the correlation between the two cell lines categorized as breast cancer. Therefore, data was not divided by cancer type when predictions of y_i were made.

4.3 Correlation between the descriptors

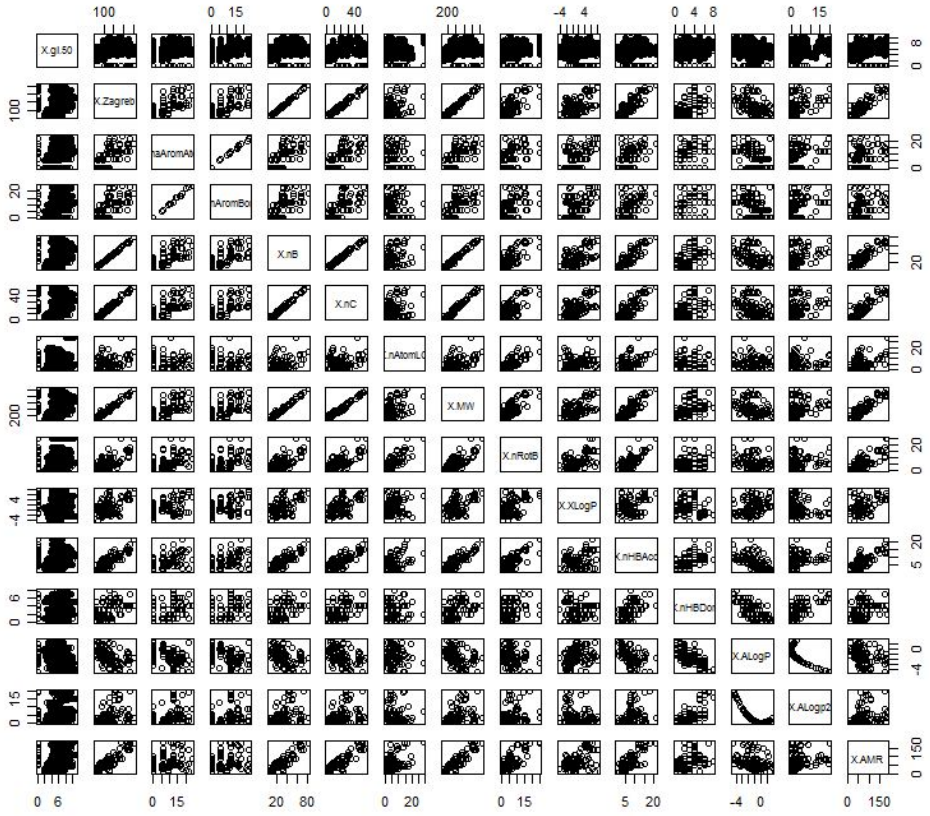

Figure 9: *Scatterplot matrix of the response and the descriptors, where the first row shows the response against each of the descriptors.*

Figure 9 is a scatterplot matrix of the response variable, y_i, and the descriptors. It is difficult to see any clear correlation between the y_i's and the descriptors. However, in a few cases the correlation between different descriptors was very clear. For example, the Zagreb index (X.Zagreb, which is the sum of the squared atom degrees of all heavy atoms) does strongly correlate with the number of bonds (X.nB), the number of carbons (X.nC) and the molecular weight (X.MW). Trying to draw conclusions about the variables' effects when they are highly correlated is difficult. When two or more independent variables in a multiple regression model are strongly correlated, difficulties arise. Even if the matrix $X^T X$ is invertible, an approximate inverse may be numerically inaccurate. However, random forest handles correlated variables well (Breiman, 2001). Therefore, the correlated variables were not removed manually.

4.4 The importance of the independent variables

To determine which variables that are of highest importance three methods was used: the variables selected by lasso, the variable importance measures in the random forest and the selection of genes present in the NetPath database.

4.4.1 Lasso selected the 14 descriptor variables together with 70 gene expression variables

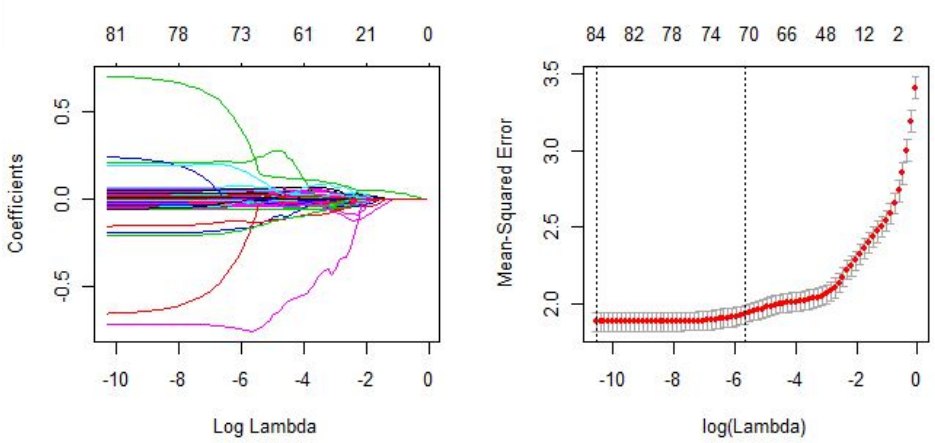

Figure 10: *(Left): The lasso coefficients as $log(\lambda)$ is varied. At the top of the plot the number of coefficients that differs from 0 is presented as $log(\lambda)$ is varied. When λ goes towards 0, i.e. when $log(\lambda) \to -\infty$, the number of coefficients that differs from 0 will be larger. (Right): By 10-fold cross-validation the mean squared error is calculated for a sequence of λ values, where the λ that minimize the mean squared error was selected, the left vertical dashed line.*

The left plot in Figure 10 illustrates the lasso coefficients as $log(\lambda)$ is varied. The number of coefficients that differs from 0 will be larger as λ goes towards 0, i.e. when $log(\lambda) \to -\infty$. For example, the green line at the top is the coefficient for the descriptor nAromBond, the pink line at the bottom is the coefficient for the gene named 22897 and the orange line at the bottom is the coefficient for the descriptor naAromAtom.

The mean squared error was calculated for a sequence of λ values by using the *glmnet* package in R, that used 10-fold cross-validation to select λ. The λ that minimize the mean squared error (Figure 10 right plot) was selected. In this case the model with the lowest mean squared error uses 84 nonzero variable coefficients and an intercept, when λ is equal to $2.68 \cdot 10^{-5}$.

Among the 84 variables that were selected by lasso, the 14 descriptors were included together with 70 gene expression variables.

4.4.2 The descriptors are most important in the random forest method

The variable importance measurements in random forest described in Section 3.3.7, %IncMSE and IncNodePurity, were used to determine which variables that are most important in the random forest method. We can observe in Figure 11 that in the two importance measures, both measures rank the 14 descriptor variables (the variables with letters after X.) as the most important when the random forest was used. Among the descriptors the importance of the variable varies a lot between the two measures, making it difficult to draw any conclusion about the most important descriptor. The same goes for the gene expression variables.

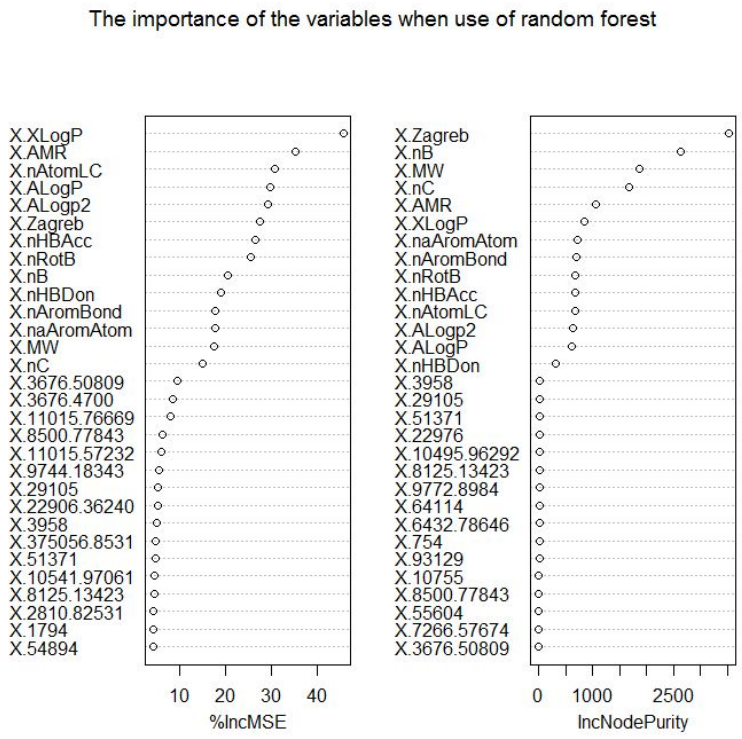

Figure 11: ***%IncMSE:*** *The average increase in squared out-of-bag residuals when the variable is permuted. A higher %IncMSE value indicates a higher importance.* ***IncNodePurity:*** *The total decrease in node impurity (residual sum of squares) from splitting on the variable, averaged over all trees. Higher IncNodePurity value represents a higher variable importance.*

Table 1: *A summary of the predicted results when data was cross-validated for cell lines. The method, variable selection and k-fold cross-validation used, computation time and mean squared error are represented below.*

method	variable selection	k-fold CV	time (h)	MSE
Random forest	-	5	60.7	1.118
Random forest	-	10	148.9	1.092
Lasso	-	5	0.4	2.218
Lasso	-	10	0.9	2.202
Random forest	Lasso	5	0.7	1.113
Random forest	Lasso	10	1.7	1.096
Random forest	NetPath+Descriptors	10	6.8	1.062
Random forest	Descriptors	10	0.1	1.174
Random forest	NetPath	10	6.3	3.361
Linear regression	Lasso	10	1.2	2.999
Linear regression	NetPath+Descriptors	10	3.1	60.475
Linear regression	Descriptors	10	0.1	2.050

4.4.3 1391 gene expression variables were selected with NetPath

By using the 708 NetPath genes, 1391 different probes were obtained. Why the number of variables that were selected is larger than the variables listed is because some of the probes are a fragment of the same gene. Thus, when the NetPath genes were used to select variables the number of gene expression variables were 1391.

4.5 The random forest method is superior to lasso and multiple linear regression in Case 1: Personalized treatment optimization

In Table 1 we have summarized the predicted results. The first column shows the method (lasso, random forest or multiple linear regression) used to construct prediction models that was used to predict the response variable y_i.

If variable selection (lasso, NetPath or/and the descriptors) was used to reduce the number of independent variables before the method was used, this is represented in the second column. For example, if the descriptors are represented in the column, the method could only use the descriptor variables to construct models. If no variable selection was used this is represented by "-".

The third column presents the number of subsets (folds) the data was divided in. The data was divided in 5 or 10 subsets, which means that approximately 80% of the data was used in 5-fold cross-validation to preselect variables and construct a prediction model, for the remaining 20% of the

data. If 10-fold cross-validation was used, approximately 90% of the data was used to preselect variables and construct a model.

The computation time was measured in hours and is presented in the fourth column. The reason why the computation time is interesting is because some calculations took a very long time. Also, to gain access to Kalkyl you may need to queue before your script is calculated. The longer time the calculations take the longer is the queuing time. The time in queue was often longer than the computation time. The time in the fourth column is only the time it took to calculate the script.

Since cross-validation was used, a model was constructed for each of the K (5 or 10) subsets. Therefore, the prediction error, MSE (20), represents the mean squared error for a specific method.

We can observe that the random forest method outperforms both lasso and linear regression in terms of smallest prediction error, MSE. The prediction errors are approximately the same when the random forest method was used with or without lasso as variable selection. However, the total time of calculations was greatly reduced for the random forest method if variable selection was used.

Since the prediction errors are approximately the same for the random forest methods, it was interesting to investigate the variation within the method, before drawing any big conclusions about the optimal variable selection and the optimal cross-validation. Since this was very time consuming the same calculations were performed 9 times more for the random forest method with lasso as variable selection and 5-fold cross-validation. Calculations generated a range of the 10 prediction errors with $MSE = 1.098$ as the smallest value and $MSE = 1.121$ as the highest value. The variation within the method was caused by differently divided data and that different variables could be selected, resulting in a different forest of trees. To investigate only the variation within random forest, the same cross-validation was used for calculating 5 prediction errors. Calculations generated a range of the 5 prediction errors with $MSE = 1.111$ as the smallest and $MSE = 1.114$ as the highest value. This is an indication of that the variation within the random forest method is negligible, and the low variation depends on the cross-validation and variable selection.

The smallest prediction error, $MSE = 1.062$, was obtained when the NetPath genes and the descriptors were used as variables and the random forest method was used. In the left plot in Figure 12, observed response values, y_i, are plotted against predicted response values, $\hat{y}_i$, for the method with the smallest prediction error. 69% of the variation in y_i can be explained by the variation in $\hat{y}_i$, $R^2 = 0.69$. If the zero-values are removed the $R^2 = 0.81$. Except for the zero-values the method seems to predict well. However, the cluster of observations in the upper right corner stand out since the method did not predict any response values between 9.5 and 10. The cluster of observations in the upper right corner are predictions for

the same chemical compound, 357704. In the right plot in Figure 12, the residuals, $e_i = y_i - \hat{y}_i$, are plotted against $\hat{y}_i$. Except for the zero-values we can see a random pattern indicating that a linear model provides a decent fit between y and $\hat{y}$.

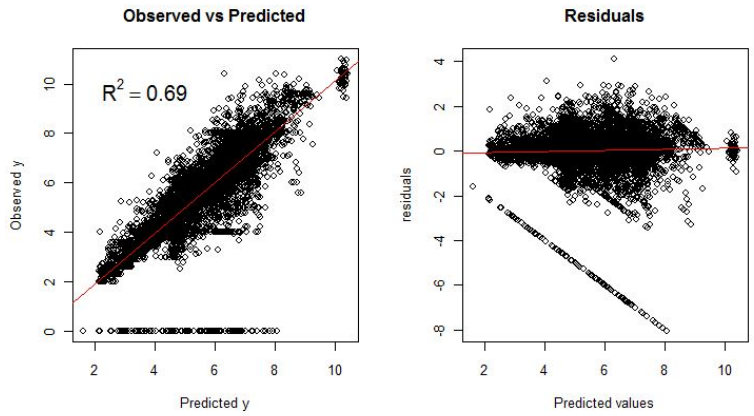

Figure 12: *(Left): The best method in Case 1. Observed response values, y_i, plotted against predicted response values, $\hat{y}_i$, when 10-fold cross-validation, variables selected by NetPath and the descriptors and random forest was used. (Right): The residuals, $e_i = y_i - \hat{y}_i$, plotted against predicted response values, $\hat{y}_i$.*

We can also observe in Table 1 that a random forest method that only used the descriptors gave a much lower prediction error, $MSE = 1.174$, than a random forest method that only used the NetPath genes, $MSE = 3.361$.

In multiple linear regression the prediction from a rank-deficient fit may be misleading and therefore variable selection had to be used to get relevant predictions. We can see that generally, the linear regression method performed badly. The worst scenario was with the NetPath and the descriptor variables, $MSE = 60.457$.

5- and 10-fold cross-validation were compared by using three different methods. By using 10-fold cross-validation the prediction error was slightly reduced compared to 5-fold cross-validation. However, it took twice as long time to perform the calculations with 10-fold cross-validation.

4.6 The random forest method generates the smallest prediction error in Case 2: Drug screening

In Table 2 the predicted results are summarized data is cross-validated for chemical compounds. The random forest method was not as superior in this case as in the personalized treatment case. When the NetPath and

Table 2: *A summary of the predicted results when we cross-validate for chemical compounds. The method, variable selection and k-fold cross-validation used, computation time and mean squared error are represented below.*

method	variable selection	k-fold CV	time (h)	MSE
Random forest	-	5	61.2	2.193
Random forest	-	10	149.2	2.131
Lasso	-	5	0.4	2.249
Lasso	-	10	0.9	2.257
Random forest	Lasso	5	0.8	2.196
Random forest	Lasso	10	1.8	1.976
Random forest	NetPath+Descriptors	10	7.1	2.313
Random forest	Descriptors	10	0.1	2.316
Random forest	NetPath	10	6.5	3.258
Linear regression	Lasso	10	0.9	2.262
Linear regression	NetPath+Descriptors	10	2.9	2.278
Linear regression	Descriptors	10	0.1	2.397

the descriptors variables were used, the prediction error in fact got smaller with linear regression ($MSE = 2.278$) than with the random forest method ($MSE = 2.397$). However, the smallest prediction error, $MSE = 1.976$, was generated by using lasso to preselect variables and the random forest method for prediction. In the left plot in Figure 13, observed response values, y_i, are plotted against predicted response values, $\hat{y}_i$, for the method with the smallest prediction error. 41% of the variation in y_i can be explained by the variation in $\hat{y}_i$, $R^2 = 0.41$. We can see that the predictions generally are worse for higher observed values on y. Most of the observations around $y = 10$ are observations from chemical compound 357704 and were predicted to low. The observations to the right that are predicted around $\hat{y} = 9$ are predicted to high and are all observations from chemical compound 740. In the right plot in Figure 13, the residuals are plotted against $\hat{y}_i$. Except for the zero-values, the observations in the upper left corner (chemical compound 357704) and the observations in the lower right corner (chemical compound 740) the residuals are spread out.

That the prediction error was higher when data was cross-validated for chemical compounds was expected, since the descriptors had higher importance and therefore higher impact on the predictions. However, it was somewhat surprising that lasso and linear regression with preselected variables generated almost as good predictions as the random forest method.

When only the NetPath genes was used in the method the prediction error was $MSE = 3.258$, which also here confirms that the descriptors was important when predictions were made. The computation time was also in the drug screening case hugely reduced when lasso was used. Generally,

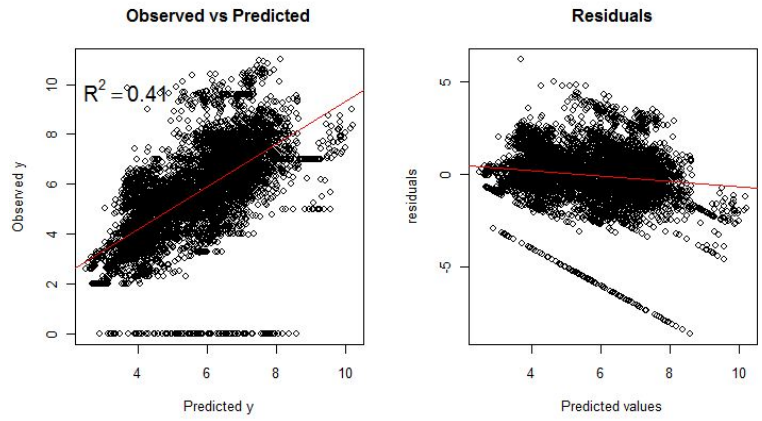

Figure 13: *(Left): The best method in Case 2. Observed response values, y_i, plotted against predicted response values, $\hat{y}_i$, when 10-fold cross-validation, lasso and random forest was used. (Right): The residuals, $e_i = y_i - \hat{y}_i$, plotted against predicted response values, $\hat{y}_i$.*

the prediction error was reduced using 10-fold cross-validation compared to 5-fold cross-validation. However, also in this case it took twice as long time to perform the calculations for 10-fold cross-validation.

4.7 Does the genetic data add any significant information? Should we personalize the treatment for a cancer patient by using characteristics of the cancer tumor?

As said, the most important variables for prediction were the variables that describes the chemical compounds, the descriptors. In both Table 1 and Table 2 we could see that a method without the descriptors produced a very high prediction error. But, how about the gene expression variables? In Case 1, the prediction error was reduced from $MSE = 1.174$ to $MSE = 1.062$ when the NetPath genes were added in the random forest method. In Case 2, the prediction error was only reduced from $MSE = 2.316$ to $MSE = 2.313$ when the NetPath genes were added in the random forest method. A relevant question is therefore if the genetic data adds any significant information? Should we personalize the treatment for a cancer patient or should we give the same treatment to all patients? To test if the gene expression variables adds any significant information, the permutation test introduced in Section 3.3.11 was used.

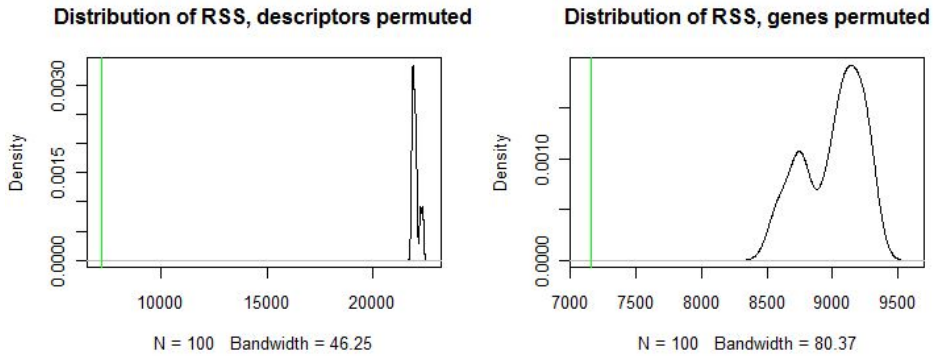

Figure 14: *(Left): A sampling distribution of the residual sum of squares when the descriptors were permuted and cross-validated over cell lines. The green vertical line is the residual sum of squares when the original data was used,* $RSS_{org} = 7160$. *(Right): A sampling distribution of the residual sum of squares when the gene expression data was permuted and cross-validated over cell lines. The vertical line is* $RSS_{org} = 7160$.

4.7.1 The gene expression data adds significant information, indicating that cancer treatment should be personalized

In Figure 14, data was cross-validated over cell lines i.e. Case 1. The left plot in Figure 14 demonstrates a sampling distribution of 100 residual sums of squares when the descriptors were permuted and 5-fold cross-validation, lasso as variable selection and the random forest method were used for prediction. The residual sum of squares obtained when the same methods were used with the "original" (not shuffled) data was $RSS_{org} = 7160$. By ranking the $RSS_{org} = 7160$ value among the 100 $RSS_{permuted}$ values, where the minimum value is $RSS_{permuted} = 21853$, a $p - value = 1/101 < 0.01$ was obtained. This means that the probability of obtaining a $RSS_{permuted}$ value, at least as low as $RSS_{org} = 7160$, is smaller than 1%.

The right plot in Figure 14 demonstrates a sampling distribution of 100 residual sums of squares when the gene expression data was permuted. By ranking the $RSS_{org} = 7160$ value among the 100 $RSS_{permuted}$ values, where the minimum value is $RSS_{permuted} = 8588$, a $p - value = 1/101 < 0.01$ was obtained.

In Figure 15, data was cross-validated over chemical compounds i.e. Case 2. The left plot in Figure 15 demonstrates a sampling distribution of 100 residual sums of squares when the descriptors were permuted. By ranking the $RSS_{org} = 14121$ value (red vertical line) among the 100 $RSS_{permuted}$ values with a minimum value of $RSS_{permuted} = 20937$, a $p-value = 1/101 <$

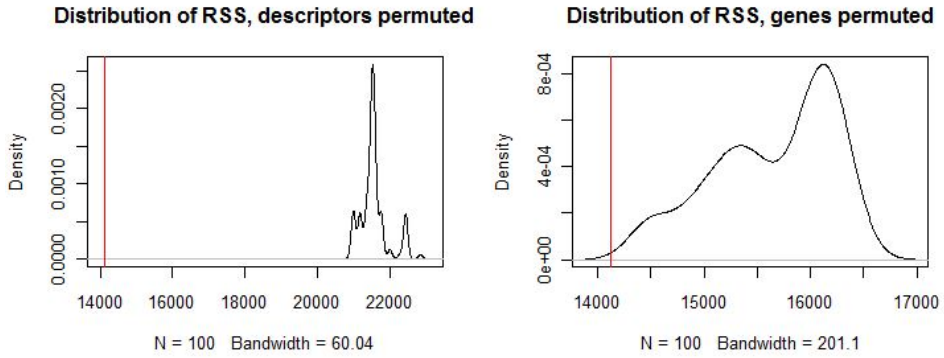

Figure 15: *(Left): A sampling distribution of the residual sum of squares when the descriptors were permuted and cross-validated over chemical compounds. The red vertical line is the residual sum of squares when the original data is used,* $RSS_{org} = 14121$. *(Right): A sampling distribution of the residual sum of squares when the gene expression data was permuted and cross-validated over chemical compounds. The red vertical line is the* $RSS_{org} = 14121$.

0.01 was obtained.

The right plot in Figure 15 demonstrates a sampling distribution of 100 residual sums of squares when the gene expression data was permuted. By ranking the $RSS_{org} = 14121$ value (red vertical line) among the 100 $RSS_{permuted}$ values with a minimum value of $RSS_{permuted} = 14495$, a $p - value = 1/101 < 0.01$ was obtained.

In the both cases the gene expression data adds substantial information. Since the gene expression data represents a description of the cell line, this indicates that cancer treatment should be personalized.

4.8 Prior knowledge vs "blind" variable selection

By using prior knowledge, in this case the NetPath genes that are up- or down-regulated, a question of interest is if we get a better prediction when we use these variables than the variables selected by variable selection?

When random forest was used in Case 1 (Table 1), the prediction error was smaller when the NetPath genes and the descriptors were used ($MSE = 1.062$) compared to when the variables were selected by lasso ($MSE = 1.096$).

When random forest was used in Case 2 (Table 2), the prediction error was considerably smaller when lasso was used ($MSE = 1.976$) compared to when the variables were selected by the NetPath genes and the descriptors ($MSE = 2.313$). Therefore, it is difficult to determine whether the variables

from prior knowledge or the variables selected by lasso should be used to reduce the number of variables. However, the reduction of variables was much greater with lasso, 84 variables compared to 1391 gene expression variables plus 14 descriptors. In a time perspective, the calculations with the lasso variables were four times as fast as the calculations with the NetPath genes and the descriptors.

5 Discussion

Important progress has been made in detection and treatment of cancer over the past twenty years. For instance, it has been realized that cancer diseases are heterogeneous diseases. To achieve the best results for a cancer patient we need the ability to provide personalized treatments based on the patients molecular tumor profile. To be able to develop more personalized treatments we need to develop better methods for drug selection in an optimal way and for this we need efficient approaches to screen for and evaluate promising candidate drugs.

To simulate this, data from 60 human tumor cell lines, called NCI-60, was used. The optimal treatment can be interpreted as the compound that needs the lowest concentration level to inhibit the cell line growth with 50%. However, the aim fo this project was not to select the chemical compound with lowest concentration level, but to use methods to construct models that can be used to predict how a chemical compound affects a particular cancer cell line. In this project the specific aims were: 1) to connect and integrate gene expression data and chemical data, 2) for both cases construct statistical models for predicting the concentration level needed for a chemical compound to inhibit the cancer cell growth with 50%, 3) to investigate if the gene expression data and the chemical data, respectively, are important for the prediction.

By using different techniques within the field of chemometrics it was possible to connect and integrate gene expression data, describing the characteristics of a cancer tumor, and chemical data, describing the properties of a chemical compound. From the analysis we could at an early stage see that the descriptors were more important than the gene expression variables when the predictions were made. Despite this, we demonstrated in Section 4.7.1 that the gene expression variables added significant information for the prediction. These results are very interesting and indicate that we should personalize the chemotherapy for a cancer patient.

In both the personalized treatment optimization case and the drug screening case the random forest method generated the best predictions. The prediction errors with the random forest methods were relatively small, with $MSE = 1.062$ as the smallest prediction error for Case 1. The choice of prediction error measure, MSE, can be discussed. As a result of squaring the difference between y_i and $\hat{y}_i$, MSE places more weight on large errors than on small errors. Thereby, more focus on outliers in the data. The 106 zero-values are very influential on the MSE, when the zero-values were removed the MSE was reduced from 1.062 to 0.576. Many zero-values can generate a model that predicts low predictions overall to reduce the prediction errors for the zero-values. However, since the zero-values are few (<2% of the response values) their impact on the models are small. If the zero-values were removed before the calculations, the prediction error ($MSE = 0.568$) was

almost the same as if they were removed afterwords ($MSE = 0.576$).

In the drug screening case the lowest prediction error was $MSE = 1.976$. Their were especially two chemical compounds that were predicted badly, 357704 (to low) and 740 (to high). In Figure 13 we could see that the predictions generally were worse for higher observed values on y, which unfortunately are the values of highest importance since a high value on y i.e. a low concentration value is favorably. Generally, the prediction error was higher in Case 2, were chemical compounds were cross-validated. This was expected since the descriptors have higher impact on the predictions. However, it was somewhat surprising that the predictions using preselected variables and linear regression was almost as good as the prediction with the random forest method in the drug screening case.

It is important to consider that $y = -log_{10}(GI_{50})$, which means that if we for example predict an observed value, $y = 3$, as $\hat{y} = 5$ this gives a predicted value for $GI_{50} = 10^{-3} = 0.001$ as $GI_{50} = 10^{-5} = 0.00001$.

The computation time was significantly reduced by preselection of variables and further reduced by using 5-fold cross-validation instead of 10-fold cross-validation, without increasing the prediction error markedly. However, as there was a small variation when the data was cross-validated, variables were selected and the trees were grown, it can be risky to draw any radical conclusions about the optimal variable selection and optimal numbers of folds. The best method for future predictions would of course be the method that generates the lowest prediction error for the personalized treatment case and the drug screening case, respectively.

Working with a large dataset can cause many problems, especially if more variables than observations are used. With a large dataset as in this case, it was not possible to perform the calculations on my own computer. To perform the calculations I used Kalkyl, a high performance computer cluster at UPPMAX. The computation time is interesting because, to gain access to Kalkyl we may need to queue before the script is calculated. The longer time the script is assigned, the longer the queuing time. The time in queue is often longer than the computation time and for some scripts it took weeks.

Is it relevant for example to use data from prostate cancer to predict the growth inhibition for a breast cancer tumor? Even though we don't cluster the observations by cancer type in this case, generally, it feels more relevant to use observations from the same cancer type that we want to predict an outcome for. Therefore, I have started working with a dataset containing only breast cancer cell lines.

For future directions a logical next step is to integrate single nucleotide polymorphism (SNP) data, which can help to locate specific genes that are associated with the disease and can work as targets. Similarly, it would be interesting to further investigate which molecular properties that are important and what values on these that are optimal for a chemical compound to

reduce the tumor growth.

In this project we used cell lines because it has the advantages in this case of having abundant publicly available data and the possibility to test new chemical compounds fast and at a low cost without ethical problems. However, an isolated tumor cell line may react differently than it would in its biological context. Eventually, the idea is to take an approach from working on general cell lines to work on specific patients tumor cell lines.

6 Appendix

6.1 An introduction to molecular biology

There exist two main types of cells: prokaryotic cells like e.g. bacteria and eukaryotic cells, where the latter ones are forming multi-cellular organisms like humans. Within each eukaryotic cell there is a nucleus in which the chromosomes are localized. In all human cells, except the germ cells, there are 46 chromosomes in total, where each chromosome is part of a chromosome pair, one from the mother and one from the father, and therefore the 46 chromosomes make up 23 chromosome pairs.

A chromosome consists of a long double-helix DNA (DeoxyriboNucleic Acid) string, which contains a large number of genes that carry the genetic information. The complete DNA sequence with genetic information is also known as the genome, which contains the genetic instructions needed to construct the proteins in the cell. The DNA molecule is buildup by a chain of nucleotides (Figure 16), each harboring a specific nitrogen base, which is a sugar molecule with one or more phosphates. In DNA there are four types

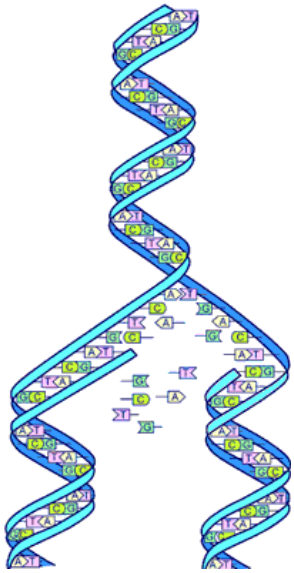

Figure 16: *A chain of nucleotides build the DNA molecule. Each nucleotide is composed with a nitrogenous base, a sugar molecule and one or more phosphates. There are four types of nitrogenous bases in DNA, A connects to T and G connects to C in order to form the double helix string. In DNA replication the DNA molecule is cleaved between the two bases, making two new strings (figure from sv.wikipedia.org).*

of bases: cytosine (C), guanine (G), adenine (A) and thymine (T) where

A connects to T and G connects to C in order to form the double helix string. The sequence of nitrogenous bases in the DNA molecule determines the structure of all proteins in the human body.

In contrast to DNA, RNA (RiboNucleic Acid) is a single-stranded molecule and instead of the nucleic base thymine (T), RNA use Uracil (U). While the more stable DNA molecule is localized in the nucleus of the cell, the unstable RNA molecule is mainly found outside the nucleus. Different types of RNA molecules perform many vital tasks in cells. Three of these tasks are: 1) mRNA (messenger RNA), which transfer information from the DNA in the cell nucleus to the ribosomes which are a protein-RNA machinery where proteins are produced, 2) tRNA (transfer RNA), which deliver amino acids to the ribosomes and 3) rRNA (ribosomal RNA), which is the major component of the ribosome, links the amino acids together to form proteins.

The central dogma of molecular biology (Figure 17) describes the information flow from DNA to protein in biological systems. In most cells, three general steps of transfer the genetic code exist: DNA→ DNA (DNA replication), DNA→ RNA (transcription) and RNA→ protein (translation).

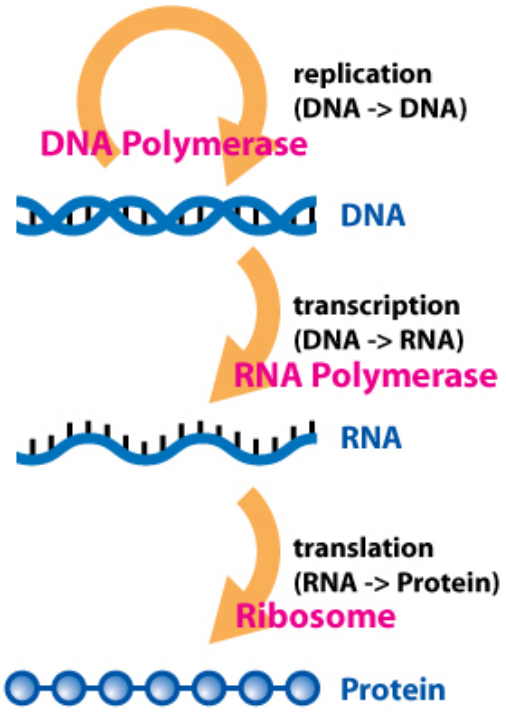

Figure 17: *The central dogma of molecular biology. The general steps of transfering the genetic code to construct proteins (figure from en.wikipedia.org).*

During cell division all DNA is copied in the chromosomes by replication.

The DNA replication process starts with a double-stranded DNA molecule which is cleaved by enzymes by breaking the bonds between the two chains nitrogenous bases. For each chain a complementary chain is produced with the nitrogenous bases as a template. In this way a DNA molecule raises two new, identical DNA molecules (Figure 16).

When a protein is made, an mRNA molecule is first built from the DNA as a template. More specifically, part of the genetic information in the DNA, i.e. a "gene" is copied into an RNA molecule through a mechanism, called transcription (or RNA synthesis). The RNA sequence will be a mirror image of the gene in the DNA-molecule except that thymine (T) has been replaced by uracil (U). The newly synthesized RNA-string (pre-mRNA) has to be transformed into a shorter, mature mRNA molecule before it leaves the nucleus and enters the cytoplasm. There are sections of nitrogenous base pairs that are unnecessary for the protein synthesis, called introns, which are removed by specific enzymes through a process called splicing (Figure 18). The sections that are left of the RNA-string are called exons and binds together into a mature mRNA molecule. Since exons can be spliced together differently, various mRNAs could be obtained from the same gene and consequently lead to determine different proteins.

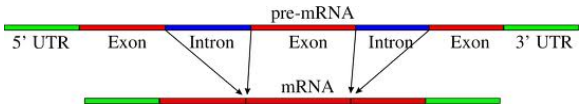

Figure 18: *Splicing. Introns are removed from the pre-mRNA by specific enzymes and the exons are bind together to a mature mRNA molecule (figure from molecularstation.com).*

Mature mRNA is transported from the cell nucleus to the ribosomes, where mRNA is translated to a sequence of amino acids. This is called translation (or protein synthesis). Three nitrogenous bases, called a codon, encodes an amino acid and a sequence of amino acids produce a protein. Proteins have many different tasks in the cell like the transport of substance in to and out from the cell. Special types of proteins, called enzymes, also control chemical reactions in the cell.

To develop new drugs and to adapt the most appropriate drug on a specific patient, we need information about how the genetics regulate the function of the body. With modern biological methods genetic information can be obtained on different "levels" such as DNA, mRNA, proteins and molecules.

Various "omics" technologies are being used to gain information on the genome and it's function. Omics refer to something that is studied in it's entirety. For example *genomics* is used to refer to the field of study of the genome.

A starting point for finding a target for drug discovery is to sequence the genome containing all the 23,000 genes coding for proteins. This was made for the first time in the Human Genome Project that began in 1990 and was completed in 2006. Although the gene sequences varies among humans. Variation in the DNA sequences can affect how humans develop diseases and respond to drugs. One type of variation, called Single Nucleotide Polymorphism (SNP), is when a single base pair (nucleotide) in the genome varies within a population (Figure 19). By identifying specific genes that are associated with a disease, SNPs are believed to be an important part towards personalized treatment and medicine.

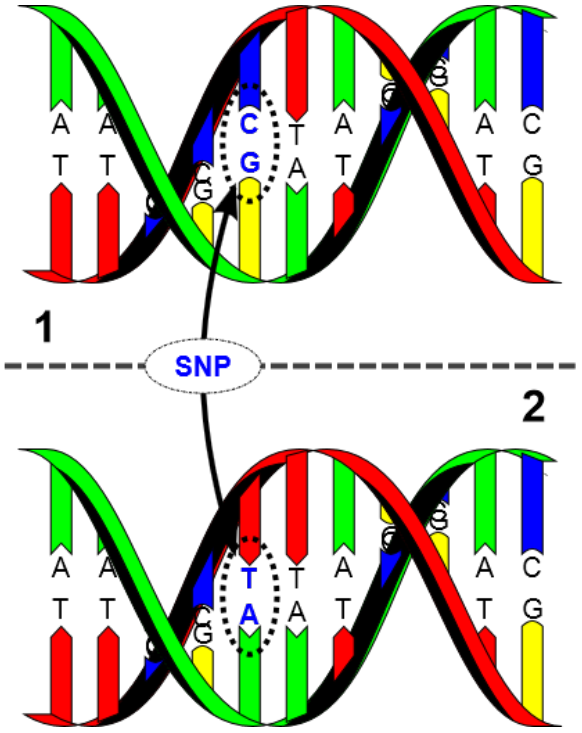

Figure 19: *Single Nucleotide Polymorphism. A single base pair differs between DNA molecule 1 and DNA molecule 2 (figure from en.wikipedia.org).*

Thus, by transcription and translation of DNA into a protein the expression of the genes are transferred to the cell's structure and functions. Even if all cells in the body have the same genome, containing the same DNA sequence, they could produce different proteins. The reason for this is that only some of the genes in a given cell are expressed at a given time. So by enabling (a gene that produce mRNA) and disabling (a gene that don't produce mRNA) genes to be transcribed, different proteins are produced in different cell types.

To profile the gene expression, in other words measure the gene activity (production of mRNA) of thousands of genes simultaneously, a gene chip called microarray is being used. This field is also called *transcriptomics* and refers to the study of the entire set of all mRNA molecules. A DNA array is a solid surface consisting of thousands of microscopic spots of DNA sequences, each sequence corresponds to a short sequence of a gene called a probe. Higher spot signal indicates higher mRNA level and a more active gene, which in a disease state might be an indication that the gene could serve as a new drug target.

Knowing the mRNA level from each gene provides a global picture of the gene expression but there is no strict correlation between the levels of mRNA and the amount of proteins in a cell, since mRNA is not always translated into protein. It can be more relevant to study the entire set of proteins, called *proteomics*. Because the proteome differs between cells and from time to time the task is more complicated than in genomics and transcriptomics.

Genomics, transcriptomics and proteomics can be completed with *metabolomics* that is the study of chemical processes involving small molecule metabolites. Metabolites are the result of metabolism which are the processes that enables the cells to grow and reproduce.

References

Anand, P., Kunnumakara, A., Sundaram, C., Harikumar, K., Tharakan, S., Lai, O., Sung, B., and Aggarwal, B. (2008). Cancer is a preventable disease that requires major lifestyle changes. *Pharmaceutical Research*, 25(9):2097–2116.

bioclipse.net (2007). `http://www.bioclipse.net`.

Boyd, M. R. and Paull, K. D. (1995). Some practical considerations and applications of the national cancer institute in vitro anticancer drug discovery screen. *Drug Development Research*, 34:91–109.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L. (2002). Manual on setting up, using, and understanding random forests v3.1. `http://oz.berkeley.edu/users/breiman/Using_random_forests_V3.1.pdf`.

Breiman, L. and Spector, P. (1992). Submodel selection and evaluation in regression. the x-random case. *International Statistical Review*, 60:291–319.

Bussey, K. J., Chin, K., Lababidi, S., Reimers, M., Reinhold, W. C., Kuo, W.-L., Gwadry, F., Ajay, Kouros-Mehr, H., Fridlyand, J., Jain, A., Collins, C., Nishizuka, S., Tonon, G., Roschke, A., Gehlhaus, K., Kirsch, I., Scudiero, D. A., Gray, J. W., and Weinstein, J. N. (2006). Integrating data on dna copy number with gene expression levels and drug sensitivities in the nci-60 cell line panel. *Molecular Cancer Therapeutics*, 5(4):853–867.

Cancerfonden (2012). `http://www.cancerfonden.se/`.

cdk.sf.net (2003). `http://cdk.sf.net/`.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *Elements of Statistical Learnings, Data Mining, Inference, and Prediction.* Springer.

Hejmadi, M. (2010). *Introduction to Cancer Biology.* Ventus Publishing ApS, BookBooN.

Kandasamy, K., Mohan, S., Raju, R., Keerthikumar, S., Kumar, G., Venugopal, A., Telikicherla, D., Navarro, J., Mathivanan, S., Pecquet, C., Gollapudi, S., Tattikota, S., Mohan, S., Padhukasahasram, H., Subbannayya,

Y., Goel, R., Jacob, H., Zhong, J., Sekhar, R., Nanjappa, V., Balakrishnan, L., Subbaiah, R., Ramachandra, Y., Rahiman, B., Prasad, T., Lin, J., Houtman, J., Desiderio, S., Renauld, J., Constantinescu, S., Ohara, O., Hirano, T., Kubo, M., Singh, S., Khatri, P., Draghici, S., Bader, G., Sander, C., Leonard, W., and Pandey, A. (2010). Netpath: a public resource of curated signal transduction pathways. *Genome Biology*, 11:R3.

netpath.org (2005). `http://www.netpath.org/`.

Shankavaram, U., Reinhold, W., Nishizuka, S., Major, S., Morita, D., Chary, K., Reimers, M., Scherf, U., Kahn, A., Dolginow, D., Cossman, J., Kaldjian, E., Scudiero, D., Petricoin, E., Liotta, L., Lee, J., and Weinstein, J. (2007). Transcript and protein expression profiles of the nci-60 cancer cell panel: an integromic microarray study. *Molecular Cancer Therapeutics*, 6(3):820–832.

Spjuth, O., Alvarsson, J., Berg, A., Eklund, M., Kuhn, S., Msak, C., Torrance, G., Wagener, J., Willighagen, E. L., Steinbeck, C., and Wikberg, J. E. (2009). Bioclipse 2: A scriptable integration platform for the life sciences. *BMC Bioinformatics*, 10:397.

Steinbeck, C., Hoppe, C., Kuhn, S., Floris, M., Guha, R., and Willighagen, E. L. (2006). Recent developments of the chemistry development kit (cdk) - an open-source java library for chemo- and bioinformatics. *Current Pharmaceutical Design*, 12(17):2111–2120.

UPPMAX (2003). `kalkyl.uppmax.uu.se`.

Wikberg, J., Lapinsh, M., Prusis, P., Gutcaits, A., and Lundstedt, T. (2001). Development of proteochemometrics: A novel technology of use for analysis of drugreceptor interactions. *Biochem Biophys Acta*, 1525:180–190.

Wikberg, J. E., Eklund, M., Willighagen, E. L., Spjuth, O., Lapins, M., Engkvist, O., and Alvarsson, J. (2010). *Introduction to Pharmaceutical Bioinformatics*. Oakleaf Academic Publishing House.