Personalized drug adverse side effect prediction

Víctor Bellón Molina

THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres

PSL Research University

Préparée à MINES ParisTech

Personalized drug side effect prediction

Prédiction personalisée des effets secondaires indésirables de médicaments

École doctorale n°432

SCIENCES DES METIERS DE L'INGENIEUR

Spécialité BIO-INFORMATIQUE

COMPOSITION DU JURY :

M Bertram M'ULLERr-MYHSOK
MPI für Psychiatrie, Président

Mme Florence d'ALCHÉ-BUC
Tèlécom ParisTech, Rapporteur

M Jean-Loup FAULON
INRA, Rapporteur

M Pierre NEUVIALI
CNRS, Membre du jury

Mme Véronique STOVEN
Mines ParisTech, Membre du jury

Mme Chloé AZENCOTT
Mines ParisTech, Membre du jury

Soutenue par **Víctor BELLÓN**

le 24 mai 2017

Dirigée par **Véronique Stoven**

Chloé-Agathe Azencott

MINES ParisTech | PSL RESEARCH UNIVERSITY PARIS

Acknowledgements

De res a poc, i sempre amb vent de cara,

quin llarg camí d'angoixa i de silencis.

Miquel Marti i Pol

Ara Mateix

Però amb tot, malgrat tot,

operem i avancem,

pacífics,potser pusil·lànimes,

però mai resignats

i sempre tossuts,

i obrim cada dia

-importuns, enfadosos, burxons-

clivelles de llum en aqueixa presó

on, al cap i a la fi, respirem;

Joan Oliver (Pere Quart)

Versos elementals als catalans de 1969

I would like to thank to my two advisors, Veronique and Chloé, who have allowed me to have the freedom of driving this thesis towards answering the questions that I found interesting while guiding me and advising me. I would also like to thank the colleagues from my group for the scientific discussion and help. Thanks, Yunlong, Benoit, Peter, Beyrem, Marine, Nelle, Elsa, Nino, Jean-Louis, Svetlana, Alice, Xiwei, Olivier, Azadeh, Judith, Joseph, Hector, Thomas and Jean-Phillipe. I would like to thank the Marie Curie Initial Training Network "Machine Learning for Personalized Medicine" for the funding, and all the people that were involved. The ITN have given me the opportunity to be part of an ambitious project, with highly talented people, and assists to

great talks and meetings during the last three years. I would like to thank to the people at the MPI für Psychiatrie in Munich, specially to the group of Professor Muller-Myshok for welcoming and helping me during my internship there. Also, thanks to the people in Roche, specially to Raul Rodriguez Esteban who welcomed me to the group and allowed me to work in an interesting project for three months.

I would like to thank to my friends in Paris: Pau, Andrea, Álvaro and Agata for sharing these years with me.

I could not forget about the Castellers de Paris that have allowed me to do things I did not know I could. We have travelled and lived fantastic experiences together, and the most important, you have become my family here. I would like to specially thank Ester, who convinced me to try it for the first time.

Thanks, also, to my friends back home. They have been a big support when I have needed one: Cris, Alex, Xavi, Didac, Alberto, Israel, Sergio and all the others I might forget, thanks.

Finally, I would like to thank my family. To my mother, who spend many hours, when I was a kid, making up simple mathematical problems for me to solve. To my brother, to whom I ought too much, he faced many problems that he didn't have to, and thanks to that I could continue studding for many years. For finishing, thanks to my uncles and cousins, who have always been an important support.

List of Figures

List of Tables

Contents

1 Introduction

French Abstact

L'objet de cette thèse est l'étude de la prédiction des effets secondaires de médicaments dans le contexte de la médecine personnalisée. Les effets secondaires indésirables jouent un rôle important dans la santé de la population mondiale, et ont un impact économique considérable sur les systèmes de santé publique, les assurances maladie, et l'industrie pharmaceutique. Notre but est de développer des algorithmes d'apprentissage statistique qui pourront nous aider à prédire si un patient est particulièrement susceptible de souffrir d'un effet secondaire particulier après avoir pris un médicament donné.

Dans ce chapitre, nous introduisons le concept d'effet secondaire indésirable et motivons notre ambition de pouvoir les prédire automatiquement. Nous présentons ensuite le paradigme de la médecine personnalisée ainsi qu'une vue d'ensemble des méthodes existantes pour la prédiction personnalisée d'effets secondaires indésirables. Enfin, nous proposons l'utilisation de techniques d'apprentissage statistique que nous présentons en détail, notamment en ce qui concerne l'apprentissage supervisé multitâche. Nous nous attardons plus spécifiquement sur les approches linéaires et à noyaux.

English Abstract

The objective of this thesis is to study the problem of drug side effect prediction in the context of personalized medicine. Drug side effects are an important issue for the health of the global population and have a big economical impact on health systems, insurances and pharmaceutical companies. Our objective

is to develop Machine Learning algorithms that can help us predict whether a given patient will suffer a specific side effect if he or she takes a given drug.

In this chapter, we introduce the concept of drug side effect and our motivation for predicting them. We present the personalized medicine paradigm and the current state of the art on the use of genetic data for solving the personalized prediction of drug side effect. Finally, we propose and introduce the use of Machine learning techniques, with a specific focus on the supervised multitask learning framework and more specifically on linear and kernel approaches to this problem.

1.1 Context

1.1.1 Adverse effects prediction

The World Health Organization defines an Adverse Drug Reaction (ADR) as "one which is noxious and unintended, and which occurs at doses used in man for prophylaxis, diagnosis or therapy" [128]. In the USA, ADRs have been estimated to have annual direct hospital cost of US$1.56 billion [75]. A meta-analysis of the incidence of ADRs [67] has estimated that the incidence of serious ADR in already hospitalized patients was $1.9\% - 2.3\%$ while the incidence of fatal ADR in the same group of patients was $0.13\% - 0.26\%$. The authors estimated that during the year 1994 a total number of 2 216 000 hospitalized patients in the US suffered from a serious ADR, and approximately 106 000 died, which could account for $3.3\% - 6.0\%$ of the total number of recorded death during that year in the US. Posterior studies have found similar results in Europe and Australia [108]. Annual costs of ADR hospitalization have been estimated to be worth 400 million euros per year in Germany [104].

Recent estimates set the cost of drug development in US$2.5 billion in

2013 [34]. A systematic review [84] found 462 medicinal products that were withdrawn from the market in at least one country due to ADR between 1950 and 2014. Of these withdrawals, 114 cases were associated with deaths.

These statistics show that the capability of predicting drug side effects would have an enormous impact on general human health. It would also have a strong economic impact, by reducing both the overall medical cost related to these episodes and the cost on drug development by detecting possible side effects early in their development, or by being able to detect those patients with no risk of suffering from them.

In general, drug side effects occur when drugs bind to off-targets, that is, proteins other than the one targeted, affecting a biological process which evolves in unintended effects. Therefore, the problem of predicting the efficacy of the drug is related to that of predicting its safety for a given patient. Previous studies have shown that different genes are related with the response of the patient or the risk for ADR [123, 44, 120]. This justifies the use of *pharmacogenetics*, which studies the involvement of genes in an individual's response to drugs, to address the issue of ADR prediction.

In [113], the authors discuss the importance of pharmacogenetics and its clinical applications. The goal of pharmacogenetics is to use genetic information to identify subgroups of patients according to the efficacy of a given drug and its safety (i.e. ADR). The efficacy of major drugs varies between different diseases and can go from an efficacy of 80% in the case of analgesics to as low as 25% in the case of oncology drugs. Hence, being able to predict drug efficacy is also of great importance. While the motivation of this thesis lies specifically in adverse effects prediction, the methodological tools we propose can also be applied to efficacy prediction.

Some authors distinguish between two main types of ADR [108]. *Type*

A ADR are the most common type of ADR; they should be predictable as exaggerations of the drug's intended effect and may occur in any individual. They are usually related with primary or secondary pharmacological action of the drug and might be dose-related. *Type B* ADR are uncommon and unpredictable based on the known pharmacology of the drug and only occur in susceptible individuals. Pharmacogenetics can play a role in preventing and understanding both these types of side effects, for example by identifying the different genes that take part in the activity of the drug, or by discovering rare genetic variations that can cause uncommon side effects. The fact that genetic features can play a role in ADRs relates the problem of side effect prediction to that of personalized medicine.

1.1.2 Personalized medicine

Personalized medicine is a recently emerging paradigm that consists in administering the best treatment to the patients according to their overall clinical status, life style, environment, and genetic background. In other words, it consists in classifying the patients who are expected to have similar responses in subgroups, and provide the treatment best fitted to each of these subgroups. Personalized medicine is a term that has been used for several years, but lately a strong claim has risen in part of the scientific community that *precision medicine* should be used [63] instead. The "personalized medicine" term might indeed be misleading, since the objective is not to create a treatment for each person, but to increase the precision of the diagnosis of the patient, so that we can give the best possible treatment at the most appropriate dose.

Precision medicine has gained more and more attention during the last years, not only from the scientific community but also from politicians and the general population. A clear example is the 2015 State of the Union speech, in

which the President of the United States Barack Obama announced the Precision Medicine Initiative. The US government has allocated US$215 millions to the initiative in the fiscal year 2016, and is seeking to recruit a cohort of 1 million volunteers during the first year of the project. The objectives of the initiative go from improving the treatments for cancer to the modernization of regulation to match the necessities of this new research and care model. The French government has also announced a plan for the development of precision medicine, and is planning to invest 670 million Euros during the next years. The plan is called *France Médecine Génomique 2025*.

While personalized medicine takes its roots in the observation that different patients respond differently to the same medication, it is important to note that this difference is greater than that observed for the same individual over his lifetime, or even between monozygotic twins [39]. This implies that genetic factors have an influence in the response of a patient. Unlike other non-genetic factors like age or organ function, these factors remain stable during the patient's life.

Pharmacogenomics. Pharmacogenomics is the field of precision medicine that focuses on the identification of gene variants that play a role in drug response, by changing either the pharmacokinetics or the pharmacodynamics of a drug [98]. Gene variations that affect the pharmacokinetics of the drug will change how it is absorbed, distributed and metabolised, which modulates the actual dose and form of the drug that is available in the body. Gene variations that affect the drug's pharmacodynamics, i.e., that alter its target or the pathway through which it is acting, can inactivate the drug or increase its likelihood to hit off-target proteins. Both can result in unwanted secondary effects; hence, adverse effects prediction can be addressed as a pharmacogenomics problem.

The small signal carried by a great number of gene variants makes the pharmacogenomics problem highly complex. As simple statistical methods fail to solve the problem, the field of machine learning might bring more appropriate approaches.

1.1.3 Machine learning approaches for personalized medicine

Machine Learning is a field of study at the intersection of statistics and computer science that aims to build mathematical models of datasets. These models can be used to extract knowledge from a dataset (i.e. learn) and to make predictions on novel data points.

Machine Learning has obtained growing attention in recent years thanks to its successful application to many fields. It is well known for its success in domains such as face recognition, text translation or text-to-speech tasks. Machine learning is also used in bioinformatics to address many different problems, such as gene expression analysis, gene function prediction, protein structure prediction, or the prediction of interaction between genes, proteins and molecules. More recently, multiple research teams have started focusing their efforts on developing and applying machine learning methods specifically to personalized medicine problems, such as biomarker discovery, survival time prediction, or drug-targetable identification of disease drivers.

In this context, the purpose of this thesis is to **build machine learning models that can be applied for discovering gene variants that modify the response of patients to a treatment and to predict it. In particular, we will focus on multitask algorithms, that will be introduced in Section 1.3.3.**

1.1.4 The machine learning for personalized medicine initial training network

This PhD thesis was conducted under the framework of the Marie Curie Initial Training Network (ITN) *Machine Learning for Personalized Medicine* (MLPM). The objective of the ITN is "to educate interdisciplinary experts who will develop and employ the computational and statistical tools that are necessary to enable personalized medical treatment of patients according to their genetic and molecular properties and who are aware of the scientific, clinical and industrial implications of this research"[1].

In the context of the MLPM ITN, each trainee attended three summer schools and did two different internships. As a trainee, I worked during three months in the Statistical Genetics Group of the Max Plank Institut for Psychiatry in Munich[2]. During this period, I participated in a metanalysis study for discovering SNPs markers for predicting the fast increase of weight in patients under antidepressant treatments. A second project consisted in a study on the association of a functional microsatellite in TLR2 with Inflammatory Bowel Disease, which has been submitted for publication. During this period I also started the work presented in Chapter 4.

I did a second internship at Roche[3]. During this internship, I worked on the problem of identifying gene mentions in scientific articles. Identifying gene names is a difficult task due to different factors: genes have different names, different genes sometimes present the same name, and some of them receive names that can be confused with a term of the common language. We studied the approach of training one single model for each one of the genes that we want to identify. The classical approach consists in using one model to detect a

[1] `http://mlpm.eu`
[2] `https://www.psych.mpg.de/1490813/mueller_myhsok`
[3] `http://www.roche.com`

gene mention without identifying the gene. A second step maps each mention to a gene. We are currently working on the publication of these results.

Although I am happy for the opportunity of these two internships, which were very fruitful experiences, I will not present in more details my contributions to the corresponding projects in this thesis. Here, I will focus on research directly related to the development of machine learning methods for adverse effect predictions, that I conducted as a member of the Centre for Computational Biology (CBIO) of MINES ParisTech, Institut Curie and INSERM.

1.2 State of the art

1.2.1 Adverse effect prediction

Risk factors for ADRs may include genetic and non-genetic risk factors, like alcohol ingest. Traditionally it is difficult to discover genetic risk factors for a given ADR, but new approaches may facilitate the identification of these genetic risk factors [126]. Drug adverse reactions may be caused by a large variety of processes, including mutations in the DNA that affect the drug's protein targets, mutations in the proteins in charge of metabolising the drug, interactions with other drugs, or lack of specificity. In that last case, the drug produces adverse reactions due to off-target interactions.

Until now, most contributions to this field have consisted in trying to predict expected side effects for a given drug, among a defined list of possible side effects observed in drugs. In [87] the authors use the presence of specific chemical substructures in a drug to predict its side effects profile. The relationship between the drugs' side effects and their protein targets profiles has also been exploited by [64] to identify proteins that are highly related with those side effects. Similarly, [25] predict side effects using protein-chemical

and chemical-chemical interactions. Indeed, there exist a large corpus on this topic, e.g. [19, 51, 78, 103]. None of these methods are personalized, in the sense that their predictions are not tailored to the specificities of the patient, but aim at discovering side effects in the general population.

1.2.2 Personalized drug effect prediction

A more personalized approach consists in studying drug-drug interaction (DDI) networks. Indeed, taking these interactions into account may improve the dosage of each drug prescribed to a patient, given the overall list of drugs this patient is exposed to, and avoid potential adverse effects. Usually DDIs are categorized into two different groups that are related with those variations caused by gene mutations. Pharmacokinetic interactions are those in which one drug is affecting the process of absorption, distribution, metabolism or excretion of another drug [133]. On the other side, pharmacodynamics interactions are those in which the effect of a drug is modified by the effect of another [60].

[46] introduces a method that uses different similarity prediction between drugs to not only discover new DDIs, but also gives dosage recommendations. The authors of [28] predict DDIs according to four different similarity measures: a phenotypic similarity based on a drug-ADR network, a therapeutic similarity based on the drug Anatomical Therapeutic Chemical classification system, a chemical structural similarity, and a genomic similarity based on drug-target interaction networks. In [50] the authors present a method to identify pharmacodynamics drug interactions. They observed that known drug pairs causing a pharmacodynamics DDI present a smaller distance between their targets, in protein-protein interaction (PPI) networks than the expected distance. They design a score between set of targets to evaluate the similarity not only on the number of edges connecting genes but also on the expression

of these genes across tissues. In [53] the authors use natural language processing techniques that are commonly mentioned in electronic health records to identify triplets formed by two drugs and an ADR.

1.2.3 Genome-based personalized drug effect prediction

Another personalized approach consists in using the genetic information of the patient to try to predict the possible side effects of a drug. Several gene associations with ADR have been found [126, 4]. There is, in fact, medication which is already labelled with information about genetic risk factors.

Since the early 2000s, several technological advances in the field of genomics have allowed the scientific community to start considering such an approach.

The human genome project (HGP) was a big effort to determine the sequence of base pairs that form the human DNA and identify and map all the genes of the human genome. The project officially started in 1990 and was completed in early 2003 [29]. It received US$2.7 billion funding from the USA government. It was conducted by an international consortium formed by 20 institutions from USA, France, Germany, United Kingdom, Japan and China.

SNP genotyping. The sequencing of the human genome made it possible to identify genomic positions that vary between individuals. If the variation affects a single base pair of DNA, it is called a single nucleotide polymorphism or SNP (pronounced ”snip”). The 1000 Genomes project [72] sequenced more than 38 million SNPs, including common (more than 5% frequency), and uncommon variants. The less frequent allele (variant of the SNP) in a population is called the minor allele, whereas the more common allele is called the major allele.

When SNPs occur in coding regions, they may change the amino acid produced (non-synonymous SNP), affecting the protein sequence by the sub-

stitution of an amino acid (missense mutation), or generating a stop codon leading to an incomplete protein (nonsense mutation). Such sequence modifications may affect the protein structure and/or function. Non-synonymous SNPs are hence good candidates for major phenotypic effects. However, synonymous SNPs (coding SNPs that do not change the corresponding amino acid) and SNPs in non-coding regions have also been associated with changes in phenotypes. This can occur when these SNPs are in linkage disequilibrium (i.e. correlated) with a causal non-synonymous SNP, but also through more complex molecular mechanisms [132].

Note that SNPs only make up part of the genetic variation between individuals. Other common variations are indels, i.e., insertion and deletions of a small sequence of nucleotides in the genome, variations in the copy number of regions of the genome, or translocations between two chromosomes in which large genetic sequences are swapped between non-homologous chromosomes. While the methods we will introduce are applied in SNPs, they can be easily applied to any other genetic data, with minor modifications.

SNP microarrays allow to measure more than 500000 SNPs for a small cost. Even tough SNP arrays are widely used, some criticism can be raised on the fact that they focus on common variants. On the other side, Next Generation Sequencing (NGS) techniques allow to capture both common and rare variants. However, the cost of whole genome sequencing is still too high for its wide application. Another strategy is to focus on the exome. Whole exome sequencing sequences the 2% of the genome containing coding sequences.

Genome-wide association studies (GWAS) usually take common sequenced SNPs from different individuals who suffer from a given pathology and compare them to the SNPs from healthy control individuals. The objective of GWAS is to find SNPs that are statistically different between the two groups.

Current limitations. GWAS approaches have been successfully applied to detecting SNPs which are related to different side effects [80, 115, 4]. One weakness of this approach is that only a small number of drugs can be studied at once, and studies on large tests have not been performed.

A common problem that bioinformatics researchers face when they tackle questions based on human clinical or genetic characteristics is the scarcity of data with respect to its dimensionality. This is usually referred to as the small n large p problem. In this context, statistical tests have less power and p-values significance threshold are smaller due to the necessary multiple testing corrections. It is also difficult to fit models because they easily become overfitted to the data due to the larger number of parameters. Many theoretical results do not hold under this setting, which are not limited to the field of computational biology and are still an area of open research [42, 59].

1.3 Supervised machine learning

Machine learning can be defined as the field that fits mathematical models to data to learn from this data. Machine learning methods can be divided into two large categories of algorithms: supervised and unsupervised learning. Supervised learning deals with inferring a function from labelled examples. Labels are typically discrete (we then talk of classification) or continuous (we then talk of regression). Unsupervised learning, by contrast, deals with the analysis of unlabeled data. The most common unsupervised learning tasks include unsupervised feature extraction, which consists in building new, more informative representations of the dataset, and clustering, which consists in separating the data in different groups that reflect some of its underlying structure.

In this manuscript, we will talk mainly about supervised learning. Supervised learning is related to the task of prediction: after training of a model on

a learning dataset, this model is then able to predict the discrete or continuous labels of new data points.

An example of a supervised learning problem is to learn a model that predicts the prognosis of cancer patients given a learning dataset of cancer patients whose prognosis is known. In the case of unsupervised learning, an example is to discover the population structure of a sample of tissue, i.e., identifying the different types of cells that are present in this sample.

More formally, in supervised learning, we are given a learning dataset $\{X, \boldsymbol{y}\}$, which consists in n training samples, or instances of the data $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$. The input space $\mathcal{X}$ is used to describe the objects about which we want to learn a property, while the output space $\mathcal{Y}$ describes the property we want to learn, which is called the target variable, or output data. The objective of supervised learning is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ that can make a good estimation of the output data from the input data . In other words, a function f is learnt such that $Y = f(X) + \epsilon$, with ϵ being as small as possible. In what follows, we will use $\mathbb{R}^p$ for $\mathcal{X}$. X will then be described as a matrix of n vectors $\boldsymbol{x}_i$, each of these vectors being p dimensional. Each dimension x_j, with $j = 1, \ldots, p$ is called a feature or variable.

Supervised learning can be divided in two different subproblems depending on whether $\boldsymbol{y}$ is a categorical variable (i.e. discrete variable): $\mathcal{Y} = \{0, 1, \ldots, k\}$ or quantitative (real valued): $\mathcal{Y} = \mathbb{R}^q$. The case of a categorical $\boldsymbol{y}$ corresponds to a classification problem, while the case of a quantitative $\boldsymbol{y}$ corresponds to a regression problem.

In this manuscript, we will focus on regression problems and therefore we will start by presenting supervised machine learning methods for regression. More precisely, we will focus on linear models. Then, we will briefly present kernel methods, a group of models that allow to perform non-linear regression.

For historical reasons, we will also briefly introduce neural networks. Finally, we will present an introduction to multitask learning, and we will shortly survey the different approaches that have been used during this thesis.

1.3.1 Linear models

One of the simplest models in Machine Learning is the linear regression, which models the output y as a linear combination of the input features $\boldsymbol{x}_1, \ldots, x_p$:

$$\boldsymbol{y} = X\beta + \beta_0 \tag{1.1}$$

with $X \in \mathbb{R}^{n \times p}$, $\boldsymbol{y} \in \mathbb{R}^n$. $\beta \in \mathbb{R}^p$ is a vector of weights and $\beta_0 \in \mathbb{R}$ is called the bias. For the sake of simplicity, we can consider that the last colunm of X is a colunm of 1 and that β_0 is the last term of β. Therefore, the linear regression equation can be re-written: $\boldsymbol{y} = X\beta$.

One common way to formulate supervised learning problems is to search for a function $\hat{f} \in \mathcal{F}$ that minimizes a loss function $l : \mathbb{R}^{n \times p} \times \mathbb{R}^n \times \mathcal{F} \to \mathbb{R}$, where $\mathcal{F}$ is the space of hypothesis functions

$$\hat{f} = \min_{f \in \mathcal{F}} l(X, \boldsymbol{y}, f). \tag{1.2}$$

The most common loss function for regression is the mean squared error (MSE), which computes the Euclidean distance between the predicted values ($\boldsymbol{y}_i$) and the corresponding true values ($f(\boldsymbol{x}_i)$):

$$l_{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - f(\boldsymbol{x}_i)\right)^2. \tag{1.3}$$

This approach naturally translates in one of the simplest methods, the ordinary least square (OLS) regression. OLS regression consists in minimizing the mean squared error of the prediction in the training set, i.e. the loss

function l_{MSE} we defined above. Given a training dataset of fixed size n, this is equivalent to minimizing the sum of squared errors:

$$\hat{\beta} = \min_{\beta} \sum_{i=1}^{n} (y_i - \beta^\top \boldsymbol{x}_i)^2 \tag{1.4}$$

where $\boldsymbol{x}_i$ is the p-dimensional vector formed by the i-th row of the matrix X. OLS is a convex minimization problem that is easily solvable. If X is a full rank matrix then the exact solution is $\hat{\beta} = (X^\top X)^{-1} X^\top Y$. If it is not, for example when the number of features p is larger than that of samples n, or when the variables are correlated, two situations that are frequent in bioinformatics settings, a solution can be obtained using a pseudo-inverse of $X^\top X$ instead of $(X^\top X)^{-1}$.

OLS regression uses all features of X. Therefore, the model can be difficult to interpret when the number of features p is large. Ideally, we would like to identify a subset of features whose variations lead to the largest effects. Reducing the number of variables might also improve the prediction accuracy. Indeed, a model with many variables is more likely to overfit, that is, to be too adapted to the training data to generalize well to new samples.

A common solution is to shrink the parameters using a penalization function [117, 49]. The most common shrinkage approaches are known as the ridge regression and the Lasso regression. Ridge regression sets a squared penalization on the weights sizes, preventing them from growing too large:

$$\hat{\beta} = \min_{\beta} \sum_{i=1}^{n} (y_i - \beta^\top \boldsymbol{x}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2. \tag{1.5}$$

Ridge regression is a convex optimization problem with solution $\hat{\beta} = (X^\top X + \lambda I)^{-1} X^\top Y$, where I is the identity matrix of size $p \times p$. However, ridge regression does not perform feature selection since it does not tend to set the weights values β_j to 0, but merely restricts their magnitudes. Nevertheless, this usu-

ally leads to models with better generalization properties, i.e. with better prediction performance on external test sets.

The Lasso [117] uses a penalization function on the sum of the absolute values of the weights. This leads to sparser solutions by setting part of the β_j features to 0:

$$\hat{\beta} = \min_{\beta} \sum_{i=1}^{n} (y_i - \beta^\top \boldsymbol{x}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{1.6}$$

In this case, the minimization problem is not convex, and there is no closed form solution to this problem. However, this is a well-studied problem and multiple algorithms exist to solve it [38].

1.3.2 Kernel approaches

Although linear methods are very common, these approaches might be too simplistic in some cases where features are more likely to interact non-linearly to produce the outcome. Kernel methods are a widely-used set of techniques that allow to adapt linear methods to explain non-linear models, thanks to the kernel trick. The kernel trick consists in projecting the instances of the learning dataset in a feature space using a non-linear mapping function ϕ. Using the kernel trick does not required an explicit calculation of the nonlinear mapping, and it can be used as long as the problem can be expressed in terms of scalar products of the instances.

A kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ can be seen as a scalar product in a feature space $\mathcal{H}$, defined as $k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle_{\mathcal{H}}$ where $\phi : \mathcal{X} \to \mathcal{H}$ is a mapping from $\mathcal{X}$ to $\mathcal{H}$. Mathematically, $\mathcal{H}$ must be a Hilbert space, in particular so that the scalar product $\langle , \rangle_{\mathcal{H}}$ is defined. In practice, $\mathcal{H}$ is more often taken to be $\mathbb{R}^d$.

Mercer's therorem allows to characterize kernel functions by representing them as a sum of a convergent sequence of product functions.

Theorem 1.3.1. Mercer's Theorem. Suppose a finite positive measure μ on $\mathcal{X}$, and $k \in L_\infty(\mathcal{X}^2)$ such that the integral operator $T_k : L_2(\mathcal{X}) \to L_2(\mathcal{X})$, defined by

$$T_k : f \mapsto \int_{\mathcal{X}} k(\cdot, x) f(x) d\mu(x) \tag{1.7}$$

is positive definite. Let $\psi_j \in L_2(\mathcal{X})$ be the eigenfunction of T_k associated with the eigenvalue $\lambda_j \neq 0$, and let it be normalized such that $||\psi_j||_{L_1} = 1$ and let $\bar{\psi}_j$ denote its complex conjugate. Then

1. $\{\lambda_j\}_{j \in \mathbb{N}} \in L_1$,

2. $$k(x, x_*) = \sum_{j \in \mathbb{N}} \lambda_j \bar{\psi}_j(x) \psi_j(x_*)$$

   holds for almost all (x, x_*), where the series converges absolutely and uniformly for almost all (x, x_*).

Here, $L_2(\mathcal{X})$ denotes the space of functions from $\mathcal{X}$ to $\mathbb{R}$ for which the square of the absolute value is Lebesgue integrable, $L_\infty(\mathcal{X})$ denotes the space of functions from $\mathcal{X}$ to $\mathbb{R}$ that are bounded up to a set of measure zero, and L_1 is the space of sequences whose series are absolutely convergent.

The Mercer's theorem stated above means that if the kernel function k is positive definite, then it can be written as the inner product of the projection of its two arguments $\boldsymbol{x}$ and $\boldsymbol{x'}$ on a potentially infinite-dimensional space. This allows us to substitute any inner product by a kernel function, and easily extend linear methods to non-linear models.

A common example of a kernel is the radial basis function (RBF) kernel. The RBF kernel is defined as $k_{RBF}(\boldsymbol{x}, \boldsymbol{x'}) = \exp\left(-\frac{||\boldsymbol{x}-\boldsymbol{x'}||_2^2}{2\sigma^2}\right)$, where exp is

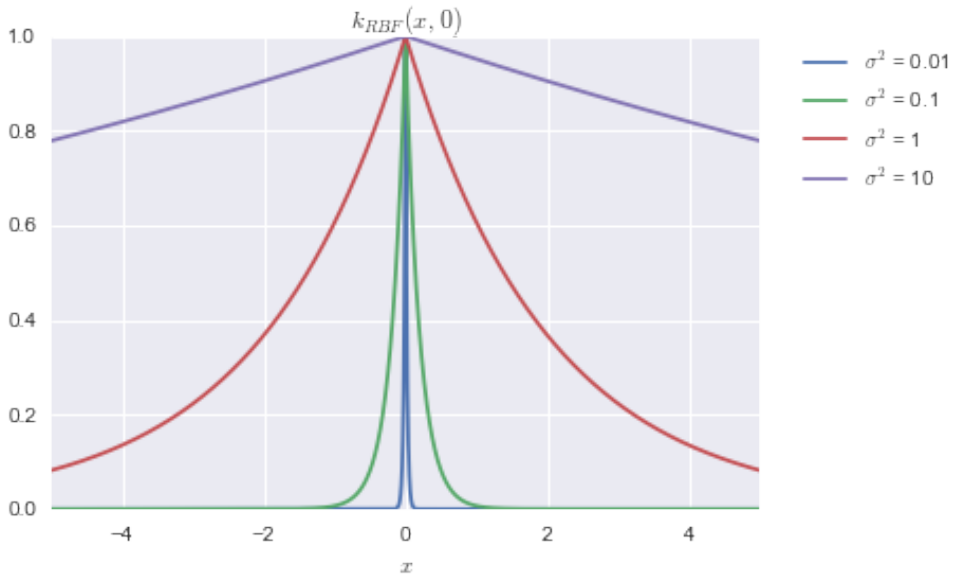


Figure 1.1 – RBF kernel value in a 1-dimensional space applied to x'=0 and $-5 < x < 5$ with different values for the scaling factor σ^2.

the exponential function and σ^2 is a scaling factor. This kernel assigns the same value to two pairs of vectors that are separated by the same distance in the original space. For this reason, it is sometimes considered as a similarity function. Figure 1.1 shows the variations of this kernel with respect to σ^2. The RBF kernel is an example of kernel that maps its two arguments $\boldsymbol{x}$ and $\boldsymbol{x}'$ to an infinite-dimensional space.

Kernel methods are common approaches in the Machine Learning community [20] and in bioinformatics [105]. They are used in Support Vector Machines (SVM) [31] which are a very widely used technique for supervised classification, and in Support Vector Regression (SVR), its counterpart for supervised regression problems. Gaussian Processes [96] are another common technique, and they use kernels as a covariance distribution. In what follows, we introduce the two kernel approaches we used in this thesis: Support Vector Regression and Gaussian Processes.

1.3.2.1 Support Vector Regression

Let us consider a training dataset $D = (X, \boldsymbol{y})$ where $X \in \mathbb{R}^{n \times p}$ and $y \in \mathbb{R}^n$. Let $\boldsymbol{x}_i$ represent each of the columns of X and y_i each of the scalars of $\boldsymbol{y}$. The linear regression problem of finding a function $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b$ that fits the data can be written as the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2} \|\boldsymbol{w}\|^2, \\ &\text{subject to} \quad \begin{cases} y_i - \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b &< \epsilon, \\ \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b - y_i &< \epsilon. \end{cases} \end{aligned} \tag{1.8}$$

This approach assumes that there exists a linear function f that approximates all the data points with precision $\epsilon > 0$. This assumption is not always true. In this case, slack variables ξ, ξ^* can be introduced, generating the new optimization problem:

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2} \|\boldsymbol{w}\|^2 + C \textstyle\sum_{i=1}^n (\xi_i + \xi_i^*), \\ &\text{subject to} \quad \begin{cases} y_i - \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - b &< \epsilon + \xi_i, \\ \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b - y_i &< \epsilon + \xi_i^*, \\ \xi_i, \xi_i^* &> 0. \end{cases} \end{aligned} \tag{1.9}$$

This new problem can be transformed into its dual problem using Lagrange multipliers:

$$\begin{aligned} &\text{maximize} \quad \begin{cases} -\frac{1}{2} \sum_{i,j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\ -\epsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*), \end{cases} \\ &\text{subject to} \quad \begin{cases} \sum_{i=1}^n (\alpha_i - \alpha_i *) &= 0 \\ \alpha_i, \alpha_i^* &\in [0, C] \end{cases} \end{aligned} \tag{1.10}$$

In the dual problem, the weights are expressed in terms of α_i, α_i^* and $\boldsymbol{x}_i$ as $\boldsymbol{w} = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \boldsymbol{x}_i$. This allows to reformulate the predictive function

as:

$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b. \tag{1.11}$$

The link of SVR to kernel methods resides in the fact that all the scalar products in equations (1.10) and (1.11) can be substituted by a kernel function k. It is also important to note that only those instances of the training data fulfilling that $|f(\boldsymbol{x}_i) - y_i| > \epsilon$ contribute to the weights. These points are called support vectors. For more details about support vector regression, one can report to [112]. Support Vector models can be viewed as sparse models, in the sense that not all instances of the training data are used. However, they are sparse in the sense of the number of samples used, not necessarily in the number of features used.

Figure 1.2 shows two SVR models fitting data that would be represented by the function $\sin(x)x$. The RBF kernel provides a non-linear model that fits the data better than the linear model. The Support Vectors are those data points that are on the decision boundaries that delimitate the bandwidth of size ϵ.

1.3.2.2 Gaussian Processes

Gaussian Processes are statistical models that can be seen as distributions over a space of functions. They can hence be used as a prior probability distribution over functions in a Bayesian inference framework. As mentioned above, Gaussian Processes fall in the category of kernel methods, which allows to apply the kernel trick, and helps working with data living in high dimensional spaces. When making a prediction of continuous variables with a Gaussian Process we talk of Gaussian Process regression.

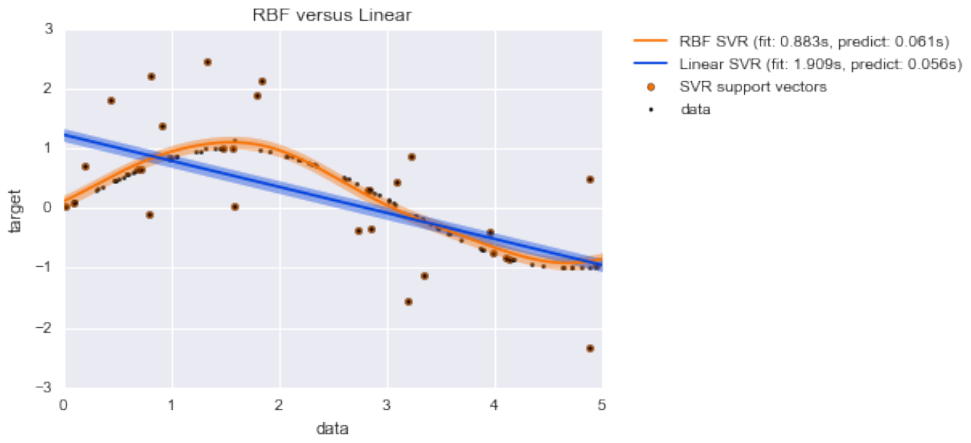


Figure 1.2 – Comparison of a Linear SVR and an SVR using an RBF kernel. Points in color are the selected Support Vectors by the SVR with RBF kernel. Noise is added to some of the points. Both functions show a bandwidth of size $\epsilon = 0.1$.

Definition 1.3.1. A Gaussian Process is a collection of random variables, any number of which have a joint Gaussian distribution.

One of the simplest Gaussian Processes regression models can be derived from Bayesian linear regression. Let us consider the problem of linear regression with Gaussian noise:

$$f(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}, \quad y = f(\mathbf{x}) + \epsilon \tag{1.12}$$

where $\boldsymbol{x}$ is the input vector, $\boldsymbol{w}$ is a vector of weights, f is the function value and y is the target value. As before, we consider a training data set $D = (X, \boldsymbol{y})$ where $X \in \mathbb{R}^{n \times p}$ and $y \in \mathbb{R}^n$. ϵ is a random variable describing the noise. We will assume that it follows an independent and identically distributed Gaussian distribution with zero mean and variance σ^2 :

$$\epsilon \sim \mathcal{N}(0, \sigma^2). \tag{1.13}$$

It is easily seen that, due to the independence assumption and the linearity of the model, the likelihood of the model follows a Gaussian distribution

$$p(\boldsymbol{y}|X, \boldsymbol{w}, \sigma) = \prod_{i=1}^{n} p(y_i|\boldsymbol{x}_i, \boldsymbol{w}) \sim \mathcal{N}(X\boldsymbol{w}, \sigma^2 I), \tag{1.14}$$

where I denotes the identity matrix of size $n \times n$. To follow the Bayesian formalism, we need to define a prior distribution over the parameters of the model. In this case, we assume that the weights follow a Gaussian distribution with 0 mean and covariance matrix Σ_p, of dimensions $p \times p$:

$$\boldsymbol{w} \sim \mathcal{N}(0, \Sigma_p). \tag{1.15}$$

The posterior distribution for the parameters can be obtained by applying Bayes rule:

$$p(\boldsymbol{w}|X, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|X, \boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{y}|X)}. \tag{1.16}$$

This allows to predict for new input $\boldsymbol{x}_*$. For shortness, let us call $f_* = f(x_*)$.

$$\begin{aligned} p(f_*|\boldsymbol{x}_*, X, \boldsymbol{y}) = &\int_{w \in \mathbb{R}^p} p(f_*|\boldsymbol{x}_*, \boldsymbol{w}) p(\boldsymbol{w}|X, \boldsymbol{y}) \mathrm{d}\boldsymbol{w} \\ &\sim \mathcal{N}\left(\boldsymbol{x}_*^\top \Sigma_p \boldsymbol{x} (K + \sigma^2 I)^{-1} \boldsymbol{y}, \right. \\ &\left. \boldsymbol{x}_*^\top \Sigma_p \boldsymbol{x}_* - \boldsymbol{x}_*^\top \Sigma_p \boldsymbol{x} (K + \sigma^2 I)^{-1} \boldsymbol{x}^\top \Sigma_p \boldsymbol{x}_* \right), \end{aligned} \tag{1.17}$$

where $K = \boldsymbol{x}^\top \Sigma_p \boldsymbol{x}$.

This formulation allows us to use the kernel trick. Indeed, $k : (x, x_*) \mapsto x \Sigma_p x_*$ is a kernel function.

We can write Equation 1.17 using only the kernel k, which can be pre-computed. This approach allows us to avoid the problem of working with

high-dimensional data: When $p \gg n$, fitting all the data at the same time can be problematic, whereas a matrix of size $n \times n$ will be more tractable.

It is easy to see that the Bayesian linear regression without noise model (i.e. $\sigma = 0$) fits the definition of a Gaussian Process, for which the joint Gaussian distribution is given by a 0 mean and the covariance function k. Therefore, the random variables $f(x)$ and $f(x_*)$ will follow the following Gaussian distribution:

$$\begin{bmatrix} f(\boldsymbol{x}) \\ f(\boldsymbol{x}_*) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{matrix} k(\boldsymbol{x},\boldsymbol{x}) & k(\boldsymbol{x},\boldsymbol{x}_*) \\ k(\boldsymbol{x}_*,\boldsymbol{x}) & k(\boldsymbol{x}_*,\boldsymbol{x}_*) \end{matrix}\right). \tag{1.18}$$

We can write the distribution of $\boldsymbol{f}_*$ for a new dataset X_*, from a training set $(X, \boldsymbol{y})$ using the mean of the conditional distribution

$$\boldsymbol{f}_*|X_*, X, \boldsymbol{y} \sim \mathcal{N}\left(k(X_*, X)k(X, X)^{-1}\boldsymbol{y}, \tag{1.19}$$

$$k(X_*, X_*) - k(X_*, X)k(X, X)^{-1}k(X, X_*)\right). \tag{1.20}$$

If we just want to perform a regression for a new X_*, we only need to predict the mean for $\boldsymbol{f}_*$. In the case where our observations are noisy, i.e., $f(X) \neq \boldsymbol{y}$, we will consider an independently identically distributed Gaussian additive noise with variance $\sigma^2 > 0$.

$$cov(\boldsymbol{y}) = k(X, X) + \sigma^2 I, \tag{1.21}$$

where $k(X, X)$ denotes the matrix where the entry corresponding to the i-th row and the j-th column corresponds to $k(x_i, x_j)$ and I is again the identity matrix of size $n \times n$. Therefore, we can modify equation 1.18 with the new

distribution for $\boldsymbol{y}$ and obtain:

$$\begin{bmatrix} \boldsymbol{y}) \\ f(\boldsymbol{x}_*) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{matrix} k(\boldsymbol{x},\boldsymbol{x}) + \sigma^2 I & k(\boldsymbol{x},\boldsymbol{x}_*) \\ k(\boldsymbol{x}_*,\boldsymbol{x}) & k(\boldsymbol{x}_*,\boldsymbol{x}_*) \end{matrix}\right). \tag{1.22}$$

Finally, if we marginalize $\boldsymbol{f}_*$ we obtain the predictive distribution for Gaussian Process regression $\boldsymbol{f}_*|X,\boldsymbol{y},X_* \sim \mathcal{N}(\bar{\boldsymbol{f}}_*, cov(\boldsymbol{f}_*))$ where

$$\bar{\boldsymbol{f}}_* = k(X_*,X)[k(X,X)+\sigma^2 I]^{-1}\boldsymbol{y}, \tag{1.23}$$

$$cov(\boldsymbol{f}_*) = k(X_*,X_*) - k(X_*,X)[k(X,X)+\sigma^2 I]^{-1}k(X,X_*). \tag{1.24}$$

A more detailed study of Gaussian Processes can be found in [96, 127].

1.3.3 Artificial neural networks

Artificial neural networks are among the first machine learning methods to have been developed. They were intended to mimic actual neural networks such as the brain.[77]

A neural network is a model that has different layers of neurons (or units), corresponding to variables (Figure 1.3). The first layer usually corresponds to the input data, and the last layer corresponds to the output. The outputs of each layer are the inputs to the units of the following layer. Each of these units is called a perceptron (Figure 1.4). It corresponds to a function, generally non-linear, that calculates a single output from all the inputs that it receives.

Neural networks have regained attention in recent years [68], and nowadays, they are applied to domains as various as natural language processing or biology related problems. Despite their success, neural networks are computationally intensive and require large amounts of data, which are usually not available in the case of genetic studies. Therefore, we did not use them in the present thesis, but mentioned them because of their historical importance in multitask learning.

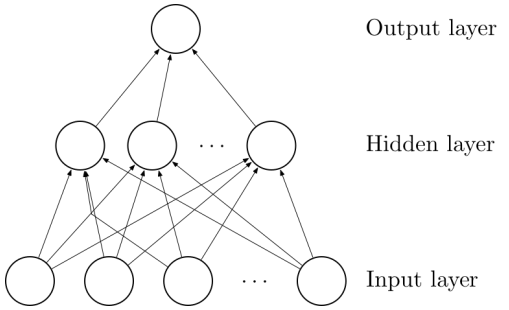


Figure 1.3 – Scheme of a neuronal network with three layers. The first layer corresponds to the input data and the last layer corresponds to the output layer. The middle layers of a neural network are called hidden layers.

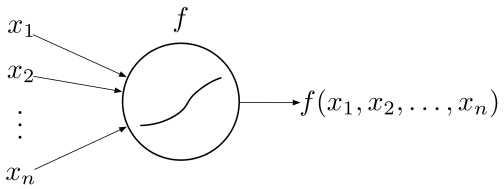


Figure 1.4 – Scheme of a perceptron unit.The perceptron receives the input from several variables and applies a non-linear function f that can be learned from data, and has a single output $f(x_1, x_2, \ldots, x_n)$.

1.4 Multitask Learning

In this thesis, our goal is to build machine learning models that predict the response of patients to various treatments, using data that include genetic information about the patients. A common strategy to solve complex problems is to break them into smaller and independent problems, called tasks, and solve each one of these problems independently, i.e. training a different model for each one of the tasks. In our case, we could break by treatment the problem of predicting the response of patients to their treatments, and build multiple models that each predict the response of patients to a specific treatment.

The main idea behind the so-called multitask learning framework is that, by learning on small related tasks at the same time, and by sharing information between these tasks, we can improve the performance of the final models.

This is of interest when the data available for each task are scarce. Indeed, the more samples are available, the easier it is to learn a good model. The multitask framework makes it possible to share samples across tasks. Since genetic data pertaining to response to treatment is usually scarce, most of the work of this thesis was developed within the multitask learning framework. This makes it possible to share information between the data points available for the different treatments, while building prediction models that are specific to each treatment.

The concept of multitask learning is related to that of inductive transfer, or transfer learning [85]. Transfer learning is motivated from the fact that humans can apply their accumulated knowledge and experience when facing new problems to solve them faster. There are different approaches to transfer learning. According to the classification by Pan and Yang [85], multitask learning falls into the category of inductive transfer learning, meaning that the training and testing domains are the same, i.e. the training and testing data are encoded in the same space, and the training tasks are different but related.

An illustrative example of multitask learning is that of teaching a biped robot how to walk on pavement and on ice. Both tasks are different, and it is hard to code the exact movement that a robot should perform depending on the ground it is walking on. However, a large part of the movements required to walk have common characteristics, whatever the ground might be. Solving the two walking problems while transferring the knowledge acquired in both tasks will be quicker and more efficient than solving each of the walking tasks independently.

In what follows, we will briefly review the main machine learning algorithms

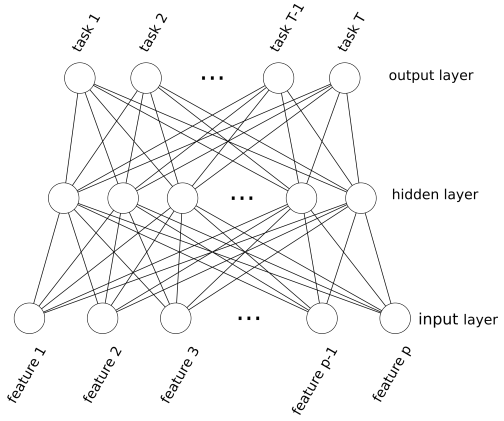


Figure 1.5 – Scheme of the multitask approach in [21]. The tasks share all the input and hidden layers of the network, and each one of them has its own output node.

that have been proposed to solve multitask learning problems.

1.4.1 Artificial neural networks for multitask learning

The first example of a multitask approach in the literature is given by Rich Caruana [21] and uses neural networks. Caruana presents a neural network that learns several tasks at the same time by sharing the hidden representation of the data but providing different outputs for each task, as seen in Figure 1.5.

The model is applied to three different problems to show that multitask learning improves the performance of single task learning methods. A full discussion on how the multitask approach helps to boost the performance of the multitask neural network with respect to the single tasks neural networks, is also given. The mechanisms by which improvement in performance is observed in the multitask neural network include:

Data amplification Multitask algorithms combine data from the different tasks, thus providing higher statistical power and allowing to learn better models.

Feature selection In multitask algorithms, data are encoded in the same

space, which means that they share a common feature representation. A given task might be associated to a small or noisy dataset. In such a case, feature selection will be prone to overfit the model. The use of data from other tasks will help to select better features and learn a better model.

Eavesdropping Let use consider two tasks that share a common feature representation. If the first task it difficult to solve, for example because it uses the features in a complex way, it might be solved more easily if it is learned while sharing information with the other and simpler task.

Representation Bias In some cases, solving an individual task may lead to optimizing a function with multiple minima. Multitask learning algorithms will help detect those local minima that are shared between the different tasks.

While these mechanisms were identified using artificial neural networks, they are believed to apply to the multitask framework at large.

Several successful studies followed the use of multitask neural networks to treat problems in different domains. In [45], the authors studied the application of multitask neural networks for stock selection. Recent breakthroughs on neural networks have allowed to efficiently train deep networks, i.e., networks with many more nodes and parameters than before. Here, we found a successful approach on using the multitask neural network framework to perform different Natural Language Processing tasks [30]. More recent work has used them to predict local properties of proteins [94]. In all these works, the tasks share the lower layers of the network to find common embedding features. However, because of the difficulty of fitting neural networks to small data sets

and the difficulty of interpreting the model we chose not to use multitask neural networks in this thesis, as we previously stated in Section 1.3.3.

1.4.2 Linear models for multitask learning

We consider the state of the art for multitask linear models, and we focus on regularized methods that enable to perform feature selection across the tasks at the same time. In the following, we formalize the multitask linear regression problem.

Let us assume that we want to learn K different tasks, corresponding to K datasets $(X^k, Y^k)_{k=1,\dots,K}$. Let $X^k \in \mathbb{R}^{n_k \times p}$ be the data matrix containing n_k instances of dimension p, and $Y^k \in \mathbb{R}^{n_k}$ the corresponding real-valued output data. Our objective is to find, for every $k = 1, \dots, K$ and for every $i = 1, \dots, n_k$, $\beta \in \mathbb{R}^{K \times p}$ such that

$$y_i^k = f\left(x_i^k\right) + \epsilon_i^k = \sum_{j=1}^{p} \beta_j^k x_{ij}^k + \epsilon_i^k,$$

where ϵ_i^k is the noise for the i-th instance of task k. For each feature j, β_j is a K-dimensional vector of weights assigned to this feature for each task. Notice that x_{ij}^k corresponds to the j-th feature of the i-th instance of dataset X^k. Direct minimization of the loss between Y and f is equivalent to fitting K different linear regressions in a single step. Therefore, this formulation does not allow to share information across tasks.

1.4.2.1 Multitask Lasso and Sparse Multitask Lasso

One of the first formulations for the joint selection of features across related tasks, commonly referred to as Multitask Lasso [82] (ML), uses a method related to the Group Lasso [131]. Information is shared between tasks through a regularization term: An l_2-norm forces the weights β_j of each feature to shrink across tasks, and an l_1-norm over these l_2-norms produces a sparsity

pattern common to all tasks. These penalties produce patterns where all tasks are explained by the same features. This results in the following optimization problem:

$$\min_{\beta \in \mathbb{R}^{K \times p}} \frac{1}{2} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \left(y_i^k - \sum_{j=1}^{p} \beta_j^k x_{ij}^k \right)^2 + \lambda \sum_{j=1}^{p} \|\beta_j\|_2, \tag{1.25}$$

A common extension of this problem is the Sparse Multitask Lasso (MSL), based on the Sparse Group Lasso [111]. It consists in adding the regularization term $\lambda_s \|\beta\|_1$ to Equation 1.25, which generates a sparse structure both on the features as well as between tasks. These sparse optimization problems have been well studied and can be solved using proximal optimization [81].

1.4.2.2 Multi-level Multitask Lasso

To allow for more flexibility in the sparsity patterns of the different tasks, the authors of the Multi-level Lasso [69] (MML) propose to decompose the regression parameter β into a product of two components $\theta \in \mathbb{R}^p$ and $\gamma \in \mathbb{R}^{K \times p}$. The intuition here is to capture the global effect of the features across all the tasks with θ, while γ provides some modulation according to the specific sensitivity of each task to each feature. This results in the following optimization problem:

$$\min_{\theta \in \mathbb{R}^p, \gamma \in \mathbb{R}^{K \times p}} \frac{1}{2} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \left(y_i^k - \sum_{j=1}^{p} \theta_j \gamma_j^k x_{ij}^k \right)^2 + \lambda_1 \sum_{j=1}^{p} |\theta_j| + \lambda_2 \sum_{k=1}^{K} \sum_{j=1}^{p} |\gamma_j^k| \tag{1.26}$$

with the constraint that $\theta > 0$.

The authors prove that this approach generates sparser patterns than the so-called Dirty model [57], where the β parameter is decomposed into the sum (rather than product) of two parameters. In practice, this model also gives

sparser representations than the ML, and has the advantage not to impose to select the exact same features across all tasks.

The optimization of the parameters is a non-convex problem that can be decomposed in two alternate convex optimizations. Furthermore, the optimal θ can be calculated exactly given γ [122]. This optimization, however, is much slower than that of the ML. Finally, note that in this approach, the multitask character is explicitly provided by the parameter θ, which is shared across all tasks, rather than implicitly enforced by a penalization term.

1.4.3 Kernel approaches for multitask learning

In some cases, we might have previous knowledge about the tasks. For example, this prior knowledge can take the form of features that describe these tasks. In the case of adverse effects prediction, if we consider that every treatment corresponds to a different task, this drug can be described by its structure, its chemical properties, its targets or its therapeutic classification. These features can be used to compare tasks and to govern how to share the information of the tasks. Most multitask methods do not use such information: they share information equally across all tasks. In other words, tasks influence each other equally, although one would intuitively prefer that they influence each other based on this tasks features.

The first approach using task features appeared in the context of Bayesian models [9], where the parameters of the model are distributed according to a prior probability distribution. The authors set the parameters of this prior distribution according to the features of the task. Then, they clustered the tasks. Finally, tasks belonging to the same clusters influenced each other more than those belonging to different clusters.

In [41] the authors propose a regularized multitask method that is similar to

the support vector machine algorithm. It relies on weighting the task features according to the sum of two parameters, one which is common to all tasks and one specific for each task.

There have been multiple proposals on how to use task feature descriptors in kernel methods. An interesting approach is to use kernels not only on the features of our instances but also on the features of the tasks, to use nonlinear relations between the tasks. The most common approach is to use the Kronecker product of two kernels [14, 15]. The first kernel corresponds to a similarity measure between the instances while the second corresponds to the similarity between the tasks. The Kronecker product of these two kernels produces a kernel for the instances-task pairs. This resultant kernel allows us to use traditional kernel methods as SVR as multitask methods. This approach might bring some memory problems if there are too many tasks.

Definition 1.4.1. Kronecker Product If A is an $n \times m$ matrix and B is a $p \times q$ matrix, then the Kronecker product $A \otimes B$ is the $mp \times nq$ block matrix:

$$A \otimes B = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1m}b_{11} & a_{1m}b_{12} & \cdots & a_{1m}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1m}b_{21} & a_{1m}b_{22} & \cdots & a_{1m}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1m}b_{p1} & a_{1m}b_{p2} & \cdots & a_{1m}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{n1}b_{11} & a_{n1}b_{12} & \cdots & a_{n1}b_{1q} & \cdots & \cdots & a_{nm}b_{11} & a_{nm}b_{12} & \cdots & a_{nm}b_{1q} \\ a_{n1}b_{21} & a_{n1}b_{22} & \cdots & a_{n1}b_{2q} & \cdots & \cdots & a_{nm}b_{21} & a_{nm}b_{22} & \cdots & a_{nm}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{n1}b_{p1} & a_{n1}b_{p2} & \cdots & a_{n1}b_{pq} & \cdots & \cdots & a_{nm}b_{p1} & a_{nm}b_{p2} & \cdots & a_{nm}b_{pq} \end{pmatrix}.$$

1.5 Contributions of this thesis

In this thesis, we study different problems related to the prediction of the response of individuals to different chemicals and drugs. We will explore different strategies, including subdividing the problem in smaller tasks and using a generic model for all drugs. Then, we will present a novel predictive algorithm that can make interpretable prediction while selecting features for each of these different tasks. Due to the scarcity of the data on side effect reaction and to the fact that other problems share similar characteristics, we didn't apply the models only to ADR data.

Chapter 2. In the next chapter we analyze the data of the *NIEHS-NCATS-UNC DREAM Toxicogenetics Challenge* [37]. In this challenge, different cell lines were exposed to a different set of chemicals and a measure of toxicity was calculated. Toxicity can be understood as an aggregation of effects of a chemical on a cell. We analyze the problem of predicting this toxicity under different assumptions. We study the use of a kernel multitask regression when predicting the toxicity of a wide variety of chemicals. We observe a high correlation of our prediction with the real value, but a small concordance when ordering the predictions according to toxicity. We observed that state-of-the-art algorithms predict the magnitude of toxicity effects better than they make accurate predictions of their values.

Chapter 3. We work on the prediction of the response to treatment of patients with Rheumatoid Arthritis. Rheumatoid Arthritis is a chronic autoimmune disease that causes the inflammation of joints. The work described in this chapter was held in the frame of *DREAM 8.5 – The Rheumatoid Arthritis Responder Challenge* [110]. We participated in this challenge as a team that qualified as the second-best predictive method. We were invited to take part

in the collaborative phase of the challenge, where we realized the following question: Can the addition of genetic data improve the prediction made based only on simple clinical covariates? Part of the work presented in this chapter has been published in [110].

Chapter 4. We introduce a new method, the Multiplicative Multitask Lasso with Task Descriptors. The main idea is to mix the interpretability of regularized linear models with the possibility of explicitly modelling the relationship between tasks that characterize the kernel methods based on Kroenecker products that we studied in previous chapters. We show that it is comparable to state of the art methods in terms of prediction performance. We apply it to the problem of predicting the binding of the Major Histocompatibility Complex alleles with small peptides.

Chapter 5. We propose an extension of the method presented in Chapter 4 to improve the stability in feature selection of our model. We adapt a previous method called the Random Lasso to solve this problem while we analyze other models and the reasons to reject them. This extension is called the Random Multiplicative Multitask Lasso with Task Descriptors. We apply this method to a GWAS data about the flowering time of *Arabidopsis Thaliana*.

2 The Toxicogenetic Dream Challenge

French Abstract

La toxicogénétique s'intéresse à la prédiction de la toxicité potentielle d'un composé chimique exogène pour une personne particulière, sur la base de ses caractéristiques génétiques. Cette toxicité peut être comprise comme l'aggrégation de divers effets délétères dudit composé chimique.

Les agences de régulation ont un grand nombre de composés chimiques sous leur juridiction, mais n'ont de mesures toxicologiques précises que pour un faible nombre d'entre eux. En 2013, le défi « Toxicogenetics DREAM Challenge » a été organisé avec comme objectif d'évaluer les méthodes utilisables pour prédire la toxicité d'un composé chimique dans différentes lignées cellulaires humaines.

Pour développer ces méthodes, les participants disposaient de descripteurs structurels et chimiques des composés chimiques étudiés, et de données génétiques concernant les lignées cellulaires. Les deux objectifs initiaux du défi étaient les suivants : prédire la toxicité d'un nouveau composé chimique sur une lignée cellulaire pour laquelle on dispose de la toxicité d'autres molécules ; et prédire la toxicité sur une nouvelle lignée cellulaire d'une molécule dont la toxicité sur d'autres lignées cellulaires est connue.

Dans ce chapitre, nous utilisons les données de ce défi pour évaluer les performances d'approches d'apprentissage multi-tâches sur les deux problèmes proposés, mais aussi sur le problème, plus difficile, de la prédiction de la toxicité d'un nouveau composé chimique sur une nouvelle lignée cellulaire. Nous montrons que la grande diversité des composés chimiques étudiés limite la capacité des approches multi-tâches à améliorer la prédictivité de leurs équi-

valents simple-tâche. Nous montrons aussi de bonnes performances pour la prédiction de la toxicité d'une molécule déjà étudiée dans de nouvelles lignées cellulaires.

English Abstract

Toxicogenetics is a field that has for objective to determine the potential toxicity of exogenous compounds for a given person, based on his or her genetic background. The toxicity can be understood as the aggregation of various deleterious effects of the chemical.

Regulatory agencies have a large number of chemicals under their jurisdiction, but they only have accurate toxicology in humans for a few of them. In 2013, the Toxicogenetics DREAM challenge took place with the objective of evaluating methods for predicting the toxicity of chemical in different human cell lines. The methods developed used the structural and chemical information of the chemical substances, and genetic information for the cell lines. The initial objectives of the challenge was to predict the toxicity of a new chemical in a cell line for which there was toxicity measures from other chemicals; and to predict the toxicity in a new cell lines of a chemical for which its toxicity was known in another cell lines.

In this chapter, we use the data from this challenge and evaluate the performance of multitask learning on the two proposed problems, but also on the more difficult problem of predicting the toxicity level for a new chemical in a new cell line. We show that the high diversity of the chemicals reduces the potential improvement brought by the multitask approach. We also show good performance when predicting toxicity in new cell lines.

Toxicogenetics aims at determining the potential toxicity of exogenous compounds for a given person based on his/her genetic background. Formerly, it

can be applied to any chemical that the person might be in contact with, or to a drug molecule. In the latter case, toxicogenetics is related to the question of side effect prediction since the toxicity of a drug plays a role in the overall side effects that a patient faces. Therefore, building models for the prediction of toxicity based on genetic information could be of interest for side effect prediction in the context of personalized medicine: being able to predict the toxicity of a given compound for a patient, from his or her genotype, could help avoiding prescribing a drug to patients that would be particularly at risk of enduring toxic side effects.

The Toxicogenetics [37] challenge took place in late 2013, in the context of the 8th set of challenges organized by the Dialogue on Reverse Engineering Assessment and Methods (DREAM)[1]. The aim of this challenge was to predict the effects of toxic compounds on human cell lines. Regulatory agencies have a large number of chemicals under their jurisdiction, but they only have accurate toxicology in humans for a few of them [61]. On the other hand, high-throughput technologies allow for the development of *in vitro* studies over a large number of different cell lines [2], in contrast with previous studies where the number of cell lines is limited [118], allowing to characterize different groups of populations. However, *in vitro* studies only act like a proxy for *in vivo* studies [24, 125]. One of the main reasons is that the compound *biokinetics* are different *in vivo* and *in vitro*, and the biotransformation of compounds can modify its toxicological activity [23]. The problem of not being able to assess the toxicology in humans for a major quantity of the approved chemicals and the ability to test chemicals in a large number of cell lines leaves room to the use of predictive models to their toxic profile [37].

The toxicity of a chemical is defined within this DREAM challenge as the EC10, that is the estimated concentration of the chemical that is necessary to

[1] `http://www.dreamchallenges.org/`

kill a tenth of the cells in the sample. This number can be viewed as a measure that aggregates various deleterious effects caused by a chemical to a living cell. This toxicity depends on the chemical, but also on the genetic profile of the cell, i.e. on the patient. Furthermore, there is no clear distinction between a side effect and a therapeutic effect. Indeed, when treating cancer, the goal is to kill tumor cells and therefore, a toxic effect for the cell can be viewed as a therapeutic effect. In that case, the goal would be to avoid prescribing a drug to a patient whose tumor cells will be resistant to the drug, because these cells will not suffer from toxic effects.

This DREAM challenge was completed a few months before I started my PhD. Overall, the prediction performances of the models built by the participants were modest, and we felt that there could be some space for improvement. In addition, it gave me the opportunity to become more familiar with genetic data, which is critical in the context of personalized medicine, and to gain some skills in chemoinformatics and encoding of molecules, which is required for building side effect prediction models. The only labelled data available was the training dataset. The challenge consisted in two different sub-challenges. The first one focused on the predictability of inter-individual toxicological variation: Can we predict the toxicity of a known chemical on cell lines on which it has not been measured? This question can be related to the prediction of adverse effects: given patients who suffered from an adverse effect, can we predict which new patients will be at high risk of suffering from it? The second subchallenge aimed at the prediction of chemical toxicity from chemical profiles: can we predict the toxicity of a given chemical never tested on a set of cell lines on which the toxicity of other compounds has been measured? Besides these two predefined problems, we also considered a more difficult question: can we predict the toxicity of a given compound for a given

cell line when we neither know the toxicity of this compound on any other cell line, nor that of any other compound for the considered cell line? This case is of interest since there is a large number of chemicals for which cytotoxicity has not been assessed before. Aside from side effect prediction, this can also be of importance for preselecting chemicals to treat tumor cells.

In the next section, we describe the data that were used for the experiment. We continue by the description of the methods used to solve the problem and finally we discuss our results.

2.1 Data

The data for the challenge were obtained by screening 106 common environmental compounds in 884 lymphoblastoid cell lines and annotating the cytotoxicity.

Cell lines Data The cell lines were obtained from 884 lymphoblastoid cell lines derived from participants in the 1000 Genomes Project [3] and represent 9 distinct geographic subpopulations (see Figure 2.1).

Genotype data The genotype data consist of 1.3 million single nucleotide polymorphisms (SNPs) for each cell line. The data were already preprocessed to impute missing SNPs. Each SNP where coded in as 0, 1 or 2 according to the number of minor alleles present, i.e., if the corresponding SNP was homogeneous and contained the major allele it was coded as 0, if it was a heterogeneous SNP, it was coded as 1, and if it was homogeneous but contained the minor allele it was coded as 2.

RNA sequencing The RNA sequencing data were available for 337 cell lines. These data are a quantitative evaluation of all genes expression levels.

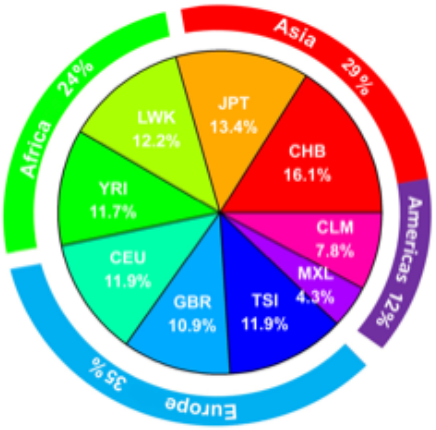


Figure 2.1 – Distribution of subpopulations in the Dream 8 Challenge on Toxicogenetics. The different subpopulations are: Han Chinese in Beijing China (CHB), Japanese in Tokyo, Japan (JPT), Luhya in Webuye, Kenya (LWK), Yoruban in Ibadan, Nigeria (YRI), Utah residents with European ancestry (CEU), British from England and Scotland (GBR), Tuscan in Italy (TSI), Mexican ancestry in Los Angeles California (MXL) and Colombian in Medellin, Colombia (CLM).

Chemical Data The chemical dataset contains 156 compounds chosen from the 1408 chemical compounds in the National Toxicology Program's library, reported by [129]. Chemicals were encoded based on their 2D structure, using a graph representation where nodes are labeled by the corresponding atoms. A visualization of the 2D structure of a chemical can be seen on Figure 2.2.

Cytotoxicity Data Cytotoxicity is represented by the one-tenth maximal effective concentration (EC10). It measures the concentration of a compound at which it induces one-tenth of the maximal cytotoxic response in that sample. Cell lines were randomly divided into 5 screening batches with equal distribution of populations and gender in each batch. COMBAT [58] was used to correct for batch effects.

2.2 Methods

To make all comparisons between different methods as fair as possible, we pruned cell lines and chemical data. We kept only cell lines for which both genotype (SNPs) and RNA sequencing data were available. Only chemicals for which toxicity was known for these selected cell lines were kept. The resulting dataset consisted in 191 cell lines and 106 different chemical compounds.

We predicted cell toxicity using Gaussian Process regression without noise, already introduced in Section 1.3.2.2. We now explain the kernels used for the cell lines and the chemical compounds.

2.2.1 Kernels for chemical compounds

Chemical compounds are usually represented by their structure. This structure can be represented by a graph (Figure 2.2), where the vertices correspond to the atoms and the bonds between them are represented by the edges. This representation can be given in either two or three dimensions; in that later case, coordinates of the atoms in 3D space are provided.

Such a graphic representation is not directly usable by an algorithm, and we would rather like to encode each chemical compound by a vector. This problem can be solved by using extended circular fingerprints (ECFP) [101, 102]. The idea is to list all the possible circular substructures of the molecule and map them to a vector where each position encodes how many times a given substructure is present in the molecule. Usually the circular substructures are limited to a maximum length: we selected all substructures of length up to 9 or 10.

Once the chemicals were encoded with this vector representation, a corresponding kernel matrix can be calculated. [95] proposed several kernels for chemicals. We selected the Tanimoto kernel and the MinMax kernel as they

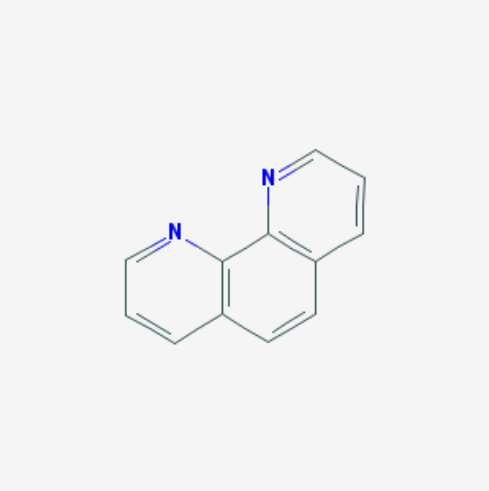


Figure 2.2 – 2D representation of o-phenanthroline. The non-annotated vertices correspond to carbon atoms and hydrogen atoms are not shown.

are commonly used [56, 54, 71].

Definition 2.2.1 (Tanimoto Kernel)**.** Let $\boldsymbol{x}$, $\boldsymbol{y}$ be two binary vectors. The Tanimoto kernel K^t is defined as

$$K^t(\boldsymbol{x}, \boldsymbol{y}) = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\langle \boldsymbol{x}, \boldsymbol{x} \rangle + \langle \boldsymbol{y}, \boldsymbol{y} \rangle - \langle \boldsymbol{x}, \boldsymbol{y} \rangle}$$

In our case, the binary representation encodes whether a path was present in the chemical structure or not. The Tanimoto kernel can be understood as the number of paths that are in both chemical compounds, divided by the number of total paths in both compounds. Another option, which is related to the Tanimoto kernel, is to use the MinMax kernel.

Definition 2.2.2 (MinMax Kernel)**.** Let $\boldsymbol{x}$, $\boldsymbol{y}$ be two vectors of length d. The MinMax kernel K^m is defined as

$$K^m(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{d} \min(x_i, y_i)}{\sum_{i=1}^{d} \max(x_i, y_i)}$$

The MinMax kernel is directly related to the Tanimoto kernel, in the sense that they are equivalent when the vectors are binary. However, it is more general since it considers the number of times that a path is repeated

2.2.2 Kernels for cell lines

We calculated cell lines kernels for both types of data, RNAseq and SNPs. In both cases, data were preprocessed by different methods.

For the RNAseq data we used the correlation as kernel function. Given two vectors $\boldsymbol{x}, \boldsymbol{z} \in \mathbf{R}^p$ we can define their correlation coefficient r_{xz} as

$$r_{xz} = \frac{\sum_{i=1}^{p}(x_i - \bar{x})(z_i - \bar{z})}{\sqrt{\sum_{i=1}^{p}(x_i - \bar{x})^2 \sum_{i=1}^{p}(z_i - \bar{z})^2}}, \tag{2.1}$$

where $\bar{x}$ and $\bar{z}$ denote the mean of vectors $\boldsymbol{x}$ and $\boldsymbol{z}$. Our group had participated in the DREAM challenge before its completion. In this previous work, all available SNPs were used [12], although many of them are expected to be unrelated to the biological problem under study. Therefore, in the present work, we decided to restrict the considered SNPs to SNPs that might modulate the observed toxicological effect.

We chose n a reduced set of proteins, instead of using the whole genome expression data. We only selected those genes that encode the 231 proteins that were related with toxicity effects in the literature [64]. Indeed, although side-effects are observed at the level of the patient, it appeared plausible that they resulted at least in part from cytotoxic effects, and we hoped that retaining only the corresponding genetic information could help to increase prediction performance.

In the case of the SNPs data, we selected presumed deleterious features according to PolyPhen [5], SIFT [65] or MutationTaster [106]. Those SNPs are most likely to be involved in diseases and therefore, potentially, in response to chemical exposure. This resulted in a list of 2763 SNPs. After selecting the

SNPs, we calculated both the Tanimoto and MinMax kernel. Instead of using the kernels on subpath of molecular graphs, as we did in the previous section, we use these kernels on the number of minor alleles present in each SNPS.

2.2.3 Kernels for chemicals and cell lines pairs

In a multi-task approach, we consider a single model across all chemicals and kernels. The objects are now pairs formed by one chemical and one cell line. As explained in Section 1.4.2.2, a kernel between (chemical, cell line) pairs can be simply formed as the Kronecker product between any of the kernels between chemicals and any of the kernels between cell lines described above:

$$K((d_1, c_1), (d_2, c_2)) = K_{\text{chemical}}(d_1, d_2) \times K_{\text{cell}}(c_1, c_2)$$

where d_1, d_2 are two chemical compounds and c_1, c_2 two cell lines. [15]

2.3 Results

As described above, we solved three different tasks in this project: two tasks corresponding to the two DREAM subchallenges, i.e. predicting the toxicity for new cell lines and predicting it for new compounds, and a more difficult task, i.e. predicting the toxicity for both new chemicals and new cell lines. For each experiment, a 10-fold cross-validation was performed. In the third case the folds were a combination of the folds for new compounds and the folds for new cell lines.

In addition to the kernels described above, we also used the identity matrix as a kernel for either the chemical compounds and the cell lines to check whether the multitask approach gives any improvement over single task approaches in the first two experiments. Indeed, using the identity kernel for one of them is equivalent to solving the corresponding tasks independently.

We estimated the performance of the methods using different measures. We used the concordance index (CI), the normalized root mean squared error (RMSE) and Pearson's correlation (PC).

CI is a well-known measure in survival prediction that measures the proportion of pairs of samples that were correctly ordered according to its predicted value. When for a pair of samples, the exact same prediction is obtained, it counts as 0.5 instead of 1 for a complete match. This was applied to compare the orders of predicted pairs of toxicity values and their real counterparts. A method that orders predictions correctly will have a perfect CI of 1 whereas a method that returns the same results for all inputs will have a CI of 0.5.

The normalized RMSE (nRMSE) is the root mean squared error divided by the difference between the maximum and minimum values of the true predicted variable. Normalizing the RMSE makes the interpretation of the results simpler. With this normalization, we remove the effect of the scale of the data.

$$NRMSE\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = \frac{\sqrt{\frac{\sum_{i=1}^{n}(\hat{y_i}-y_i)^2}{n}}}{y_{\max}-y_{\min}}, \tag{2.2}$$

where $\boldsymbol{y} \in \mathbf{R}^n$ is a vector containing the measured values, $\hat{\boldsymbol{y}} \in \mathbf{R}^n$ is a vector containing the predicted values, and $y_{\max}$ and $y_{\min}$ denote the maximum and minimum measured values respectively.

Pearson Correlation (Equation 2.1) is a measure of the linear dependence between two variables. It gives a maximum and minimum value of +1 and -1 for total positive linear correlation and total negative linear correlation respectively. For random results, we would expect and absence of dependence, and therefore we will obtain a correlation value of 0.

We report the cross-validated CI performance of predicting the toxicity of new cell lines across the chemicals in Figure 2.4. We also present the RMSE of the predictions in Figure 2.5. The maximum CI is obtained when using

the RNAseq data with the identity kernel for chemicals, or equivalently, in a single-task setting. The minimum normalized RMSE obtained is 0.038, when using the MinMax kernel for SNP data and the MinMax kernel for chemicals fingerprints with paths of length up to 9. The performance does not change significantly across the different chemical kernels. We observe that we obtained a bad CI while obtaining good RMSE. As it can be seen in Figure 2.6, while the magnitude of the effect is correctly predicted, as shown by the low RMSE, the predictions are not accurate enough to be correctly ordered.

We also present Pearson's correlation and the normalized RMSE for predicting the toxicity of new chemicals in Figures 2.7 and 2.8. We can see that the RMSE is worse when of predicting for new chemicals than for new cell lines. The performance in this setting is dominated by the selected chemical kernel. In this case, the best performing kernel is the Tanimoto kernel with subpath of length up to 9, as shown in Figure 2.3.

When predicting toxicity for new cell lines, we observe that the results only vary with the genetic kernels used (see Figure 4.6). The same applies to the prediction of the toxicity of new compounds: the performance vary significantly only when the kernel for the chemical compounds is changed. This indicates that sharing information between variables of the problem (i.e. between cell lines or between chemicals) does not improve the performance, although such improvement has been observed in the past on other multitasks learning problems in bioinformatics [130, 13].

When turning to the more difficult task of predicting the toxicity of a new chemical compound for a new cell line, all the multitasks methods displayed very poor performances (see Figure 2.9 and Figure 2.10). Nevertheless, in this case, only the multitask approach is applicable, since predictions can only be made by sharing information between chemicals and between cell lines, since

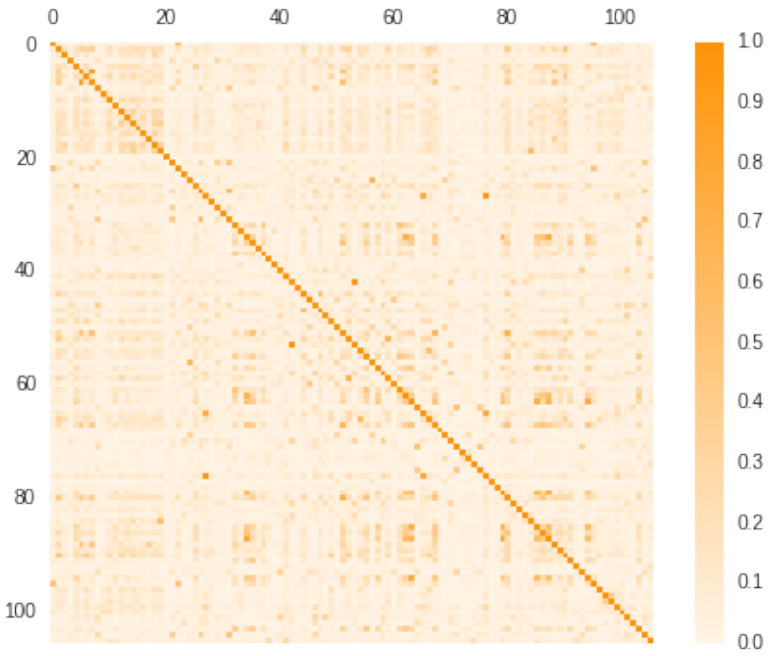


Figure 2.3 – Tanimoto kernel matrix between all chemicals using ECFP with circular substructures of length up to 9.

no information is previously available for the considered compound or cell line. As it was expected, the performance of predicting toxicity for new chemicals and new cell lines is worse than in the two previous tasks. Indeed, in the present case, we lack prior toxicity information both for the new cell lines and the new chemicals, while in the two other tasks, training sets are available for the cell lines or for the chemicals.

When predicting the toxicity of a known compound for a new cell line, we can observe that kernels display similar performances. While CI is slightly bigger for RNA-seq data than for SNP data (Figure 2.4), the RMSE is better when using SNPs, even though these differences do not seem to be significant (see Figure 2.4 and Figure 2.5).

In the case of predicting the toxicity of a new compound for known cell lines, the multitask approaches that include chemical kernels do not improve the

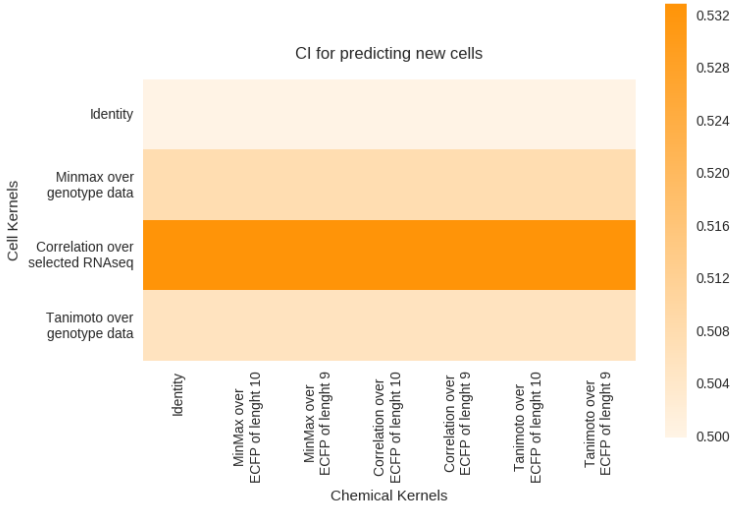


Figure 2.4 – Cross-validated CI for predicting the toxicity of a new untested cell line using different kernels. CI is calculated independently for every chemical and then the mean CI across all chemicals is reported. Cell lines kernels are displayed along the vertical axis and chemical kernels along the horizontal axis.

prediction performance over single task approaches based on cell line kernels only (see Figure 2.7). The MinMax kernel tends to give better results than the Tanimoto kernel and the Correlation kernel.

2.4 Discussion

A possible explanation of the generally poor performance of the multitask approach is that the chemical diversity was high in the chemical dataset. Considering the toxicity of very diverse chemicals to predict that of a given chemical can introduce errors. In other words, sharing information between tasks can decrease prediction performances if the tasks are too different. This could explain why the performance of the multitask approach was reduced in the two DREAM subchallenges, and did not improve over its single task counterparts when predicting toxicity for new cell lines.

To illustrate that the chemicals are indeed very diverse, we show the Tani-

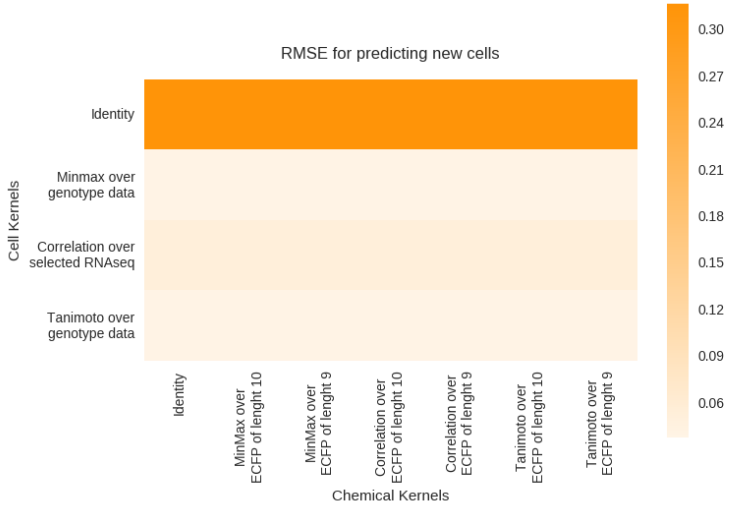


Figure 2.5 – Cross-validated RMSE for predicting a new cell line toxicity using different kernels. Cell lines kernels are presented along the vertical axis and chemical kernels along the horizontal axis.

moto kernel for substructures of length up to 9 in Figure 2.3. This figure shows that there is not much structure among the different chemicals, and that the kernel matrix is quite sparse. This implies that the model could not benefit from a multitask approach. One possibility would be to choose a less sparse kernel function, but this strategy might introduce similarities and relations between the different chemicals that are not real.

213 people from different countries competed in the challenge. All of them used different methods to predict cytotoxicity. General results of the DREAM toxicogenetic challenge [37] align with ours. Participants reported an overall poor predictive performance in both DREAM subchallenges. Nonetheless, results were better than random prediction. In Subchallenge 1, the ability to predict the individual variability is also found to be consistent with performances of similar methods to predict complex genetic traits such as height where each SNP contributes to explain a small part of the phenotype.

An interesting result observed in the challenge is that it is an easier task to

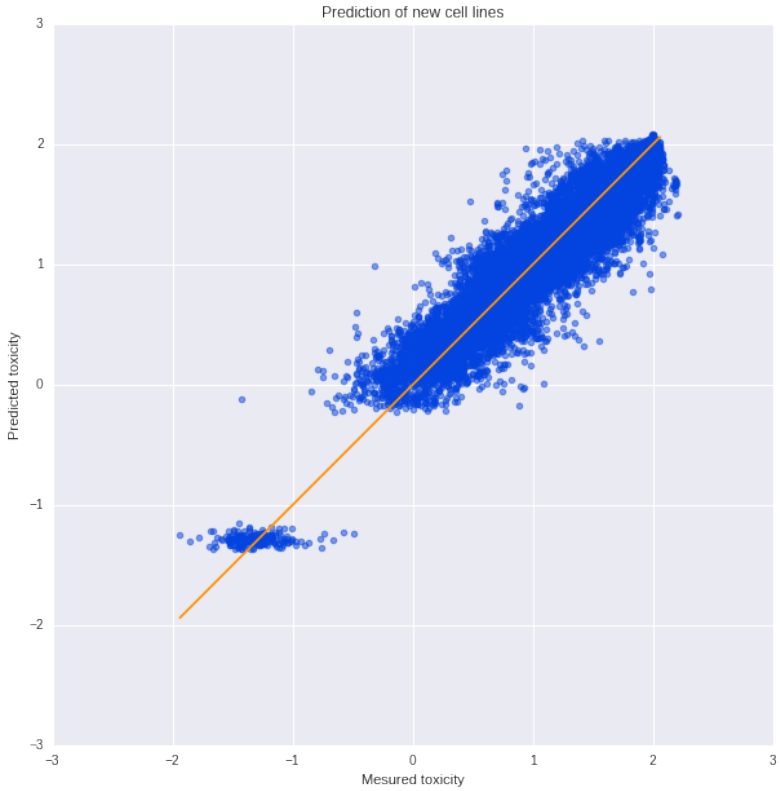


Figure 2.6 – For the model with best RMSE, predictions of new cell lines toxicity values (vertical axis) as a function of the measured value (horizontal axis). The MinMax kernel was used for cell lines, and a MinMax kernel with substructures of length 9 for the chemicals

classify the compounds as cytotoxic or non-cytotoxic than to predict or order their EC10 values. The criteria to divide the compounds between cytotoxic and non-cytotoxic used an EC10 threshold of 1.25 [37]. In this classification task, the participants obtained an AUC-ROC score above 0.9. This result is consistent with the behavior observed in Figure 2.6.

The second main output presented in [37] is that of an increased predictability in the cell lines for which RNAseq data was available. This appears to be in contradiction with the fact that we did not see any clear increase in performance with this type of data. One possible explanation is that in [37] the prediction was performed with the combination of both data types (RNAseq

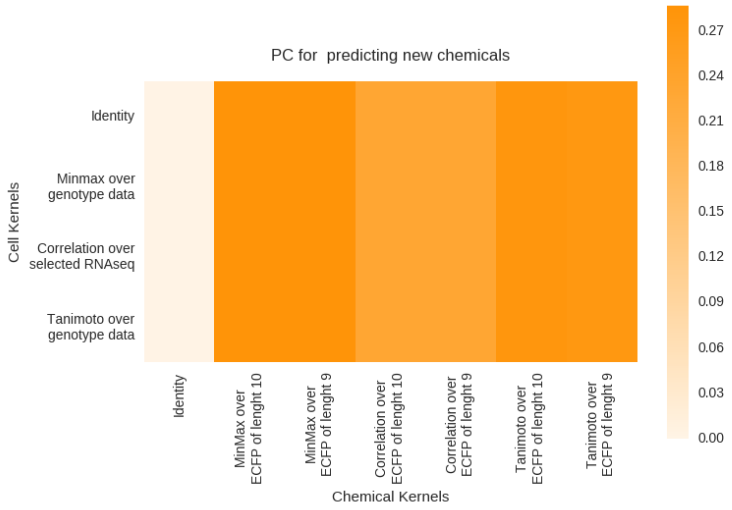


Figure 2.7 – Cross validated PC for predicting a new chemical compound toxicity using different kernels. Cell lines kernels are in the vertical axis and chemical kernels in the horizontal.

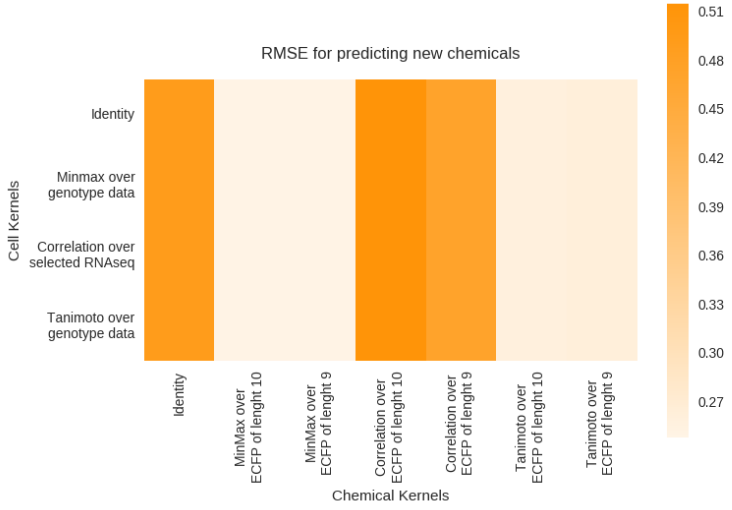


Figure 2.8 – Cross validated PC for predicting a new chemical compound toxicity using different kernels. Cell lines kernels are in the vertical axis and chemical kernels in the horizontal.

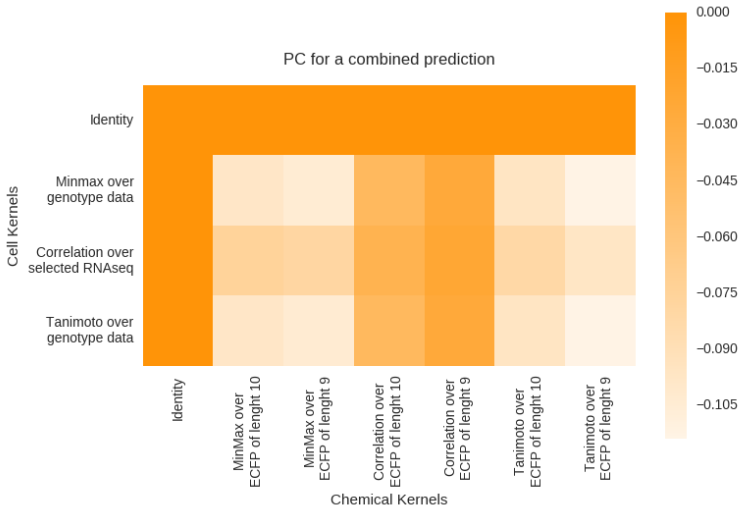


Figure 2.9 – Cross validated CI for predicting a new cell line and new chemicals toxicity using different kernels. Cell lines kernels are in the vertical axis and chemical kernels in the horizontal.

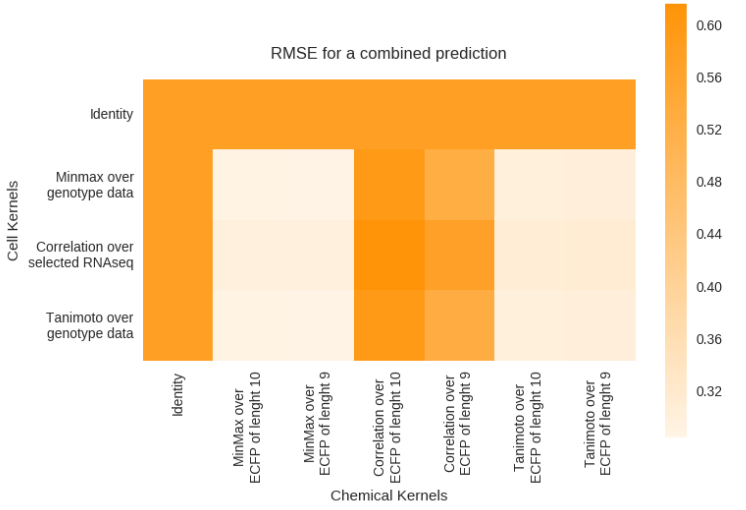


Figure 2.10 – Cross validated CI for predicting a new cell line and new chemicals toxicity using different kernels. Cell lines kernels are in the vertical axis and chemical kernels in the horizontal.

and SNPs) to predict those cell lines for which RNAseq data was available, while for the others cell lines only SNP data was used. In our experiments, we directly compare them. This shows that different types of data, when available, may explain different variability of the phenotype, and therefore, when combined, can increase the prediction performance.

It is worthy to mention again that we selected SNPs according to predictions of deleterious SNPs. Therefore, our results might have suffered from this assumption, particularly since a recent study [47] suggests that these tools have not been properly evaluated and might provide inaccurate predictions.

2.5 Conclusions

In this chapter, we have compared multitask and single task approaches, using GP, to predict the toxicity level of chemicals in cell lines. We have studied the performance in three scenarios defined by the presence or absence of toxicity information in the training set for the chemical or the cell lines presents in the test set, or both simultaneously.

We have shown that making an exact prediction of the toxicity level when predicting the toxicity of new chemicals in cell lines is a difficult problem, but we obtained a good correlation between prediction and measured values. This suggest it is possible to use machine learning to separate toxic and non-toxic compounds. We have shown that it is difficult to use the multitask approach for chemicals and one possible explanation is the high difference between the chemicals that were studied in this dataset. The problem of predicting for new chemicals and new cell lines was, as expected, an even more difficult problem, where the performance obtained was random or worse.

Future work devoted to better assess the interest of multitask approaches on the problem of toxicity prediction would require addressing an easier prob-

lem and to have access to a training dataset containing for more similar chemicals.

In the next chapter, we participate in a second DREAM challenge where we attempted to predict the effect of drugs on patients. This challenge will help us to understand how and whether we can use genotype data to better predict the effect of drugs in patients.

3 The Rheumathoid Arthritis Responder Challenge

French Abstract

La polyarthrite rhumatoïde est une maladie dégénérative inflammatoire chronique qui affecte les articulations synoviales. Dans les cas les plus sévères, elle est généralement traitée par des molécules qui suppriment l'activité du TNFα. Cependant, 30% des patients ne répondent pas à ce traitement. Ainsi, la possibilité de prédire la réponse des patients à un de ces traitements peut avoir un impact considérable sur le choix de la thérapie à prescrire.

C'est avec cet objectif en tête que DREAM a organisé un défi dont l'objectif était de prédire l'amélioration de l'état de santé d'un patient à partir de ses données génétiques. Nous avons participé à ce défi, dont l'objectif est conceptuellement équivalent à la prédiction d'effets secondaires indésirables.

Notre équipe est arrivée seconde dans la première phase, compétitive, de ce défi. Nous avons ensuite pris part à la deuxième phase, collaborative, en compagnie des autres équipes dont les résultats avaient été jugés suffisamment bons. Dans cette deuxième phase, nous avons cherché à déterminer si l'utilisation des données génétiques disponibles nous permettait d'améliorer les performances obtenues en utilisant simplement quelques indicateurs cliniques. Répondre à cette question est devenu un des principaux objectifs de cette deuxième phase, qui nous a menés à conclure que, conformément aux attentes des rhumatologues, l'information contenue dans les données de SNP ne permet pas d'améliorer significativement la performance prédictive de modèles qui utilisent uniquement des données cliniques usuelles.

Ces résultats ont été partiellement publiés dans [110].

English Abstract

Rheumatoid Arthritis is an autoimmune chronic inflammatory disorder affecting the synovial joints. In its more severe form, it is usually controlled by treatment with drugs that suppress the activity of TNFα. However, 30% of the patients do not respond to their treatment. The capability of predicting the response of patients to their medication can have a huge impact on choosing the right treatment for each patient. The DREAM initiative organized a challenge with this objective in mind. Because this problem is conceptually no different from predicted side effect, we took part in this challenge. The goal was to predict the improvement of the patients using genetic data.

Our team performed second in the first competitive phase. We then participated in a second collaborative phase, with other good performing teams. Here, we raised the question of whether we were improving the performance of the predictions when using simple clinical data and genetic information with respect to only using the clinical data. Answering this question became one of the main objectives of the second phase, which led us to conclude that, in agreement with the expectations of the rheumatology community, SNP information does not significantly improve predictive performance relative to standard clinical traits.

This work was partially published in [110].

3.1 Introduction

In the previous chapter, we tackled the problem of characterizing the toxicity of different chemicals in different cell lines. In this chapter, we will discuss the problem of predicting the effect of drugs in patients with *rheumatoid arthritis* (RA). Here, the range of considered chemicals is much smaller since we only

considered RA treatments. Another difference is that cell lines can be exposed to different chemicals, but patients will only receive one treatment that might be composed of one or more chemicals.

RA is an autoimmune chronic inflammatory disorder affecting the synovial joints; it can lead to substantial loss of functioning and mobility. RA is usually controlled by treatment with drugs that try to suppress the activity of TNFα, a transcription factor that plays a major role in the immunological response and in the inflammation pathway. However, 30% of the patients do not respond to anti-TNFα treatments [32]. Furthermore, no substantive methodology exists that can be used to identify anti-TNF non-responders before treatment [114].

In 2014, the DREAM initiative released the Rheumatoid Arthritis Responder Challenge. The objective of this challenge was to use genomic data to predict the response of patients to treatment. The challenge was divided in two different phases. The first phase was organized in a competitive manner, in which different teams were competing to obtain the best score predicting on a test set. The second phase consisted in a collaborative effort between the first phase best performing teams to improve the overall performance and get more insight into the problem. Based on the results of the first phase, the focus of the second phase became to determine whether the SNP information was contributing to the overall performance.

There were two tasks to solve. First, a regression problem consisting in predicting the improvement of disease level after the treatment, and second, to classify the patients as responder or non-responder to anti-TNF drugs. The level of the disease was measured using the absolute change in disease activity scores in 28 joints (DAS28) [92], and the categorical non-response as defined by the EULAR (European League Against Rheumatism) response criteria [119]. DAS28 uses the state of the joints and the erythrocyte sedimentation rate

to assess the actual disease activity, and is computed by a physician using a standard questionnaire. EULAR classifies a patient as responder if his DAS28 has significantly decrease and is under a certain threshold after the treatment. We participated as team Lucia and we focused on the first task (regression) during the first part of the challenge.

3.2 Data

The training data consist of $2,706$ individuals of European ancestry, compiled from 13 cohorts [33], of which 675 patients were used as leaderboard test set. All patients were required to have at least moderate disease activity score at baseline (DAS28$>$ 3.2).

Each patient was treated with one of six different treatments. The treatments consisted in the use of the drugs adalimumab, etanercept or infliximab, combined or not with methotrexate in cotherapy. The first three treatments are TNF-α inhibitors, but they differ in terms of the nature of the drug, and of their precise mode of action. Methotrexate is a DNA synthesis inhibitor used in various auto-immune diseases.

Clinical Data. Collected clinical data consisted in the gender and the age of the patient, the corresponding cohort and the batch from the experiment, the treatment that was prescribed, and the DAS before the treatment was initiated.

Outcome Data. A second evaluation of the DAS28 between 3 and 12 months after the treatment was initiated was also recorded. This was used as a basis to decide whether or not the patient was responding to the treatment. The condition of responder to treatment was assessed following the EULAR criteria.

Genotype Data. Genotype data were obtained using different methods for different batches. Therefore, the intersection of SNPs from the different batches was small (20,411 SNPs). Because of the small number of shared SNPs, data were combined, and missing SNPs were imputed separately for each batch, resulting in 2.5 millions of SNPs for each sample.

The final test set was derived from a subset of patients enrolled in the CORRONA CERTAIN study [86]. At the time of the challenge launch, 723 subjects had initiated anti-TNF therapy and had a 3 month follow-up visit. This test set contained some patients who have been treated with two new medications that were not present in the original dataset: golimumab and certolizumab. The 39 patients receiving golimumab were eventually excluded because predictions showed that participants were unable to successfully predict response in these subjects.

3.3 SNPs Selection

Due to the high dimensionality of the genotype data, we used two different feature selection methods. One of them was statistical and the other was based on biological knowledge.

Statistical feature selection We selected SNPs for each treatment individually by assessing the mutual information (MI) of each SNP with the response variable. MI is a well known concept in information theory [70] that measures how much the entropy of one variable is decreased when the other is known. In other words, the larger the mutual information, the more predictive power the variable will have. Entropy is a measure of uncertainty that is defined as

$$H(X) = \sum_{x \in A_X} P(x) \log \frac{1}{P(x)}, \tag{3.1}$$

where A_X is the set of possible events encoded by the random variable X. The conditional entropy of X of given Y is defined by

$$H(X|Y) = \sum_{y \in A_Y} P(y) \sum_{x \in A_X} P(x|y) \log \frac{1}{P(x|y)}, \tag{3.2}$$

which is the average over Y of the entropy of the conditional distribution of X given Y. Finally the mutual information can be calculated in the following way:

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X). \tag{3.3}$$

In our case, we calculated the mutual information between each SNP and the response level variable. If we assume X to be a discrete variable representing the SNP and Y a Gaussian variable representing the response level, we can calculate the mutual information between the two variables as in [90]

$$I(X,Y) = \frac{1}{2} \left(\log(\sigma_Y^2) - \sum_{x \in A_X} P(x) \log(\sigma_{Y|x}^2) \right), \tag{3.4}$$

where σ_Y^2 is the variance of Y. We use the MI to rank the SNPs [89, 52] for each treatment separately, we varied the number of top SNPs selected for each treatment: $k = 100$, 300, 500, 1000, 2000, and 3000.

Biological feature selection Another set of SNPs was selected by using biological knowledge. Since the considered drugs are TNF-α inhibitors, we selected SNPs related to TNF-α and RA pathways according to the KEGG Database [83, 62], genes that are targets of the 4 drugs used with the patients according to Drug Bank Database [66], and genes cited in various publications as modulators of the patients response to anti TNF-α treatments. From this reduced list of SNPs, only those inside exons were conserved. The final list contained 3840 SNPs.

3.4 Results

3.4.1 First phase

We chose SVR (see Section 1.3.2) to perform an initial approach to the problem, in order to explore the feature space.

We had to perform predictions for a total of 6 anti-TNFα treatments. Different strategies can be applied to predict response to 6 different RA treatments: learning one model per treatment, or ignoring treatment and learning one unique model for all the data. We decided to compare both approaches.

We used the two different feature selection strategies for the genomic data mentioned in Section 3.3. We compared models using the union of both sets of SNPs with models using just one of the two sets.

Each patient is represented by both genomic data and clinical data. We computed different kernels for each data type. For the genomic features, we used either the Linear or the MinMax Kernel. We calculated a clinical kernel by combining kernels computed for each of the clinical variables: for sex we used the Dirac kernel, and for age and initial DAS28 we used the linear kernel. We combined the different kernels using the mean of the kernels [11].

We submited the different models to the challenge leaderboard for evaluation. The performance was assessed using Pearson's correlation between the predicted values and the real values. We report the performance of the methods submitted to the leaderboard. In Figure 3.1 we present the results obtained using only the genetic data, and in Figure 3.2 the results obtained using both the genetic and clinical data. As we can see, there is a clear improvement from adding the clinical data to the model. Correlation improves from 0.16 in the best case using only genetic data to more than 0.45 when using also clinical data. Using a single model for all the treatments, and with only a small num-

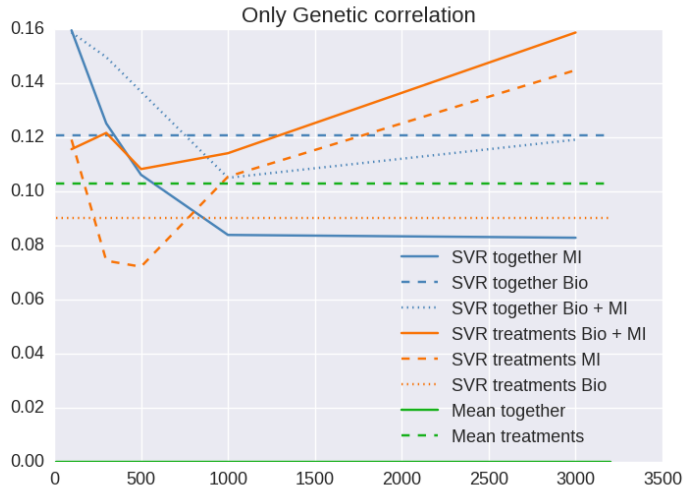


Figure 3.1 – Performance of our methods on the leaderboard of the DREAM challenge. Only SNPs data were used to learn the models. The plot shows the correlation of the predictions of the model with respect to the real response level (vertical axis) as a function of the number of MI SNPs used (horizontal axis). Methods that build a single model for all treatments are labelled 'together', and those corresponding to one model per treatment (performance averaged over the 6 treatments) are labelled 'treatment'. Methods including MI selected SNPs are labelled 'MI' and those including biologically selected SNPs are labelled 'Bio'. The models that do not include MI selected features have been plotted as an horizontal line to make comparisons easier. Those methods labelled with 'Mean' correspond to predicting the mean response of the training data.

ber of MI selected features, performs better than using a model per treatment. This situation is reversed when more features are added. This could be because we selected the SNPs based on the individual treatments. Using both types of selected SNPs (statistical and biological) always performs better than the baseline methods, which consisted in predicting the mean disease level in the training set; independently of the usage of a single predictor for all the treatments, or a predictor for each treatment. However, it is not clear if it is better to use a single predictor for all the treatments, with only a few SNPs selected, or use more SNPs and a different predictor for each treatment.

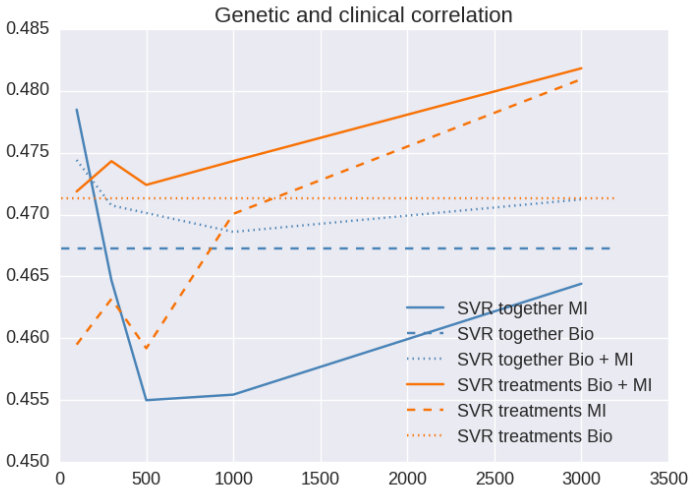


Figure 3.2 – Performance of our methods on the leaderboard of the DREAM challenge. Clinical data and SNPs were both used to learn the models. The plot shows the correlation of the predictions of the model with respect to the real response level (vertical axis) as a function of the number of MI SNPs used (horizontal axis). Methods that build a single model for all treatments are labelled 'together', and those corresponding to one model per treatment (performance averaged over the 6 treatments) are labelled 'treatment'. Methods including MI selected SNPs are labelled 'MI' and those including biologically selected SNPs are labelled 'Bio'. The models that do not include MI selected features have been plotted as an horizontal line to make comparisons easier.

Another member of our team also submitted some predictions based on Random Forests [16], using the biological SNPs and the top 500 SNPs selected using MI. It scored a correlation of 0.24 using genetic data and 0.46 using genetic and clinical data. According to the leaderboard, our results were far from the best, which scored a correlation of 0.37 using only genetic data, and 0.54 using also clinical data.

To avoid overfitting the first test dataset by submitting many predictions to the leaderboard, we decided to run a 10-fold cross-validation over the training set. We expected the models that produce better predictions in the cross-validation to be more robust, and less biased towards the initial test dataset.

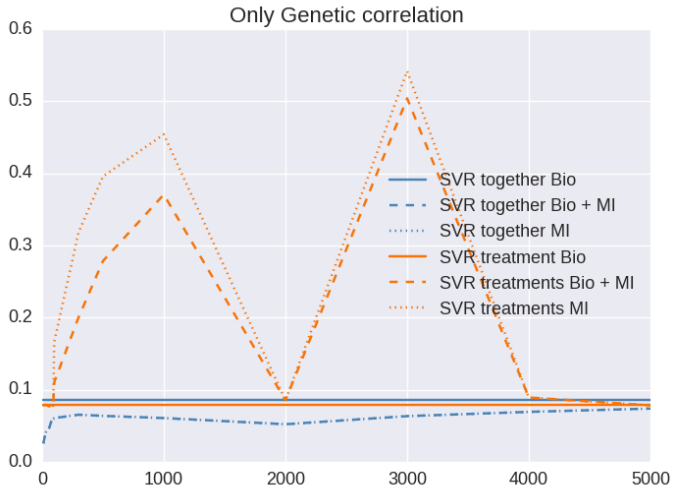


Figure 3.3 – Performance of our methods on a 10-fold cross-validation over the training data. Only SNPs data were used to learn the models. The plot shows the correlation of the predictions of the model with respect to the real response level (vertical axis) as a function of the number of MI SNPs used (horizontal axis). Methods that build a single model for all treatments are labelled 'together', and those corresponding to one model per treatment (performance averaged over the 6 treatments) are labelled 'treatment'. Methods including MI selected SNPs are labelled 'MI' and those including biologically selected SNPs are labelled 'Bio'. Those models that do not include MI selected features have been plotted as an horizontal axis to make comparisons easier.

The results in the cross-validations show two peaks of performance around 1000 and 3000 SNPs, as can be seen in Figure 3.3 and Figure 3.4. Surprisingly we obtain a dip in the performance when $k = 2000$. We did not find any explanation, but in both testing frameworks (leaderboard and 10-fold cross-validation) we obtain an increased performance when $k = 3000$. Overall, the results indicate that a few thousand SNPs are needed to capture relevant information.

Results obtained when including the clinical data are much better than those obtained using only the SNPs. A natural question that arises is whether the SNPs are bringing any additional information that helps make better pre-

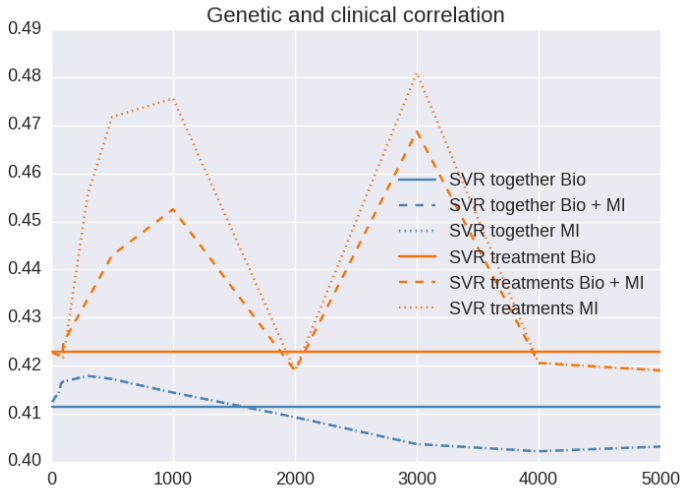


Figure 3.4 – Results obtained by our methods on a 10-fold cross-validation over the training data. Clinical data and SNPs were both used to learn the models. The plot shows the correlation of the predictions of the model with respect to the real response level (vertical axis) as a function of the number of MI SNPs used (horizontal axis). Methods that build a single model for all treatments are labelled 'together', and those corresponding to one model per treatment (performance averaged over the 6 treatments) are labelled 'treatment'. Methods including MI selected SNPs are labelled 'MI' and those including biologically selected SNPs are labelled 'Bio'. Those models that do not include MI selected features have been plotted as an horizontal axis to make comparisons easier.

dictions, or if it they just explain a fraction of the variability already covered by the clinical data. To answer this question, we ran again a 10-fold cross validation using the previous model which gave us the maximum performance, i.e. using 3000 selected SNPs for each treatment and the SNPs selected according to the bibliography. We used two different kernels for the SNPs, the linear kernel and the minmax kernel. We compared these approaches with an SVR that uses only the non-genetic data. The results are presented in Figure 3.5. It is clear that there is no difference between the performance of the different methods, showing that there is no gain from using SNP data on top of clinical data.

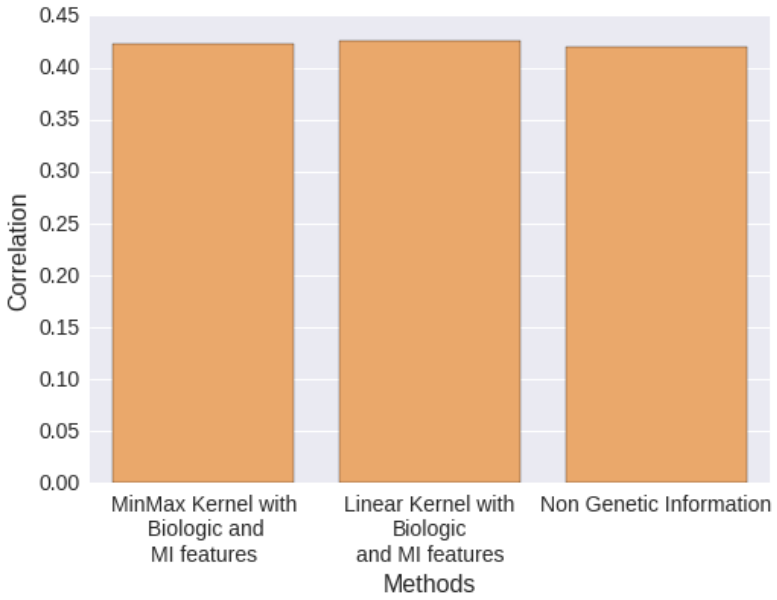


Figure 3.5 – Results obtained using Pearson correlation. The plots compare three models. The first and the second plots use SNP and clinical information while the third uses clinical data only. No significant difference was found.

Therefore, we made our final submission including only the clinical information. We obtained the second best score in the first subchallenge with a correlation of 0.38503. The top performing team obtained a correlation of 0.39307, which was significantly better according to a Wilcoxon signed-rang test of bootstraps with p-value$=2 \times 10^{-32}$.

3.4.2 Second phase

The second phase of the challenge consisted in attempting to improve the results obtained during the first phase. For that reason, all the teams that performed better than the baseline method used by the organizers during the first phase were invited to participate in a second collaborative phase. During this phase, teams came together to develop research questions and analytical strategies related to the ability to predict non-response to anti-TNF treatment. For this phase, we used the same approach described above and Support Vector

Machines for the classification problem.

During the first sessions of discussions we raised the question of comparing models using only clinical variables with models using both clinical and genetic variables. As explained above, we were not able to improve the performance of our model trained only clinical variables by incorporating genetic information. This indicates that response could be predicted using only simple clinical variables. The organizers decided to design experiments to systematically compare models using all variables with equivalent models using only the clinical variables.

Hence, each team submitted several predictions that were designed to assess to which degree genetic data were contributing to the models. Participants were asked to use a total of 102 different set of features in their models. The first submission used only clinical data. The second submission contained clinical data and the set of SNPs selected by the team. Finally, the participants submitted 100 predictions containing the clinical data and 100 different sets of randomly selected SNPs. The predictions were scored and analyzed across teams by the challenge organizers. The organizers asked for classifications and predictions using different sets of features: one prediction using our own selected SNPs, one that did not include any genetic features, and 100 sets of randomly selected SNPs.

All models using knowledge-mined SNP selection significantly outperformed models using random SNPs for AUPR, AUROC or both at a nominal p-value< 0.05 (one sided Kolmogorov-Smirnov test for enrichment of p-values vs. uniform p-values$= 4.2e-05$) (Figure 3.6). This suggested that for these models there was a non-zero contribution of genetic information to treatment effect. Furthermore, the best performing team (Outliers), used regularized method to select the best number of SNPs, and they found, using cross vali-

dation, that the optimal model was not selecting any.

We next compared the models incorporating the non-randomly selected SNPs to the clinical data and the models containing only the clinical data. Pairwise comparisons across models demonstrated no statistical differences (paired t-test p-value = 0.85 and 0.82 for classification AUPR and AUROC, respectively, and p-value = 0.65 for continuous prediction correlation, Figure 3.7), indicating that the contribution of SNP data to the prediction of treatment effect was not of sufficient magnitude to provide a detectable contribution to overall predictive performance.

The team that submitted the best model during this second face was the Outliers team. They studied a total of 160 SNPs extracted from pharmGKB database [116] and TNF related genes. They used a hierarchical regression model for predicting the DAS28 improvement after treatment. It combined a gamma distributed generalized linear model [35] and a Lasso regression. Our method did not perform so well in this phase and we obtained the second worst score when classifying between responders and non-responders. In the quantitative problem, we obtain the 4th best score when using the full model, and the 5th best when using only the clinical variables. Overall, the difference between the methods were relatively small.

3.5 Conclusions

The RA Responder DREAM challenge was designed to assess the ability to develop a clinically actionable predictor of response to treatment using common SNP variants. Thorough analysis by the different teams showed that current predictive algorithms are not able to produce such predictors. In fact, we were not able to detect any genetic contribution to predictions. This may reflect the complex nature of genetic contribution to complex phenotypes. Al-

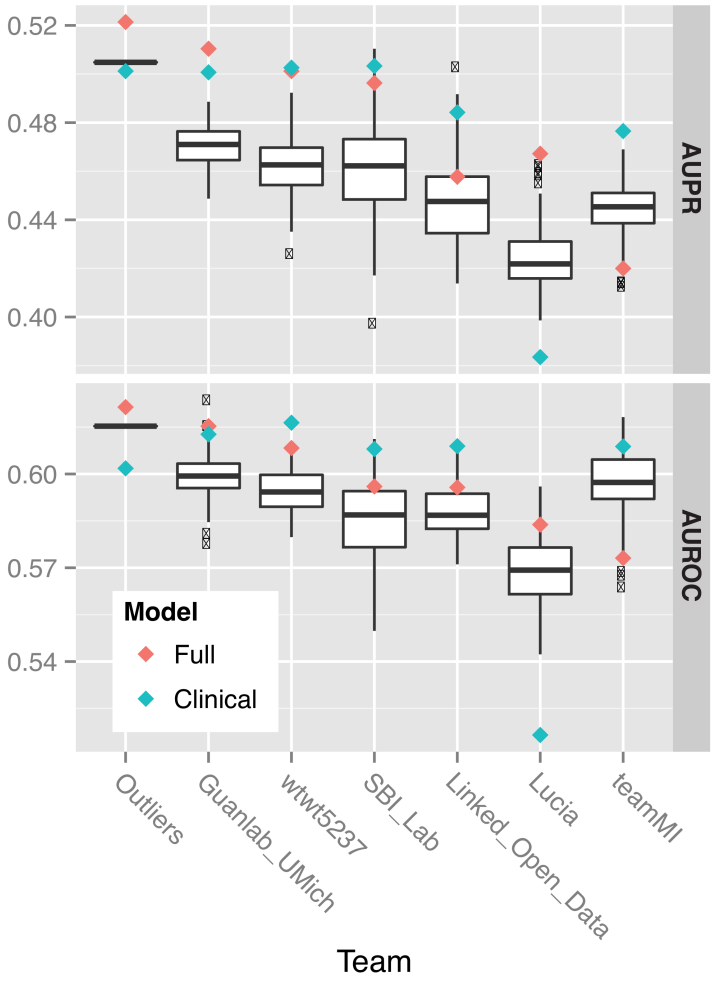


Figure 3.6 – Distributions of the models built with randomly sampled SNPs, by team, along with scores for their full model, containing data-driven SNP as well as clinical variable selection, (pink) and clinical model, which contains clinical variables but excludes SNP data (blue). For 5 of 7 teams, the full models are nominally significantly better relative to the random SNP models for AUPR, AUROC or both (enrichment p-value 4.2e-5).

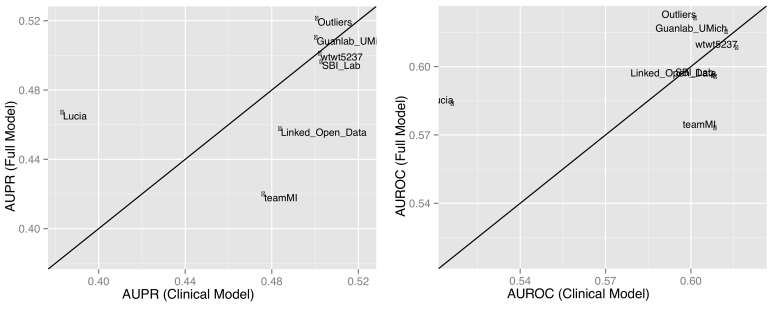


Figure 3.7 – AUPR and AUROC of each collaborative phase team's full model, containing SNP and clinical predictors, versus their clinical model, which does not consider SNP predictors. There was no significant difference in either metric between models developed in the presence or absence of genetic information (paired t-test p-value = 0.85, 0.82, for AUPR and AUROC, respectively).

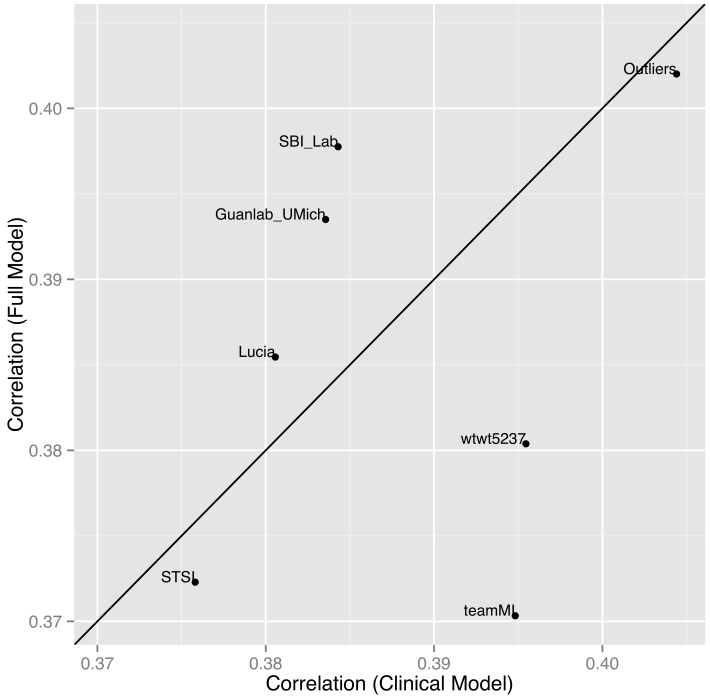


Figure 3.8 – Full model versus clinical model performance. Score (correlation with true values) of each team's collaborative phase full model, incorporating SNP and clinical data, versus their clinical model, which excludes SNP information, for the quantitative prediction subchallenge. There was no significant difference between full and clinical models (paired t-test p-value = 0.65).

though these genetic data did not provide a meaningful contribution to the predictions in the study, the methods used in the analysis were able to make use of the small set of available clinical features to develop a prediction that performed significantly better than random. These results suggest that future research efforts focused on the incorporation of a richer set of clinical information – including seropositivity, treatment compliance, and disease duration – may provide opportunity to leverage these methods in clinically meaningful ways. In addition, the identification of data modalities that are more effective than genetics in capturing heterogeneity in RA disease progression – whether clinical, molecular, or other – may also improve predictive performance.

We have shown that predicting the effect of drugs in patients is a complex problem, whether it is the main effect of the drug or a side effect. Although there are reasons to believe genetics play a role, we should be cautious when building models based on SNP data. As we have seen, clinical covariates, that are easier to measure can be very predictive and contain the same information. Moreover, working with SNPs introduces the problem of handling millions of features, many of which had to be imputed, while the number of samples is several orders of magnitude smaller,therefore introducing difficulties. Other types of genetic data might contain information about the treatment response. For example, gene expression might help to identify those genes involved in the treatment response. Finally, we also have seen that, by using different models for the different treatments, we can obtain better performances.

4 The Multiplicative Multitask Lasso with Task Descriptors

French Abstract

Nous avons observé dans les chapitres précédents que la prédiction personnalisée des effets secondaires indésirables se prête bien à l'analyse simultanée de plusieurs médicaments, ce qui justifie notre intérêt pour des approches multitâches. De plus, les données génétiques disponibles sont généralement peu abondantes (en termes de nombres d'échantillons), et l'identification de biomarqueurs génétiques explicatifs fait partie intégrante du problème. Cela suggère l'utilisation dans nos modèles de techniques de régularisation.

Nous avons aussi remarqué que l'on peut souvent disposer d'informations supplémentaires concernant les différents traitements que l'on étudie. Dans ce chapitre, nous proposons donc d'utiliser ces informations dans nos modèles, afin de formaliser notre intuition que deux tâches devraient avoir d'autant plus de biomarqueurs en commun qu'elles sont similaires. Nous notons l'absence de méthodes linéaires, régularisées et multitâches qui utiliseraient cette information, et proposons une nouvelle approche qui réponde à ces trois attentes. Nous montrons que le modèle que nous proposons a des performances compétitives par rapport à l'état de l'art, et a la capacité non seulement de faire des prédictions pour de nouvelles instances pour les tâches connues, mais aussi pour de nouvelles tâches qui leur sont liées.

Enfin, nous appliquons la méthode proposée à la prédiction de la liaison de peptides courts à différents allèles du complexe MHC-I. Ce problème a des applications importantes pour la mise au point de vaccins peptidiques.

Ce travail a été publié dans [10].

English Abstract

In the previous chapters, we have observed that the problem of predicting personalized drug effects ends itself well to working with multiple drugs at the same time, hence our interest in multitask approaches. Furthermore, available genetics data are generally scarce, and identifying explanatory genetic biomarkers is an important part of the problem. This suggests introducing regularization in our models. In this chapter, we therefore study regularized multitask machine learning techniques.

We also note that there often is additional infomation available about the different drugs and treatments. In this chapter, we propose to make use of this information in our models, to formalize our intuition that the more similar tasks are, the more biomarkers they should share. We identify the lack of linear, regularized, multitask methods that would use this information, and propose a new method that presents all three properties. We show that our proposed model is competitive with other state-of-the-art methods, and has the ability to make good predictions not only for new instances of the known tasks, but also for new related tasks.

Finally, we apply our proposed method to the prediction of the binding of small peptides and different alleles of the MHC-I. This is an important problem in the development of peptide vaccines.

This work was published in [10].

4.1 Introduction

A substantial limiting factor for many machine learning applications in bioinformatics is the scarcity of training data. This issue is particularly critical in precision medicine applications, which revolve around the analysis of consid-

erable amounts of high-throughput data, aiming at identifying the similarities between the genomes of patients who exhibit similar disease susceptibilities, prognoses, responses to treatment, or immune responses to vaccines. In the case of ADR prediction and treatment response prediction, patients do receive different treatments as we have seen in Chapter 3. Therefore, the data available for a single treatment is even scarcer. In such applications, collecting large numbers of samples is often costly. It is therefore frequent for the number of samples (n) to be orders of magnitudes smaller than the number of features (p) describing the data. Model estimation in such $n \ll p$ settings is a major challenge of modern statistics, and the risk of overfitting the training data is high.

Fortunately, it is often the case that datasets are available for several related but different problems (or tasks). While such data cannot be pooled together to form a single large data set because they are expected to be relevant to answer to different questions, the *multitask* framework makes it possible to leverage all the available information to learn related but separate models for each of these problems.

For example, genetic data may be available for patients who were included and followed under different but related conditions. If each condition is considered separately, we may not have enough data to detect the relevant genetic variations associated to the trait under study. The purpose of this chapter is to design a method that can leverage genetic data from the different conditions resulting in better models for each one of them.

Multitask learning approaches, where each condition corresponds to a task, can be used to circumvent this issue by increasing the number of learning examples while keeping the specificity of each dataset [93, 27]. Another prevalent strategy to avoid overfitting the training data is to apply *regularization*, that

is to say, to impose an l_1-norm over the weights assigned to the features to drive many of their weights to 0, as explained in Section 1.3.1. This, property makes these models suitable for biological applications, where it is often desirable for models to not only exhibit good predictive abilities, but also to be interpretable. For example, if samples are patients encoded by genetic features, if only a small number of features are selected by the model (i.e. are assigned non-zero weights), it may be possible to relate these features to the biological pathways involved in the predicted trait. Further down the line, these features can be used to aid diagnosis or design companion tests. However, l_1-regularized methods are sensitive to small perturbations of the data, and it is therefore necessary to pay attention to their stability.

MML and MMLD methods discussed in Section 1.3.3 are two common examples of multitask models that apply regularization. These approaches have two limitations. First, they cannot be directly applied to make predictions for new tasks for which no training data is available. This could be relevant to predict the cytotoxicity of a new drug on cells or patients, or to evaluate the prognosis of a previously unseen cancer subtype. Second, the degree of similarity between tasks is not explicitly considered. However, intuitively, we would like to explicitly enforce that more information should be shared between more similar tasks.

These two limitations can be addressed by defining an explicit representation of the tasks. This provides a convenient way to relate tasks and to share information between them, as is done in kernel methods [40, 15]. Based on the intuition that the second factor of the MML [69] should be similar for similar tasks, we propose to characterize each task by a set of descriptor variables and re-write this factor as a linear combination of these descriptor variables.

In this chapter, we introduce a new model to solve the problem stated in

1.3.3, give a result on the asymptotic convergence of the estimator, and present an algorithm for solving the optimization problem. Experimental results on simulated data show our approach to be competitive both in terms of prediction error and in terms of the quality of the selected features. Finally, we illustrate the validity of the proposed method for the prediction of new tasks by applying it to MHC-I binding prediction, a problem relevant to the design of personalized vaccines.

4.2 Multiplicative Multitask Lasso with Task Descriptors

The approaches presented in Section 1.3.3 do not explicitly model relations between tasks. However, an explicit representation of the tasks space might be available. Inspired by kernel approaches, where tasks similarities are encoded in the model [40, 15], we introduce a new model called Multiplicative Multitask Lasso with tasks Descriptors (MMLD), where we use a vector of tasks descriptor variables to encode each task, and to explain the specific effect modulating each feature for each task.

Following the MML formulation [69] presented in section 1.3.3, we decompose the parameter β into a product of two components. We keep the notation θ for the first component, which corresponds to the global feature importance common to all tasks. The second component is now a linear combination of the L-dimensional task descriptors $D \in \mathbb{R}^{L \times K}$. The L task descriptors must be defined beforehand and depend on the application. For example, if the different tasks are sensitivity to different drugs to which cell lines are exposed, one could use molecular fingerprints [43] to describe the drugs, i.e. the tasks, as done in Chapter 2. The regression parameter $\alpha \in \mathbb{R}^{p \times L}$ indicates the importance of each descriptor for each feature, and controls the specificity of each

task. Hence, we formulate the following optimization problem:

$$\min_{\theta\geq 0,\alpha\in\mathbb{R}^{p\times L}} \frac{1}{2}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\left(y_i^k - \sum_{j=1}^{p}\theta_j\left(\sum_{l=1}^{L}\alpha_{jl}d_l^k\right)x_{ij}^k\right)^2 + \lambda_1\sum_{j=1}^{p}|\theta_j| + \lambda_2\sum_{j=1}^{p}\sum_{l=1}^{L}|\alpha_{jl}| \quad (4.1)$$

where λ_1 and λ_2 are the regularization parameters for each component of β.

Predictions for a new data point x are made as $\sum_{j=1}^{p}\theta_j(\sum_{l=1}^{L}\alpha_{jl}d_l^k)x_{ij}$. This formulation allows to make predictions for tasks for which no training data is available: the only task-dependent parameters are the descriptors d_l^k. This ability to extrapolate to new tasks is not shared by the existing multitask Lasso methods. As we have seen in Chapters 2 and 3, this ability might be desirable to make new predictions for previously unseen chemicals or new drugs.

4.2.1 Theoretical guaranties

Let us define, for all $k = 1,\ldots,K$, $i = 1,\ldots,n_k$, $j = 1,\ldots,p$, $l = 1,\ldots,L$, $\xi_{ijl}^k = d_l^k x_{ij}^k$ and $\mu_{jl} = \theta_j\alpha_{jl}$. Problem 4.1 can be reformulated as

$$\min_{\theta\geq 0,\mu\in\mathbb{R}^{p\times L}} \frac{1}{2}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\left(y_i^k - \sum_{j=1}^{p}\sum_{l=1}^{L}\mu_{jl}\xi_{ijl}^k\right)^2 + \lambda_1\|\theta\|_1 + \lambda_2\sum_{j=1}^{p}\theta_j^{-1}\|\mu_j\|_1 \quad (4.2)$$

Following Lemma 1 in Ref. [122], it is immediate to prove that, when $\omega = 2\sqrt{\lambda_1\lambda_2}$, Problem 4.2 is equivalent to

$$\min_{\mu\in\mathbb{R}^{p\times L}} \frac{1}{2}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\left(y_i^k - \sum_{j=1}^{p}\sum_{l=1}^{L}\mu_{jl}\xi_{ijl}^k\right)^2 + \omega\sum_{j=1}^{p}\sqrt{\|\mu_j\|_1}, \quad (4.3)$$

with $\hat{\theta}_j = \sqrt{\frac{\lambda_1}{\lambda_2}\|\mu_j\|_1}$. Problem 4.3 has a convex loss function and a non-convex regularization term. The characterization of the asymptotic distribution of the estimator for this problem, as well as its $\sqrt{n}$-consistency, have been previously given by Lozano and Swirszcz [69], based on a more general result [100].

4.2.2 Algorithm

Problem 4.1 is non-convex. We therefore propose to adapt the algorithm of Ref. [69] and separate it in alternate convex optimization steps: the optimization of θ for a fixed α, corresponding to a nonnegative Garrote problem [17], and the optimization of α for a fixed θ, corresponding to a Lasso optimization [117]. Details can be found in Algorithm 4.2.1. Python code is available at: `https://github.com/vmolina/MultitaskDescriptor`

Algorithm 4.2.1.

Input $\{X^k, Y^k, D^k\}_{k=1,\dots,K}$, $\lambda_1, \lambda_2, \epsilon, m_{max}$.

Define $n = \sum_{k=1}^{K} n^k$, $\tilde{X} = \left\{x_1^1, \dots, x_{n^1}^1, x_1^2, \dots, x_{n^k}^k\right\}$ and $\tilde{Y} = \left\{y_1^1, \dots, y_{n^1}^1, y_1^2, \dots, y_{n^k}^k\right\}$

Initialize $\theta_j(0) = 1$ and $\alpha_j(0)$ according to an initial estimate, for j = 1,. . . ,p.

For $m = 1, \dots m_{max}$:

Solve for α:

$w_{ijl}(m) = \theta_j(m-1) d_{il} \tilde{x}_{ij}$.

$\alpha(m) = \arg\min_\alpha \frac{1}{2} \sum_{i=1}^{n} \left(\tilde{y}_i - \sum_{j=1}^{p} \sum_{l=1}^{L} \alpha_{jl} w_{ijl}(m)\right)^2 + \lambda_2 \sum_{j=1}^{p} \sum_{l=1}^{L} |\alpha_{jl}|$

Solve for θ:

$z_{*j}(m) = \left[\sum_{l=1}^{L} \alpha_{jl}(m) d_l^1 x_{1j}^1, \dots, \sum_{l=1}^{L} \alpha_{jl}(m) d_l^k x_{n^k j}^k\right]$, for $j = 1, \dots p$

$\theta(m) = \arg\min_{\theta \geq 0} \frac{1}{2} \sum_{i=1}^{n} \left(\tilde{y}_i - \sum_{j=1}^{p} \theta_j(m-1) z_{ij}(m)\right)^2 + \lambda_1 \sum_{j=1}^{p} |\theta_j(m-1)|$

$\beta_j^k(m) = \theta_j(m) \sum_{l=1}^{L} \alpha_{jl}(m) d_l^k$

If $R(\beta(m-1)) - R(\beta(m)) \leq \epsilon$ (where $R(\beta)$ denote the squared loss over all tasks)

Break

Return $\beta(m)$

4.3 Experiments on simulated data

In this section, we compare our method, the MMLD, to the models presented in Section 1.3.3: the ML, the MSL and the MML based on two different

criteria. First, we compare them in terms of the *quality* of the selected features. By quality, we mean the ability to recover the true support of β (that is, it corresponds to non-zero entries), as well as the stability of the feature selection upon data perturbation.

Second, we evaluate the methods in terms of prediction performance.

4.3.1 Simulated data

We simulate K design matrices $X_k \in \mathbb{R}^{n_k \times p}$ according to a Gaussian distribution with mean 0 and a precision matrix $\Sigma \sim Wishart(p + 20, I_p)$, where I_p is the identity matrix of dimension p. In our simulations $n_1 = n_2 = \ldots = n_K$. For each task k, we sample L descriptors d_l^k from a normal distribution with mean $\mu_{d_l} \sim \mathcal{N}(0, 5)$ and variance $\sigma_{d_l}^2 \sim \mathcal{G}amma(0.2, 1)$. We build θ by randomly selecting $p_s < p$ indices for non-zero coefficients, which we sample from a Gamma distribution $\mathcal{G}amma(1, 2)$. All other entries of θ are set to 0. We build α in the following manner: For each of the non-zero θ_j, we randomly select $L_s < L$ entries of α_j to be non-zero, and sample them from a Gaussian distribution $\mathcal{N}(0, 2)$. All other α_{jl} are set to 0.

We then compute $\beta_j^k = \theta_j \left(\sum_{l=1}^{L} \alpha_{jl} D_l^k \right)$ and normalize it by dividing by $\beta^* = \max_{j,k} |\beta_j^k|$. Finally, we randomly chose with replacement S_s entries of β_k. If the chosen entry is different from 0, we set it to 0; conversely, if it was equal to 0, we set it to a new value sampled from a Gaussian distribution $\mathcal{N}(0, 0.5)$. This last randomization step is performed to relax the structure of β. Finally, we simulate $Y = \beta X + \epsilon$ where ϵ is Gaussian noise with $\sigma^2 = 0.1$.

Each of our experiments consists in evaluating the different models in a 10-fold cross-validation. We create a first set of experiments containing 5 datasets generated with the parameters $K = 4$, $n_k = 100$, $p = 100$, $L = 10$, $p_s = 20$, $L_s = 4$, and $S_s = 100$. We generate a second set of experiments using $n_k = 20$

to simulate a scarce setting. We report the results of additional experiments in a scarcer setting ($p = 8000$, $n_k = 20$) in the section 4.3.4.

In each experiment we train 4 different models: the ML[82], the MSL[111], the MML[69], and the MMLD we propose here.

In order to better understand the role of the task descriptor space, we use 3 variants of the MMLD: one that uses the same task descriptors as those from which the data was generated; one that uses these descriptors, perturbed with Gaussian noise ($\sigma = 0.1$); and one with a random set of task descriptors, sampled from a uniform distribution over $[0, 1]$. Perturbing the task descriptors with more noise should give results in between those obtained in those last two scenarios.

Each of these 6 methods estimates a real-valued matrix $\hat{\beta} \in \mathbb{R}^{K \times p}$. We then consider as selected, for a given task k, the features j for which $\hat{\beta}_j^k$ is different from 0. For all methods, λ is set by cross-validation: Let $\lambda_{\min}$ be the value of λ that yields the lowest cross-validated RMSE $E_{\min}$. Then, we pick, among all $\lambda > \lambda_{\min}$ resulting in a cross-validated RMSE less than one standard deviation away from $E_{\min}$, the λ that yields the median cross-validated RMSE. This heuristic compromises between optimizing for RMSE and imposing more regularization.

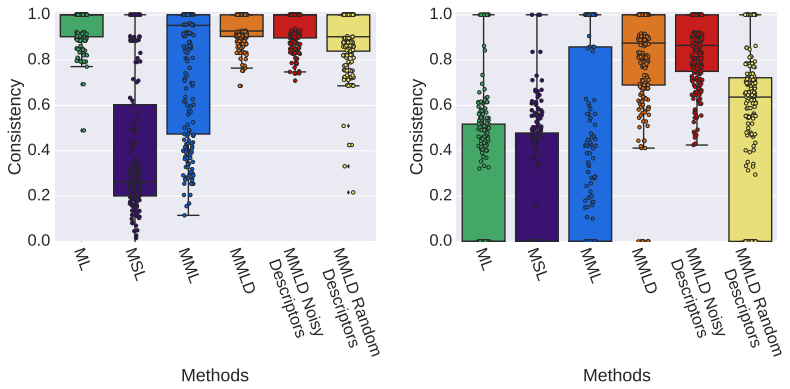
4.3.2 Feature selection and stability

In this section, we evaluate the ability of the feature selection procedure to select the correct features, as well as the stability of the procedure, on two sets of experiments.

Stability of the feature selection In precision medicine applications, it is often critical that feature selection methods be stable: If a method selects different features under small perturbations, we cannot rely on it to identify

biologically meaningful features. To evaluate the stability of the feature selection procedures, we calculate the consistency index [1] between the sets of features selected over each fold.

Figure 4.1a shows the consistencies of the different methods for the first set of experiments. We observe that the consistency of the feature selection for the proposed MMLD method is much higher than the consistency of MSL and MML. By contrast, ML presents a very high consistency index, that decays when the data is scarcer. (Figure 4.1b). The addition of small noise to the task descriptors does not have a strong effect on the stability of the selection, using random task descriptors negatively affects it, especially when data is scarce. In an even scarcer scenario the consistency presents high variation for all methods (Figure 4.7e).



(a) $n_k = 100$ instances per task (b) $n_k = 20$ instances per task

Figure 4.1 – Boxplot depiction of the consistency index of the different methods for simulated data.

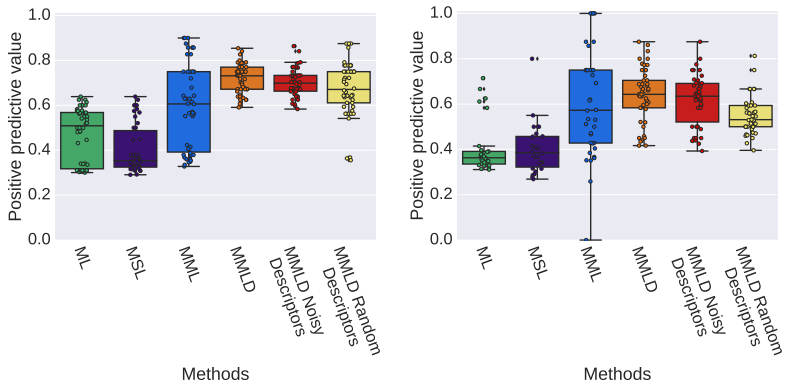
Number of selected features We report in Table 4.1 the mean number of non-zero coefficients assigned by each method in each scenario. We evaluate sparsity at the level of the β coefficients, hence the total number of coefficients is $n_k \times K$. The ML and the MSL both recover more features than all other

methods. The MML chooses more features than the MMLD when $n_k = 100$, but selects fewer parameters when the number of instances is reduced. Finally, the MMLD presents a much lower variation in the number of selected features than all other methods.

	True	ML	MSL	MML	MMLD	Noisy MMLD	Random MMLD
$n_k = 100$	126.8 ± 6.8	169.28 ± 163.4	231.62 ± 121.3	83.9 ± 80.7	54.88 ± 9.8	56.88 ± 11.2	49.12 ± 56.5
$n_k = 20$	126.8 ± 3.2	80.88 ± 79.8	43.96 ± 48.8	17.82 ± 21.7	46.24 ± 15.9	48.56 ± 18	46.72 ± 34.6

Table 4.1 – Mean number of non-zero coefficients assigned by each method.

Ability to select the correct features We report the Positive Predictive Value (PPV, Figure 4.2) and the sensitivity (Figure 4.3) of the feature selection for the different methods. The PPV is the proportion of selected features that are correct. The sensitivity is the proportion of correct features that are selected. Ideally, both numbers should be high.

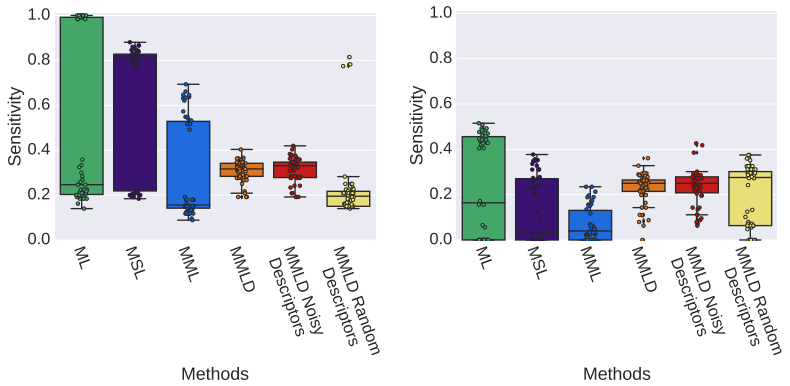


(a) $n_k = 100$ instances per task (b) $n_k = 20$ instances per task

Figure 4.2 – Boxplot depiction of the positive predictive value of the different methods for simulated data.

While the MML outperforms the ML and the MSL in terms of PPV (Figure 4.2), its sensitivity is worse (Figure 4.3). Indeed, the ML and the MSL select many more features: this higher sensitivity comes to the price of a large number of false positives. By contrast, the proposed MMLD performs well

according to both criteria. It clearly outperforms all other methods in terms of PPV (Figure 4.2), even when using noisy descriptors. In the case of random descriptors, the performance is close to that of the MML, and more degraded when the data are scarce. In terms of sensitivity (Figure 4.3), the MMLD also outperforms its competitors. We observe a higher variability in performance for these other methods, due to the higher variability in the number of features they select. The ML, MSL and MML suffer greater losses in sensitivity than the proposed method when data are scarce. Hence, using task descriptors seems to increase the robustness of the feature selection procedure. As would be expected, using random task descriptors negatively affects the ability of the MMLD to recover the correct features. Small perturbations of the task descriptors appear to have little effect on the quality of the selected features. We report similar results for the setting where $p = 8000$ (Supp. Mat.).

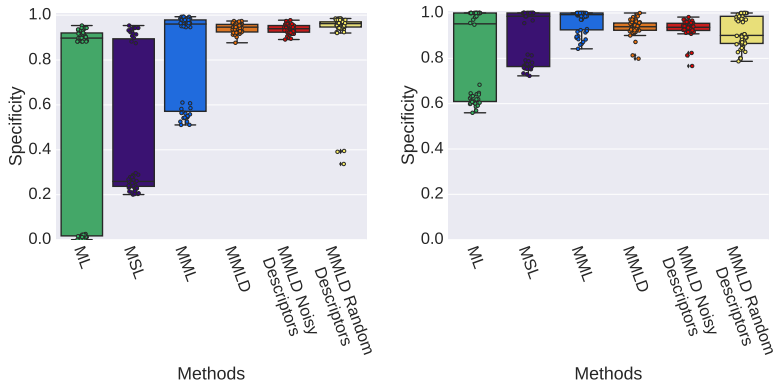


(a) $n_k = 100$ instances per task (b) $n_k = 20$ instances per task

Figure 4.3 – Boxplot depiction of the sensitivity of the different methods for simulated data.

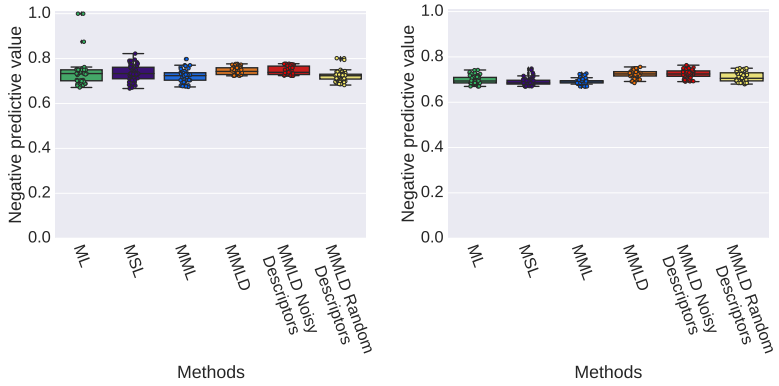
The results we obtain in terms of specificity (Figure 4.4) are consistent with our previous observations in terms of sensitivity and PPV. Once again, the high variability in the number of selected features explains the high variation in specificity of the ML, the MSL and the MML. The specificity of the proposed

MMLD is more stable, even for scarce data ($n_k = 20$, Figure 4.4b), except when using random task descriptors. All methods perform similarly in terms of NPV; because there are many more "negative" than "positive" features, differences in the number of correctly rejected features are not as noticeable. These results confirm the superiority of the MMLD in terms of the quality of the features it recovers.



(a) $n_k = 100$ instances per task (b) $n_k = 20$ instances per task

Figure 4.4 – Boxplot of the 10-fold cross-validated specificity of the different methods for simulated data.



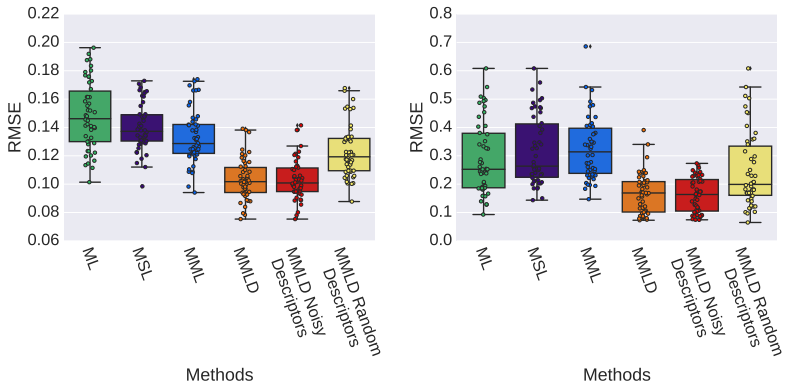
(a) $n_k = 100$ instances per task (b) $n_k = 20$ instances per task

Figure 4.5 – Boxplot of the 10-fold cross-validated negative predictive value of the different methods for simulated data.

4.3.3 Prediction error

We also propose to compare models based on the quality of their predictions. Figure 4.6 presents the 10-fold cross-validated Root Mean Squared Error (RMSE) of the different methods, for both $n_k = 20$ (Table 4.2) and $n_k = 100$ (Table 4.3). We observe that the proposed method performs better than its competitors, even with perturbed task descriptors. According to a paired Wilcoxon signed rank test, these differences in RMSE on scarce data are significant (Table 4.2). Interestingly, this is true even in comparison with the ML and the MSL, which select more features and could hence be expected to yield lower RMSEs.

This improvement in predictive performance is particularly visible in the scarce setting (Figure 4.6). In addition, the variance of the RMSE of the MMLD remains stable when the number of samples decreases, while it clearly increases for the other approaches.



(a) $n_k = 100$ instances per task (b) $n_k = 20$ instances per task

Figure 4.6 – Boxplot of the 10-fold cross-validated Root Mean Squared Error (RMSE) of the different methods for simulated data. For readability, (a) and (b) are plotted on different scales.

	ML	MSL	MML	MMLD	Noisy MMLD	Random MMLD
ML		-0.025 ± 0.077	-0.032 ± 0.076	$\mathbf{0.13 \pm 0.11}$	$\mathbf{0.13 \pm 0.11}$	$\mathbf{0.046 \pm 0.078}$
MSL	0.025 ± 0.077		-0.0069 ± 0.075	$\mathbf{0.15 \pm 0.11}$	$\mathbf{0.16 \pm 0.11}$	$\mathbf{0.071 \pm 0.088}$
MML	0.032 ± 0.076	0.0069 ± 0.075		$\mathbf{0.16 \pm 0.11}$	$\mathbf{0.16 \pm 0.1}$	$\mathbf{0.078 \pm 0.097}$
MMLD	$\mathbf{-0.13 \pm 0.11}$	$\mathbf{-0.15 \pm 0.11}$	$\mathbf{-0.16 \pm 0.11}$		0.0041 ± 0.037	$\mathbf{-0.08 \pm 0.097}$
Noisy MMLD	$\mathbf{-0.13 \pm 0.11}$	$\mathbf{-0.16 \pm 0.11}$	$\mathbf{-0.16 \pm 0.1}$	-0.0041 ± 0.037		$\mathbf{-0.084 \pm 0.1}$
Random MMLD	$\mathbf{-0.046 \pm 0.078}$	$\mathbf{-0.071 \pm 0.088}$	$\mathbf{0.078 \pm 0.097}$	$\mathbf{0.08 \pm 0.097}$	$\mathbf{0.084 \pm 0.1}$	

Table 4.2 – Difference in 10-fold cross-validated RMSE (mean and standard deviation) between the method in the row and the method in the column, on simulated data with $n_k = 20$. Differences that are significant according to a Wilcoxon signed rank test for a confidence interval of 0.99 are shown in bold.

	ML	MSL	MML	MMLD	Noisy MMLD	Random MMLD
ML		0.0085 ± 0.019	$\mathbf{0.016 \pm 0.019}$	$\mathbf{0.045 \pm 0.025}$	$\mathbf{0.045 \pm 0.025}$	$\mathbf{0.026 \pm 0.02}$
MSL	-0.0085 ± 0.019		0.007 ± 0.017	$\mathbf{0.036 \pm 0.018}$	$\mathbf{0.037 \pm 0.017}$	$\mathbf{0.017 \pm 0.016}$
MML	$\mathbf{-0.016 \pm 0.019}$	-0.007 ± 0.017		$\mathbf{0.029 \pm 0.019}$	$\mathbf{0.029 \pm 0.019}$	$\mathbf{0.01 \pm 0.014}$
MMLD	$\mathbf{-0.045 \pm 0.025}$	$\mathbf{-0.036 \pm 0.018}$	$\mathbf{-0.029 \pm 0.019}$		$7.3e-5 \pm 0.002$	$\mathbf{-0.019 \pm 0.015}$
Noisy MMLD	$\mathbf{-0.045 \pm 0.025}$	$\mathbf{-0.037 \pm 0.017}$	$\mathbf{-0.029 \pm 0.019}$	$-7.3e-5 \pm 0.002$		$\mathbf{-0.019 \pm 0.015}$
Random MMLD	$\mathbf{-0.026 \pm 0.02}$	$\mathbf{-0.017 \pm 0.016}$	$\mathbf{-0.01 \pm 0.014}$	$\mathbf{0.019 \pm 0.015}$	$\mathbf{0.019 \pm 0.015}$	

Table 4.3 – Difference in 10-fold cross-validated RMSE (mean and standard deviation) between the method in the row and the method in the column, on simulated data with $n_k = 100$. Differences that are significant according to a Wilcoxon signed rank test for a confidence interval of 0.99 are shown in bold.

4.3.4 Results for scarcer simulated data ($p/n = 400$)

In order to approximate the situations usually encountered in precision medicine, we simulated a scarcer case where $n_k = 20$ and $p = 8000$, all other parameters being the same as in the previous simulations. We report the consistency, specificity, sensitivity, PPV, NPV and RMSE for all methods in Figure 4.7, and the number of features they select across all tasks in Table 4.4. In addition, Table 4.5 reports the differences in RMSE between all methods.

Overall, we observe a degradation in performance for all methods, with respect to less scarce scenarios. In particular, all methods suffer from inconsistency when selecting features (Figure 4.7e).

As in less scarce settings, the MSL and the MML have lower consistencies than the proposed MMLD. The ML is more consistent in these experiments,

but with higher variability. In addition, it also presents a higher variation than other methods in sensitivity (Figure 4.7a) and specificity ((Figure 4.7b). Once again, this is related to the high variation in the number of features the ML selects (Table 4.4). When the task descriptors are not chosen at random, the MMLD performs the best in terms of those two measures. As in less scarce scenarios, the MMLD also outperforms all other methods in terms of PPV (Figure 4.7c). The differences in NPV between all methods are small (Figure 4.7d), due to the large number of features that are not included in the support of the data. Finally, the MMLD still outperforms all other methods in terms of RMSE (Figure 4.7f). This improvement is significant according to a Wilcoxon signed rank test with Bonferroni correction for multiple testing (Table 4.5).

True	ML	MSL	MML	MMLD	Noisy MMLD	Random MMLD
178.8 ± 1.6	455.7 ± 420.9	104.4 ± 121.9	23.4 ± 18.6	37.2 ± 13.9	37.9 ± 10.2	31.12 ± 26.3

Table 4.4 – Mean number of non-zero coefficients assigned by each method.

	ML	MSL	MML	MMLD	Noisy MMLD	Random MMLD
ML		-0.03 ± 0.069	-0.031 ± 0.078	$\mathbf{0.062 \pm 0.082}$	$\mathbf{0.064 \pm 0.074}$	0.026 ± 0.081
MSL	0.03 ± 0.069		-0.0013 ± 0.078	$\mathbf{0.092 \pm 0.09}$	$\mathbf{0.094 \pm 0.083}$	$\mathbf{0.055 \pm 0.082}$
MML	-0.031 ± 0.078	0.0013 ± 0.078		$\mathbf{0.093 \pm 0.078}$	$\mathbf{0.095 \pm 0.075}$	$\mathbf{0.057 \pm 0.081}$
MMLD	$\mathbf{-0.062 \pm 0.082}$	$\mathbf{-0.092 \pm 0.09}$	$\mathbf{-0.093 \pm 0.078}$		0.0021 ± 0.033	$\mathbf{-0.036 \pm 0.073}$
Noisy MMLD	$\mathbf{-0.064 \pm 0.074}$	$\mathbf{-0.094 \pm 0.083}$	$\mathbf{-0.095 \pm 0.075}$	-0.0021 ± 0.033		$\mathbf{-0.038 \pm 0.073}$
Random MMLD	-0.026 ± 0.081	$\mathbf{-0.055 \pm 0.082}$	$\mathbf{-0.057 \pm 0.081}$	$\mathbf{0.036 \pm 0.073}$	$\mathbf{0.038 \pm 0.073}$	

Table 4.5 – Difference in 10-fold cross-validated RMSE (mean and standard deviation) between the method in the row and the method in the column, on simulated data with $p = 8000$. The differences that are significant according to a Wilcoxon signed rank test for a confidence interval of 0.99 are shown in bold.

4.4 Peptide-MHC-I binding prediction

With the following experiment, we wanted to evaluate the predictive capabilities of our model in real data. We decided not to use the data from previous chapters for two different reasons. First of all, performance was not great for any method, specially in the experiments of Chapter 3. Second of all, we showed that the data used in Chapter 2 is not suitable for multitask models. For these reasons, we decided to test our method in data about the binding of small peptides to the different MHC-I (major histocompatibility complex class I) alleles. The prediction of whether a peptide can bind to a given MHC-I protein is an important tool for the development of peptide vaccines. MHC-I genes are highly polymorphic, and hence express proteins with diverse physicochemical properties across individuals. The binding affinity of a peptide is thus going to depend on the MHC-I allele expressed by the patient. It is therefore important that predictions are allele-specific. This in turn opens the door to administering patient-specific vaccines.

While some MHC-I alleles have been well studied, others have few if any known binders. Sharing information across different alleles has the potential to improve the predictive accuracy of models. Indeed, the multitask framework, where different tasks correspond to different MHC-I proteins, has been previously shown to be beneficial for this problem [55, 124]. In addition, it can be necessary in this context to make predictions for tasks (i.e. alleles) for which no training data is available.

4.4.1 Data

Following previous work[55], we test our model on three freely available benchmark datasets[48, 91]. The data consists of pairs of peptide sequences and MHC-I alleles, labeled as binding or non-binding. Ref. [48] provides two

datasets for the same 54 alleles, containing 1363 (resp. 282) positive and 1361 and (resp. 141784) negative examples. The dataset from Ref. [91] has 35 different alleles, 1548 positive examples and 4331 negative examples. As an example of an allele with few training data, allele B*57:01 in Ref. [91] only has 11 known binders.

The peptides are of length 9 and are classically represented by a 20-dimensional binary vector indicating which amino acid is present. While in this case $p < n$, this example allows us to evaluate the proposed method on real data, relevant for precision medicine applications. Because the MHC-I alleles are much longer than that, we do not adopt the same representation and define task descriptors as follows: Using sequences extracted from the IMGT/HLA database[99], we keep only the amino acids located at positions involved in the binding sites of all three HLA superfamilies[36, 55]. Inspired by the Linseq kernel [55], we then compute a similarity matrix between all alleles (tasks), based on the proportion of coincident amino acids at each position. We then perform a Principal Component Analysis on this matrix and keep the first 4 principal components, having observed that the structure of this matrix is not much perturbed by this dimensionality reduction. In the end, each task is represented by the 4-dimensional vector of its projections on each of these 4 components.

4.4.2 Experiments

We predict whether a peptide binds to a certain allele using the ML, the MSL, the MML and the MMLD. Additionally, we compare these approaches to single task Lasso regressions.

We run cross-validation using the same folds as in the original publications [48, 91]. The first Heckerman dataset [48] is divided in 5 folds and the

second one in 10. Because this second dataset is highly unbalanced, we randomly keep only one negative example for each of the positive examples. The Peters dataset [91] is divided in 5 folds. We run an inner cross-validation to set the regularization parameters.

We show in Figure 4.8 the Receiver Operator Curves (ROC) for the three datasets. Each curve corresponds to one fold. We additionally report the mean and standard deviation of the area under the ROC curve (ROC-AUC) for each approach. We observe that the ML, the MSL and the MMLD perform comparatively, and consistently outperform the two other methods.

Furthermore, we evaluate the ability of the different methods to predict binding for alleles for which no training data are available. For this purpose, we use the models previously trained on the folds of the two first datasets to predict on the folds of the third dataset. When predicting for a new task with the ML, the MSL and the MML, we use the mean of the predictions made by all trained models. As can be seen in Figure 4.9, the proposed method is the only one that outperforms the trivial baseline (ROC-AUC=0.5), hence illustrating its ability to make predictions for previously unseen tasks, by contrast with all other methods.

4.5 Conclusion

We have presented a novel approach for multitask least-squares regression. Our method extends the MML framework [69] to leverage task descriptors. This allows to tune how much information is shared between tasks according to their similarity, as well as to make predictions for new tasks. Multitask kernel methods [40, 55, 124] also allow to model relations between tasks, but do not offer the advantages of the Lasso framework in terms of sparsity and interpretability, which are key for biomedical applications.

The features it selects are hence more reliable, and the resulting models more easily interpreted. In addition, true support recovery suffers less in scarce settings. Finally, the predictivity of the resulting models is competitive with that of other Lasso approaches. Unsurprisingly, performance deteriorates when task descriptors are inappropriate. However, neither the quality of the selected features nor the model predictivity suffer much from the addition of small noise to these descriptors. These results suggest that the MMLD approach we propose is well adapted to precision medicine applications, which require building stable, interpretable models from $n \ll p$ data.

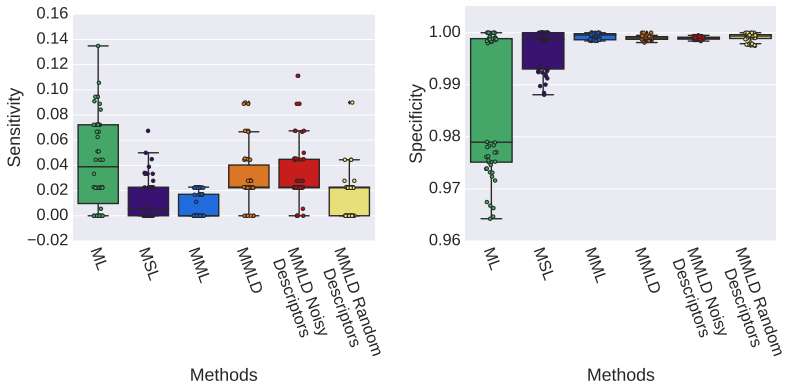
In terms of stability, our model shows a better consistency when $n = p$ but when p is increased all methods reduce its stability. This is a well-known problem in l_1 regularized methods, especially when dealing with correlated data. The lack of stability makes difficult a proper interpretation of the results obtained. In the next chapter, we should design a strategy to address this issue.

Finally, our experiments on MHC-I peptide binding prediction illustrate that the method we propose is well-suited to making predictions for tasks for which no training data is available.

These results were published in [10].

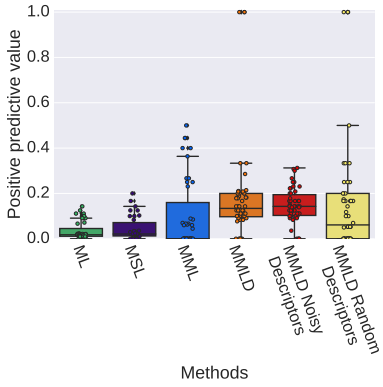
4.6 Code

Python code is available at `https://github.com/vmolina/MultitaskDescriptor`.

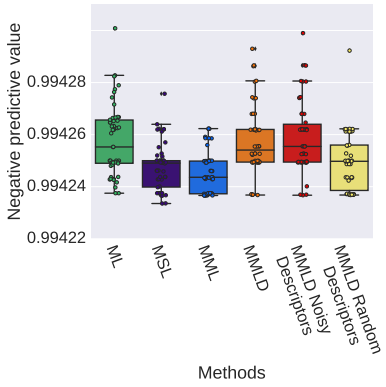


(a) Sensitivity.

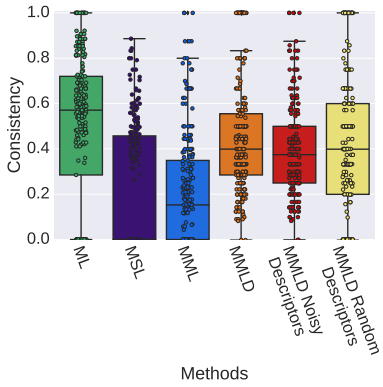
(b) Specificity.



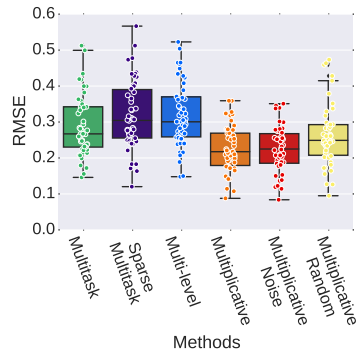
(c) Positive predictive value.



(d) Negative predictive value.

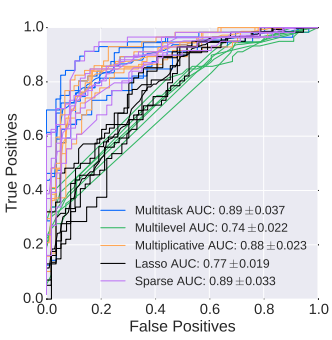


(e) Consistency.

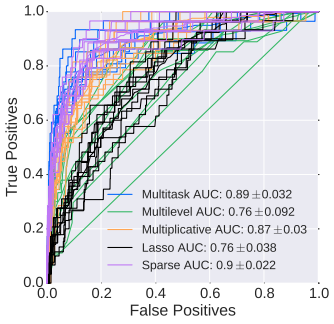


(f) Root mean square error.

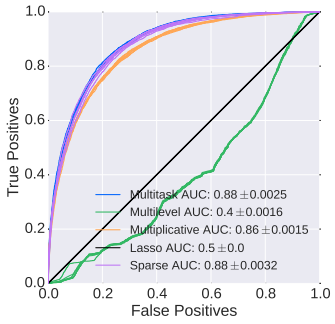
Figure 4.7 – Boxplots of the different performance measures for the 10-fold cross-validated experiments on simulated data with $p = 8000$.



(a) First Heckerman dataset

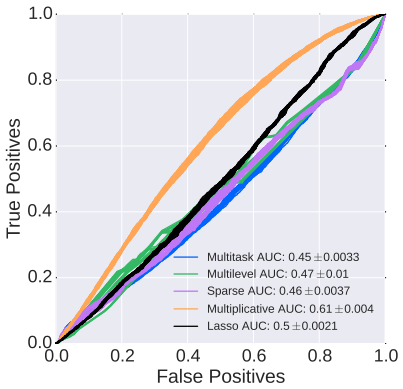


(b) Second Heckerman dataset

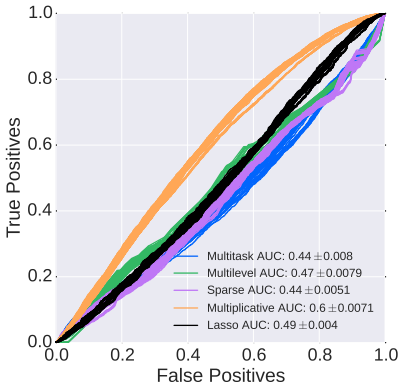


(c) Peters dataset

Figure 4.8 – Receive Operator Curves of the different methods in the different datasets. We show a line for each fold prediction. We report the mean and the standard deviation area under the curve for each method.



(a) Models trained on the first Heckerman dataset, evaluated on the Peters dataset



(b) Models trained on the second Heckerman dataset, evaluated on the Peters dataset

Figure 4.9 – ROC curves for the prediction of MHC-I binding, cross-dataset.

5 The Random Multiplicative Multitask Lasso with Task Descriptors

French Abstract

Dans le chapitre précédent, nous avons présenté le MMLD, une approche de régression multitâches qui utilise des descripteurs de tâches. Bien que le modèle que nous proposons aie de bonnes performances prédictives, il présente les inconvénients classiques des méthodes de régularisation $\ell 1$ quand la dimensionalité des données est élevée et que les variables sont fortement corrélées, ce qui est le cas des données génétiques. Cela se traduit généralement par une faible stabilité de la sélection de variable : ces méthodes sélectionnent des variables très différentes pour divers sous-ensembles des données. Cette instabilité rend l'interprétation des résultats obtenus avec notre méthode délicate.

Dans ce chapitre, nous présentons différentes techniques qui ont été proposées pour résoudre ce problème, et les adaptons à notre problème. Nous évaluons ces techniques sur des données synthétiques, avant de les utiliser pour analyser les facteurs génétiques impliqués dans des phénotypes liés au temps de floraison d'*Arabidopsis thaliana*. Ces techniques nous permettent d'obtenir une meilleure stabilité ainsi qu'une plus grande prédictivité que les modèles précédents.

English Abstract

In the last chapter, we presented the MMLD, a multitask regression model that makes use of task descriptors. Although it has good predictive performance, it suffers from known problems of ℓ_1-regularized methods when data are high-

dimensional and features highly correlated, which is the case for genetic data. This behaviour usually translates into the low stability of the feature selection, that is to say, these methods selects widely different sets of features when the data has been subjected to small perturbations. This unstability hinders the interpretation of the results of the model.

In this chapter, we explore different techniques that have been proposed to solve this problem, and adapt them to our model. We then evaluate these techniques on synthetic data. Finally, we use them to analyze the genetic background of traits related to the flowering time of *Arabidopsis thaliana*, showing improved stability and better prediction over previous models.

5.1 Introduction

In the previous chapter we have presented a new model that uses l_1 regularization to create interpretable predictive models. Even though we have shown that it has good predictive performance, it exhibits some limitations when dealing with feature selection in highly correlated datasets. Indeed, the l_1 penalty sets most features to zero while selecting as non-zero those that are relevant for the prediction. While this makes it very interesting for feature selection, it has two drawbacks that are important in our setting, that is (1) its behavior with high-dimensional data and (2) its behavior in presence of correlated features [134]:

1. In the $p > n$ case, the Lasso selects at most n variables before it saturates, because of the nature of the convex optimization problem. This seems to be a limiting feature for a variable selection method since n might be smaller than the number of relevant features.

2. Whenever there is a group of features with very high pairwise correlation,

the l_1 regularization tends to select only one variable from the group and does not care which one is selected. This behaviour might induce several limitations for interpretation, like not selecting the relevant feature in the group, or not selecting the whole group when all the variables in it are relevant. This usually leads to *unstable* results, meaning that running the algorithm several times, or several times on slightly different subsets of the data, will return very different sets of selected features.

Furthermore, [73] shows that in high dimension, the Lasso feature selection is inconsistent (meaning that it fails to recover the correct features even with infinite sample size) when the regularization parameter is optimal for prediction.

These drawbacks are important in the bioinformatics setting, where problems usually present a very limited number of instances (usually a few hundreds or thousands), while the number of variables can rise to the order of a few milions. In addition, these variables are usually highly correlated. In this chapter, we will show an example of such a situation in the case of the plant *Arabidopsis thaliana*. We will study its flowering time, which is a complex biological phenotype that might involve whole biological pathways involving many different genes. The corresponding DNA sequences will contain many SNPs and will tend to contain correlated SNPs. Indeed, neighbouring SNPs are likely to be mutated together due to linkage disequilibrium. Furthermore, SNPs might be in linkage disequilibrium with the causal SNP that might not be present in the data, making it more difficult to make sense of the result.

5.2 Approaches in the single task framework

Many different approaches have been proposed to deal with these problems and try to make l_1 regularized models more consistent [134, 135, 121]. The most popular choice to solve this problem is the elastic net approach [134]. The authors proposed a mixed regularization (Equation 5.1)

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \tag{5.1}$$

where the first term corresponds to the loss function, the second term is the l_1 regularization over the weights of the regression, and the last term is an l_2 regularization. The l_2 regularization also shrinks the parameters towards zero, but it will associate similar values to correlated features. On the other hand, the fact that these values are similar implies that correlated features are assigned coefficients with the same sign, which might not match biological reality.

Instead of adding another regularization term, the Adaptive Lasso [135] adds weights to the Lasso penalties of the model according to an ordinary least square regression (i.e. without regularization):

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j| \tag{5.2}$$

where $w_j = |\beta_j^{ols}|^{-r}$ with $r > 0$ and β_j^{ols} is the ordinary least square estimator. The Adaptive Lasso has nice asymptotic properties when p is fixed and n tends towards infinity. The authors show that, in this setting, the Adaptive Lasso recovers the true underlying model with probability tending to 1. This property is referred to as the oracle property. However, when the data are limited and the OLS estimates are unstable, as it is the case in our setting, the Adaptive Lasso presents worse performance than the original Lasso.

To overcome this problem, the authors of [121] presented a new approach known as the Random Lasso. The method presents similarities the the Random Forest [16] where many classification and regression trees (CART) [18] are learned from different bootstrap samples and a final prediction is made as the mean prediction of the different CART.

The algorithm is divided in two parts. First, the importance of each predictor variable is calculated according to the weights given in the different bootstrap samples by the model. Here, a bootstrap sample consists in generating a new dataset by sampling with replacement the same number of instances from the original dataset, and sampling according to a uniform distribution a preset number of features. A second round of bootstrap iterations is used to calculate the final model; again, we sample with replacement as many instances as in the original data set, but now the features are sampled according to the importance calculated in the previous step. The elevated number of repetitions makes that highly correlated features are selected together and allows to select more than n variables. The main drawback of this approach is its high computational demand, but it can be overcome by parallelizing the computation of the repetitions.

Furthermore, in [74] the authors study the stability selection problem, and they propose a solution for the case of the Lasso algorithm. They called this method the Randomized Lasso. It also consists in solving many times several Lasso problems. In this case, the method uses only a set of repetitions. Also, for each repetition and for each feature a regularization parameter is sampled. The output of the method is a set of features that are selected with probability higher to a given threshold. This method does not produce a predictive method; therefore, a second predictive method must be trained.

5.3 Random MMLD and Randomized MMLD

Despite the high computational demand of the model, we decided to adapt the Random Lasso to the MMLD model proposed in Chapter 4. Algorithm 5.3.1 shows the Random Lasso algorithm adapted for the MMLD. We kept the structure of the original algorithm: We first train a set of models according to different bootstrap samples and we calculate the importance of each variable.

In the second step, we sample variables according to their importance and we train a second set of models from which we derive the final model. The main change is how we calculate the importance of variables in our model. We calculate each of the β_j^k coefficients and compute their mean across the tasks. Another option would have been to select the maximum β_j^k across the tasks. However, by selecting the mean, we prioritize variables that are important across the tasks over variables that might be important for just one of the tasks.

Algorithm 5.3.1.

Input $\{X^k, Y^k, D^k\}_{k=1,\dots,K}$, $B \in \mathbf{Z}$, $q_1 \in \mathbf{Z}$, $q_2 \in \mathbf{Z}$.

Step 1 We generate importance measures for all samples.

Draw B bootstraps data sets by sampling with replacement n instances from the original dataset.

For each sample $b_1 \in \{1, \dots, B\}$, randomly select q_1 candidate variables.

Learn a MMLD model to obtain the coefficients

$\beta^{(b_1)} = \theta^{(b_1)}\gamma^{(b_1)}$, where $\beta^{(b_1)} \in \mathbf{R}^{K \times q_1}$.

Calculate the importance measure for feature j as

$I_j = B^{-1}K^{-1}|\sum_{b_1=1}^{B}\sum_{k=1}^{K}\beta_j^{k,(b_1)}|$

Step 2 We calculate the final estimators and select the variables.

Draw B bootstraps data sets by sampling with replacement n instances from the original dataset.

For each sample $b_2 \in \{1, \ldots, B\}$, randomly select q_2 candidate variables according to their importance.

Learn a MMLD model to obtain the parameters $\theta^{(b_2)}$ and $\gamma^{(b_2)}$ according to Algorithm 4.2.1.

Compute the final parameters θ and γ as the mean of the parameters estimated in Step 2.

We also adapted the concept of stability selection [74] to the MMLD model. We call the resulting algorithm the Randomized MMLD (Algorithm 5.3.2). As opposed to the Random MMLD, this model does not sample the features at each iteration and only requires one pass over the data, but as a drawback, it requires setting a threshold to define which variables are accepted as selected.

Algorithm 5.3.2.

Input $\{X^k, Y^k, D^k\}_{k=1,\ldots,K}$, $B \in \mathbf{Z}$, $\alpha \in (0, 1]$, $\tau \in [0, 1]$.

Step 1 We solve an MMLD many times.

Draw B bootstraps data sets by sampling with replacement n instances from the original dataset.

For each sample $b_1 \in \{1, \ldots, B\}$, randomly sample a weight vector W of size p from a uniform distribution $\mathcal{U}(\alpha, \infty)$.

Scale the feature of X^k by W for $k = 1, \ldots, K$.

Learn an MMLD model to obtain the coefficients $\beta^{(b_1)} = \theta^{(b_1)}\gamma^{(b_1)}$, where $\beta^{(b_1)} \in \mathbf{R}^{K \times q_1}$.

Keep the features that have been selected.

Step 2 Get the final set of variables.

Select those variables that are selected with probability larger than the threshold τ.

Following, we test these algorithms on a synthetic dataset, and use them to study the genetic basis of flowering traits in *Arabidopsis thaliana*.

5.4 Experiments on synthetic data

As we want to evaluate the stability of the two methods that we proposed in this chapter (Random MMLD and Randomized MMLD), and compare them to the original MMLD, we create 5 synthetic datasets similarly as we did in Section 4.3.1.

In this chapter, we are interested in highly correlated data and the problems they pose. For this reason, we start by generating a correlation matrix with some blocks of highly correlated data. To generate the data, we sample a first precision matrix from a Wishart distribution with degree of freedom equal to $p + 20$ and a second precision matrix from a Wishart distribution with degree of freedom equal to p. The second matrix corresponds to highly correlated features. We then, randomly separate the features in 100 clusters and substitute the corresponding values from the first precision matrix for the corresponding values from the second matrix. We used this matrix as the covariance of our data X. The rest of the procedure is the same as in the previous chapter. We sample $n_k = 20$ instances for each task. Each instance is a $p = 2000$ dimensional vector, we sample a total of $K = 4$ tasks and $L = 4$ descriptors.

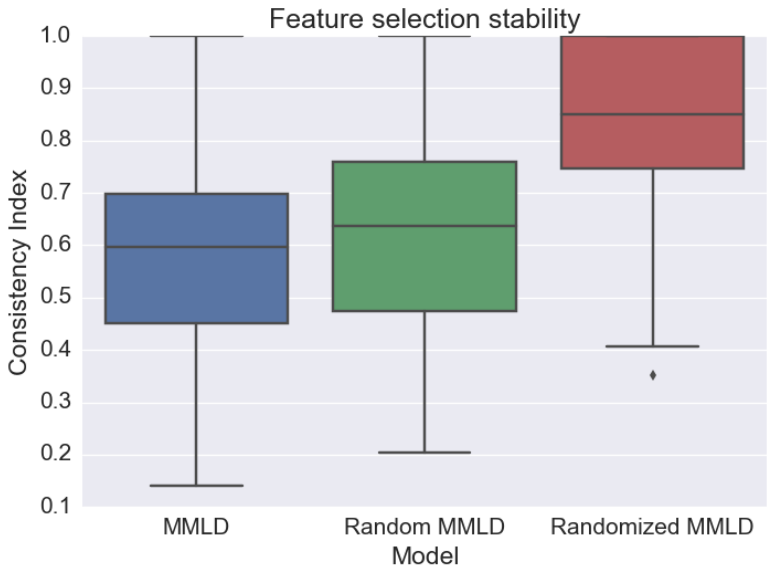
To evaluate the stability of the feature selection we run a 10-fold cross validation on each one of the datasets. For selecting the hyperparameters of the models we run an inner 3-fold cross-validation. We select the parameters that minimize the prediction error, but we only consider those models that select at least one variable across the 3 folds, and for which the mean number of selected variables is at most 1% of the variables. If no model fulfils these

conditions, we increase the maximum mean number of selected variables by 1%.

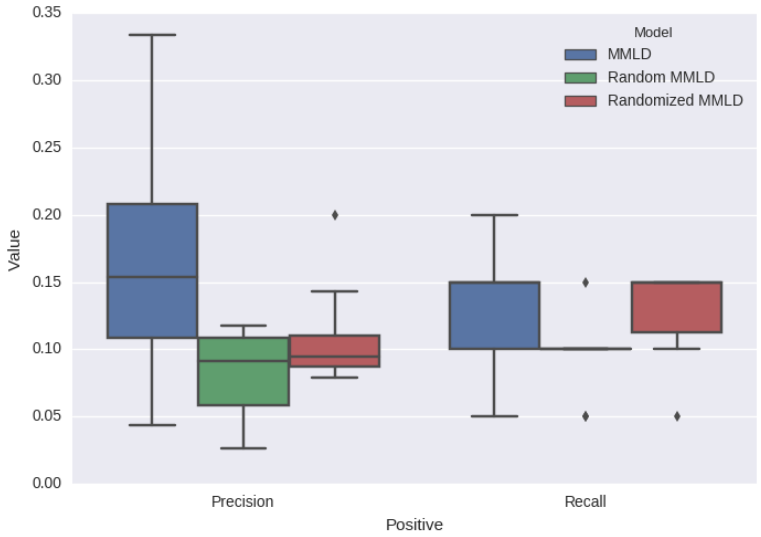
We start by testing the stability of the feature selection. To evaluate it we use, as we did in the previous chapter, Kuncheva's Consistency index [1]. As we can observe in Figure 5.1a, Random MMLD does not perform significantly better than MMLD (p-value= 0.039 in a dependent t-test for paired samples), while the Randomized MMLD outperforms both the original MMLD and the Random MMLD (p-values of $1e-33$ and $6e-24$ respectively).

While the Randomized MMLD has improved the stability of the feature selection, we observe that the precision of the selection is worse than that of MMLD in absolute terms, but with a lower variance. In the case of Recall, we get slightly better results for the Randomized MMLD. In the case of Random MMLD, the results are clearly worse. Worse precision and better recall and stability can be explained by selecting more features, but there's no escaping this: making these approaches more stable requires selecting correlated features together, instead of picking only one of them.

Finally, we did a second experiment to better understand the bad performance of the Random MMLD when compared with the Randomized MMLD. We used the same 5 datasets with the same 10-fold cross-validations as in the previous experiment. In this case, we test the three methods with multiple parameters and we check the distribution of the correlation index for each set of hyperparameters. Figure 5.2 shows the cumulative distribution of the performance of the three models. Here we observe that both Random MMLD and Randomized MMLD have a long tail, i.e., there are more combinations of parameters for which the model has a high consistency. In the case of MMLD, we observed that the maximum consistency obtained is 0.6. The tail of Randomized MMLD is in fact better than that of Random MMLD.



(a)



(b) W

Figure 5.1 – We show the feature selection performance of three different methods, the MMLD, the Random MMLD, and the Randomized MMLD. The models are evaluated using 10-fold cross-validations over 5 synthetic datasets. Figure 5.1a shows the stability of the feature selection, the measure used is the Consistency Index. Figure 5.1b shows the performance according to precision and recall.

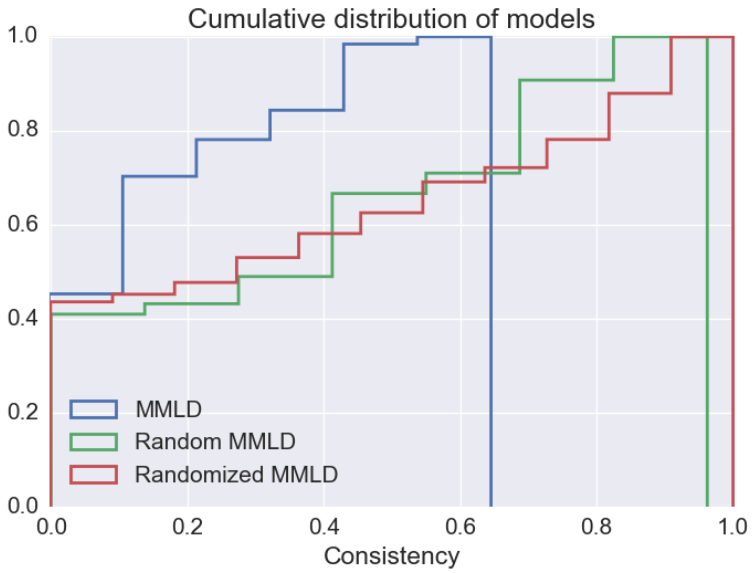


Figure 5.2 – We show the cumulative distribution of the Consistency Index for 3 different methods. Each data point corresponds to the mean performance across the datasets of a single model with fixed hyperparameters.

In summary, we observe that while both proposed methods show to have potential to increase the stability of the feature selection, selecting the correct parameters is harder with Random MMLD in practice. Furthermore, the multiple rounds of iterations make it longer to run than the Randomized version. Even if the number of features sampled is reduced, the number of iterations should be increased to compensate, which does not improve overall runtime. Therefore, in the next section we only use the Randomized MMLD and the MMLD to analyse a dataset about *Arabidopsis thaliana* flowering time.

5.5 *Arabidopsis thaliana* experiments

To show the performance of the method on real datasets, we studied *Arabidopsis thaliana* flowering times [7]. This dataset contains genotypic and phenotypic measurements for different plants that grew in different conditions. The

Task name	Task description	Number of instances
LDV	18°C, 16hrs daylight, vernalized (5wks, 4°C)	168
LN10	10°C, 16hrs daylight	177
LN16	16°C, 16hrs daylight	176
LN22	22°C, 16hrs daylight	176
SDV	18°C, 8hrs daylight	162
SD	18°C, 8hrs daylight, vernalized (5wks, 4°C)	159

Table 5.1 – Number of instances presents in each task for the *Arabidopsis thaliana* dataset. Tasks names are the same as used in [7].

dataset contains a total of 199 lines and a total of 214 051 SNPs. We reduced the dataset to plants that have been grown in controlled conditions in a greenhouse. We used the different conditions (temperature, sunlight exposure, natural or artificial light, and the number of weeks of vernalization) as task descriptors.

The final dataset contained $n = 1\,018$ non-unique instances belonging to $K = 6$ different tasks with a total of $L = 5$ task descriptor variables. The number of instances available for each task can be observed in Table 5.1

We performed a 10-fold cross-validation to evaluate the feature selection procedure according to its consistency and to the predictivity of the models. For each fold, we trained both the Randomized MMLD and a standard MMLD. We also used a single task Lasso for each task for comparison. As a ground truth we use a list of genes that has been associated with the flowering time of *Arabidopsis thaliana* [107]. This list contains a total of 164 genes. We consider any SNP located inside one of these genes to be a candidate SNP, that is, we consider it as a feature that should be recovered by our models.

We did not run the models directly on the whole set of SNPs, but we first selected the top 9 000 SNPs, according to a t-test, for each fold. We selected this number of 9 000 SNPs according to the consistency index across a 10-fold cross-validation over all the data 5.3. To fit the models from each fold we

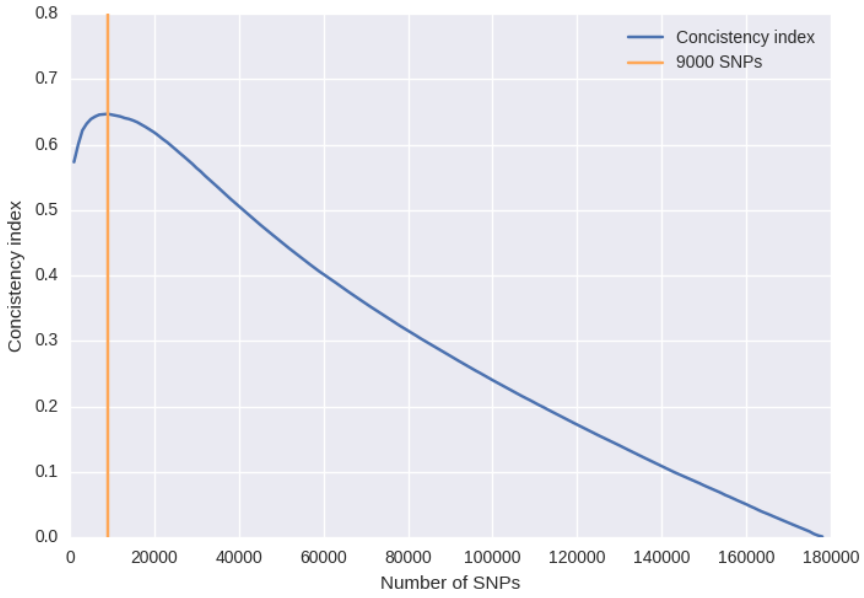


Figure 5.3 – Evaluation of the consistency index over selecting features from a t-test across 10 fold cross validation. The maximum is obtained at 9000 SNPs, which is showed in a vertical line.

performed and inner 3-fold cross validation for selecting the hyper parameters of each model; we previously calculated the top 9 000 SNPs for the inner folds. After we perform feature selection with the proposed methods, we used one ridge regression for each task, with the selected features, for prediction. We use this model for prediction for comparison with [8].

For feature selection, we report the number of candidate SNPs that are recovered by each method, and the total number of selected SNPs in Table 5.3. We also present the number of candidate SNPs that have a $r^2 > 0.6$, where r^2 corresponds to the square of the correlation. To determine the number of selected SNPs, we considered that a SNP was selected if $\theta \neq 0$ for MMLD and for Lasso; in the case of Randomized MMLD we consider those features with higher probability of being selected than a certain threshold selected using

cross validation. We also calculated the consistency of the feature selection across folds (Table 5.2): the consistency of two selected sets of features was defined according to [1]. We finally report the mean of the pairwise comparison of the different folds.

We observe that Randomized MMLD selects more than 2 000 variables, which exceeds the number of features selected by MMLD (615) and those reported in [8]. This behavior is expected, since our method selects features according to the selection of many repetitions of the MMLD. Our method is expected to select a higher number of features but with more stability, particularly for features with high predictive power. As we can see, the number of candidate SNPs selected and the number of candidate SNPs that are highly correlated ($r^2 > 0.6$) with one of the selected SNPs are approximately 4 times larger for the Randomized MMLD with respect to the MMLD; this is equivalent to the increase in the number of selected features. However, we can see how MMLD clearly outperforms Lasso on terms of feature selection.

If we consider the stability of the feature selection (Table 5.2), the Randomized MMLD clearly outperforms the other two methods. Not only is the consistency index significantly higher, but when we restrict the set of features to the candidate SNPs, the consistency of MMLD and Lasso decrease dramatically, while that of Randomized MMLD increases. This is clearly visible in Figure 5.5, where the distribution of the number of times that a SNP is selected has a much heavier tail in the case of Randomized MMLD, especially when we look at the distribution of the number of times a candidate SNP was selected. These two results suggest that it is easier to identify parts of the candidate genes by training many repetitions with the Randomized MMLD than with the simple MMLD.

To better understand the selected features, we focused on the SNPs that

Method	Consistency	Consistency of the candidate SNPs
Randomized MMLD	0.37 ± 0.021	0.45 ± 0.13
MMLD	0.18 ± 0.032	0.01 ± 0.074
Lasso	0.21 ± 0.042	0.011 ± 0.1

Table 5.2 – Feature selection performance of the three methods. Here we show the consistency of the feature selection across the folds and the consistency along the candidate genes.

Method	Selected SNPs	Recovered candidate SNPs	Highly correlated SNPs
Randomized MMLD	2295.0 ± 118.1	12.1 ± 3.02	37.0 ± 8.23
MMLD	615.9 ± 770.62	3.3 ± 5.0	12.7 ± 7.27
Lasso	870.8 ± 12.34	2.2 ± 0.87	8.7 ± 3.32

Table 5.3 – Feature selection performance of the three methods. We show the number of selected SNPs, the number of recovered candidate SNPs, how many candidate SNPs have at least one highly correlated SNP $r^2 > 0.6$ selected.

were selected 9 or more times by the Randomized MMLD repetitions. We obtained a total of 217 SNPs, contained in a total of 112 different genes. Only 3 of them belonged to the list of candidate genes. To better understand the results, we attempted to perform a pathway analysis of those genes. We used the Metacyc [22] database, but unfortunately too many genes have a missing pathway annotation, both among the retrieved list of genes and among the candidate genes, to allow for any conclusive analysis. The biological relevance of the SNPs we identified thus remains unclear.

We also measured the predictivity of the different methods. We applied the same procedure described in [8] for comparison: we select the predictive features according to our method on the training set, and we predict on the test set using a ridge regression trained on the training set only over those

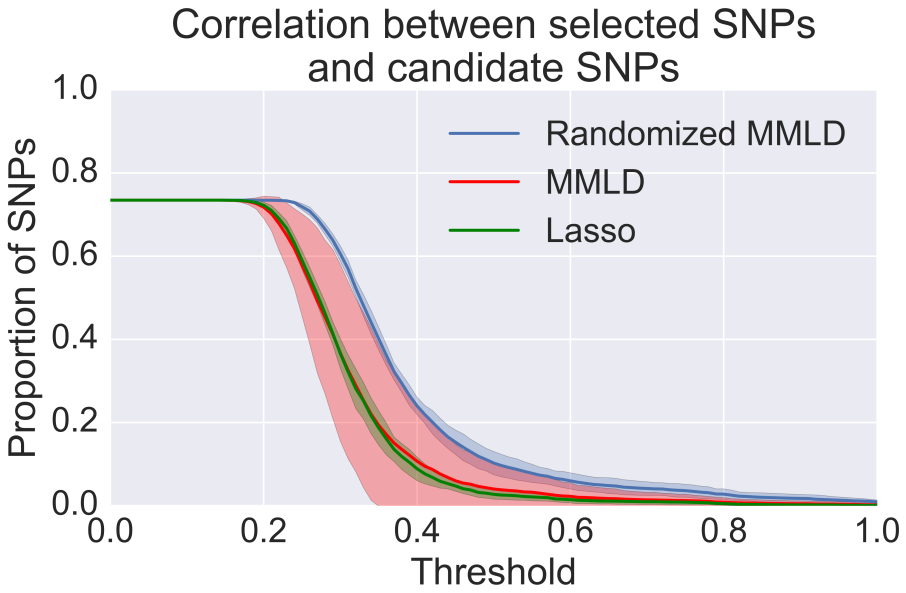


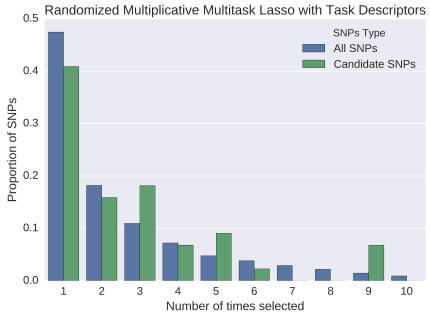
Figure 5.4 – Curve of the mean and standard proportions of selected SNPs that are correlated with the candidate SNPs above a certain threshold.

selected features.

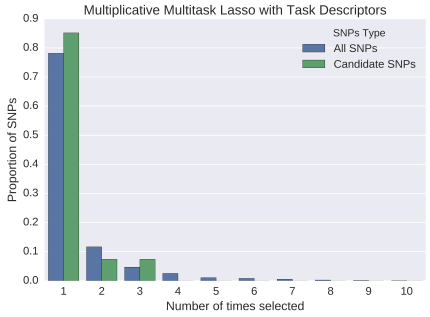
We can observe that Random MMLD is performing better than other models in term of correlation of the predicted with measured phenotype in all tasks. This suggests that the selected SNPs are more informative than those selected by comparison partners. However, this improvement in performance might happen because the Random MMLD selects more features than its counterparts.

5.6 Conclusions

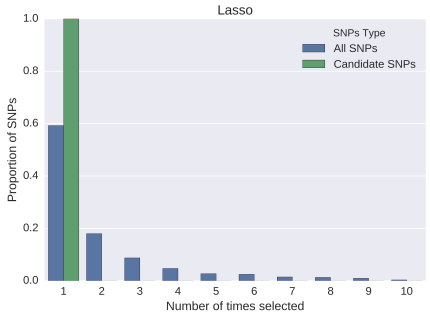
We have presented in this chapter two extensions of the MMLD model to stabilize the feature selection. These models are based on repeated MMLD models. We have evaluated these extensions on a synthetic dataset and a dataset con-



(a)



(b)



(c)

Figure 5.5 – We show the estimated distribution of number of times a SNP is selected by the method after 10 repetitions in blue, and in green we show the distribution for the candidate genes.Figure 5.5a shows it for Randomized MMLD, Figure 5.5b shows the estimation for MMLD and Figure 5.5c shows it for Lasso.

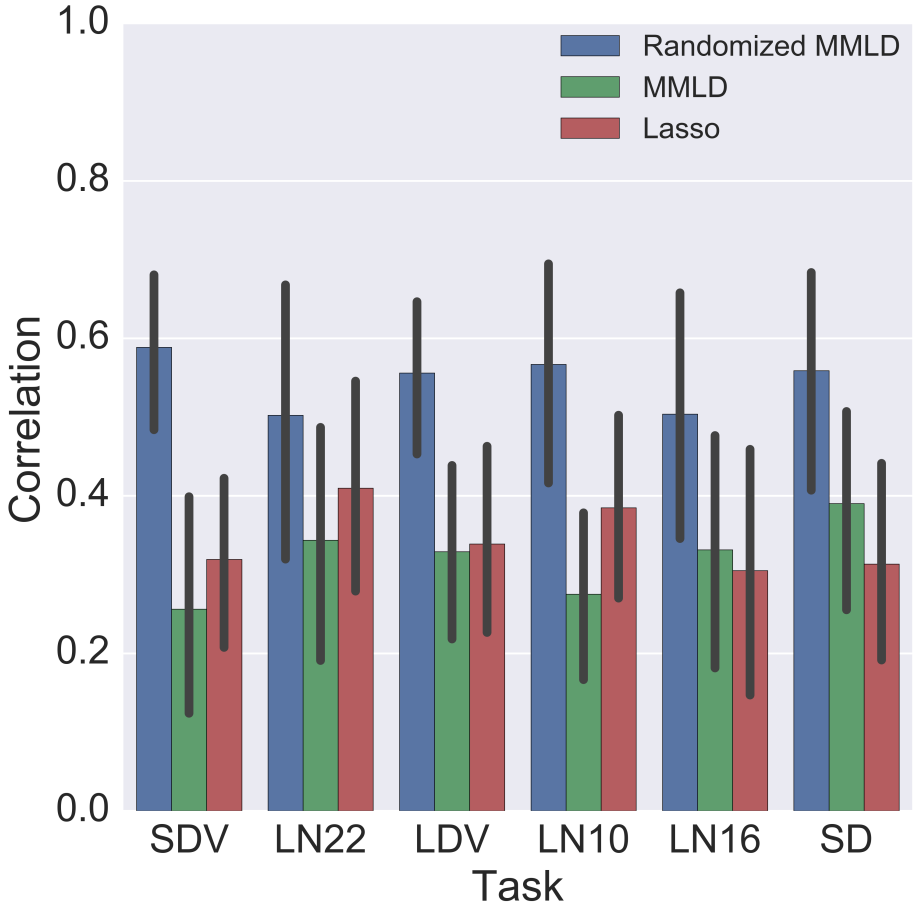


Figure 5.6 – Predictivity measured as Pearson's correlation between the measured values and those predicted by a ridge regression trained with the features selected by the different models.

taining genotype information for *Arabidopsis thaliana*, for which we have a most likely partial ground truth.

Randomized MMLD showed the best performance over synthetic data, while Random MMLD only showed a comparable performance to that of regular MMLD. Nonetheless, Random MMLD seems to have better potential performance than MMLD but the hyperparameters were hard to choose. The analysis of the feature selection in *Arabidopsis thaliana* showed that we obtained a better selection with the Randomized MMLD than with the original MMLD model, according to the list of candidate SNPs. Feature selection is more stable and the features selected are more correlated with the candidate SNPs. However, the total number of selected features is larger in the case of Randomized MMLD, which raises the number of false positives when recovering the correct SNPs. The prediction is improved with respect to MMLD and Lasso for every task, which speaks in favor of this randomized method.

It is also important to notice that due to the lack of information about *Arabidopsis thaliana* pathways, we have not been able to properly analyze the retrieved SNPs that were located in genes. Further experiments on a dataset with more complete information about the organism or studied phenotypes would be required for drawing more robust conclusions about the power of this method.

The methods we presented are an extension of the model presented in Chapter 4. Even though we have seen an improvement over MMLD, they do not result in selecting a clear small set of features that are related with the phenotype. As stated before, experiments where more information is available are necessary to conclude on the usefulness of the method in the context of the clinical question of interest in this thesis, i.e., to find biomarkers for drug side effect prediction.

6 Conclusion

The problem of predicting adverse drug reactions is open and challenging.Any advance on its solution could have significant impact on the health of the global population and save millions of dollars for public health systems, insurance companies and the pharmaceutical industry.

Along this PhD thesis, we have explored how multitask learning methods can be adapted to solve various problems closely related to side effect prediction from genetic data. In the first two chapters, these problems included prediction of drugs toxicology and drug response of patients in the context of DREAM challenges. In the last two chapters, we have presented another model to solve this type of problems in terms of both prediction and feature selection, as well as two extension to stabilize the set of selected markers.

More precisely, in Chapter 2, we focused on predicting the toxicity of various chemical compounds on different cell lines. The data consisted in many organic compounds covering highly diverse chemical structures. This allowed us to explore one limitation of multitask approaches: if tasks are too dissimilar, i.e. if the molecules are structurally very different, multitask methods do not perform better than single tasks methods. Indeed, we observed an improvement in the prediction performance with multitask methods only when tasks were similar, i.e. when predicting the toxicities of molecules with related structures.

In Chapter 3, we present our contribution to the RA Responder DREAM Challenge. We ranked second in the competitive phase, which allowed us to participate in a collaborative phase. In this second phase, we lead the main discussion on whether complex genetic information was bearing relevant information for the prediction task, with respect to the more easily available

clinical information. The overall conclusion was that use of the SNPs information solely did not improve the prediction performance. This indicates that response to treatment is a highly complex phenomenon. In this particular case of response to anti TNF treatment, other genetic, epigenetic, or environmental data would be required to better explain patients response.

In Chapter 4, we presented a new multitask model that uses task descriptors. This allows to make predictions for tasks not seen previously, which is often referred as prediction for orphan tasks. It also provides a way to better understand the relation between tasks, i.e. to better quantify their similarity. The model uses l_1 regularization to enable feature selection. We showed that the model displays state of the art performance for multitask approaches, and that it outperforms single tasks approaches.

In Chapter 5, we studied the problem of the stability of the feature selection of l_1 regularized methods. We considered an adaptation of the Randomized Lasso to our model. Unfortunately, when tested on real-world problems, the method failed to provide reliable sets of biomarkers. A lot of work remains on strategies for feature selection, on how to combine these heterogeneous datasets to discover real biomarkers that yield better explanations of ADRS than more easily obtained non genetic factors. For example, we have seen in the Rheumatoid Arthritis challenge that in datasets that live in large dimensions (the SNPs, in this case), it can be difficult to select actionable information because traits can be complex. This complexity might be because there is a non-linear interaction between biomarker and traits. These relationship might include several genes or they might be better explained by other genetic factors. The studied traits can sometimes be explained by easily obtainable non genetic factors, that should be taken into account.

In summary, we have explored multitask models in various difficult settings.

We concluded that they can be useful to deal with complex datasets and provide better predictions than single task methods when tasks are related, or when a distance between tasks can be defined. In particular, the multitask approach allows training of a predictive model from groups of small related datasets. We have shown that using task descriptors tends to improve the performance of the model. It is also useful to understand tasks similarity, which governs to which extent information should be shared between tasks. We found that this is an important contribution to better identify situations where multitask approaches are relevant. In fact, we have seen in Chapter 2 that there exist datasets for which, in spite of having a lot of information about the tasks, the tasks are too dissimilar for the multitask approach to work. In the case where descriptors do not exist, understanding the relationship between the tasks is fundamental for removing those task that might harm the performance of the model. Automatic methods that learn the relation and similarities between the task might be useful for solving the problem [88, 79].

As a conclusion remark, we found that a lot of work remains to be done in order to provide the community with appropriate datasets. Severe ADR can be rare, complex, and difficult to define unambiguously. Although some databases such as SIDER constitute efforts in this direction, consortia should be build between countries to monitor ADR, adopt common definitions, and obtain enough cases and genotype them. Such data would be very useful in order to train prediction models displaying good generalization performance. We consider this to be the first step towards enabling the training of machine learning models for ADR prediction from genetic features. This work is also a first methodological approach to the problem; many alternative strategies remain to be explored. First of all, the use of non-linear model should be more widely studied, for this problem and for other in bioinformatics, where feature

selection still mostly relies on regularized regression. We used kernel methods for prediction, which already show good results. Part of the community is beginning to show interest in the application of deep learning techniques to computational biology [76, 97, 6]. This is an additional new lead worth following.

Different types of data should also be considered. Different biological tests might be of importance for predicting ADR, from concentration of various substances in blood, to different genetic data, such as genetic expression, mutation data, and also proteomics.Furthermore, probably none of this data by itself is enough for prediction, and strategies for their integration should be studied. Finally, patient records, which have become more exploitable thanks to the use of electronic health records, could also be an invaluable resource.

The identification of plausible side effects for a given molecule is also an interesting complementary topic that can help avoid ADR that were undetected on clinical trials [26, 109]. Finally, obtaining more complete information on the targets of the different drugs would help advancing our understanding of side effects. In particular, it could help selecting which genes may be involved in ADR and therefore help guide feature selection in the type of models we have proposed.

As a summary, general ADR prediction is still a far objective, due to the high complexity of the problem, the unavailability of relevant and complete data, and the necessity of identifying appropriate computational models for predicting response and identifying biomarkers.

Bibliography

[1] *A stability index for feature selection*, volume 25, 2007.

[2] Nour Abdo, Menghang Xia, Chad C Brown, et al. Population-Based in Vitro Hazard and Concentration–Response Assessment of Chemicals: The 1000 Genomes High-Throughput Screening Study. *Environmental health perspectives*, 123(5):458–466, 2015.

[3] Gonçalo R Abecasis, David Altshuler, Adam Auton, et al. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–73, October 2010.

[4] DE Adkins, SL Clark, K Åberg, et al. Genome-wide pharmacogenomic study of citalopram-induced side effects in STAR*D. *Translational psychiatry*, 2:129, January 2012.

[5] Ivan A Adzhubei, Steffen Schmidt, Leonid Peshkin, et al. A method and server for predicting damaging missense mutations. *Nature methods*, 7(4):248–9, April 2010.

[6] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep learning for computational biology. *Molecular systems biology*, 12(7):878, 2016.

[7] Susanna Atwell, Yu S Huang, Bjarni J Vilhjálmsson, et al. Genome-wide association study of 107 phenotypes in a common set of Arabidopsis thaliana inbred lines. *Nature*, 465(7298):627–631, 2010.

[8] Chloé-Agathe Azencott, Dominik Grimm, Mahito Sugiyama, Yoshinobu Kawahara, and Karsten M Borgwardt. Efficient network-guided multi-

locus association mapping with graph cuts. *Bioinformatics*, 29(13):i171–i179, 2013.

[9] Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. *Journal of Machine Learning Research*, 4(1):83–99, 2003.

[10] Victor Bellon, Véronique Stoven, and Chloé-Agathe Azencott. Multi-task feature selection with task descriptors. In *Pacific Symposium on Biocomputing*, volume 21, pages 261–272, 2016.

[11] Asa Ben-Hur and William Stafford Noble. Kernel methods for predicting protein-protein interactions. *Bioinformatics*, 21(SUPPL. 1), 2005.

[12] Elsa Bernard. Kernel bilinear regression for toxicogenetics. In *RECOMB Conference on Regulatory and Systems Genomics*, 2013.

[13] Steffen Bickel, Jasmina Bogojeska, Thomas Lengauer, and Tobias Scheffer. Multi-task learning for hiv therapy screening. In *Proceedings of the 25th international conference on Machine learning*, pages 56–63. ACM, 2008.

[14] Edwin V Bonilla, Felix V Agakov, and Christopher K I Williams. Kernel multi-task learning using task-specific features. *The 11th International Conference on Artificial Intelligence and Statistics*, pages 43–50, 2007.

[15] Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task Gaussian process prediction. *Advances in Neural Information Processing Systems 20*, 20:153–160, 2007.

[16] L Breiman. Random forests. *Machine learning*, pages 5–32, 2001.

[17] Leo Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.

[18] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[19] Aurel Cami, Alana Arnold, Shannon Manzi, and Ben Reis. Predicting adverse drug events using pharmacological network models. *Science translational medicine*, 3(114), 2011.

[20] Colin Campbell. Kernel methods: a survey of current techniques. *Neurocomputing*, 48(1):63–84, 2002.

[21] Rich Caruana. Multitask Learning. *Machine Learning, 28*, 75:41–75, 1997.

[22] Ron Caspi, Richard Billington, Luciana Ferrer, et al. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway / genome databases. *Nucleic Acids Research*, 44(November 2015):471–480, 2016.

[23] José V Castell and Maria Jose Gómez-Lechón. *In vitro methods in pharmaceutical research*. Academic press, 1996.

[24] Stephen Checkley, Linda MacCallum, James Yates, et al. Bridging the gap between in vitro and in vivo: Dose and schedule predictions for the atr inhibitor azd6738. *Scientific reports*, 5, 2015.

[25] Lei Chen, Tao Huang, Jian Zhang, et al. Predicting drugs side effects based on chemical-chemical interactions and protein-chemical interactions. *BioMed research international*, 2013:485034, January 2013.

[26] Lei Chen, Tao Huang, Jian Zhang, et al. Predicting drugs side effects based on chemical-chemical interactions and protein-chemical interactions. *BioMed research international*, 2013, 2013.

[27] Liuhong Chen, Changxi Li, Stephen Miller, and Flavio Schenkel. Multi-population genomic prediction using a multi-task bayesian learning model. *BMC Genetics*, 15(1):53, 2014.

[28] Feixiong Cheng and Zhongming Zhao. Machine learning-based prediction of drug-drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties. *Journal of the American Medical Informatics Association*, 21(e2):e278–e286, 2014.

[29] Francis S Collins, Michael Morgan, and Aristides Patrinos. The Human Genome Project: lessons from large-scale biology. *Science (New York, N.Y.)*, 300(5617):286–90, April 2003.

[30] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. *Architecture*, 20(1):160–167, 2008.

[31] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[32] Jing Cui, Eli a Stahl, Saedis Saevarsdottir, and et. al. Genome-wide association study and gene expression analysis identifies CD84 as a predictor of response to etanercept therapy in rheumatoid arthritis. *PLoS genetics*, 9(3):e1003394, March 2013.

[33] Jing Cui, Eli a Stahl, Saedis Saevarsdottir, et al. Genome-wide association study and gene expression analysis identifies CD84 as a predictor of

response to etanercept therapy in rheumatoid arthritis. *PLoS genetics*, 9(3):e1003394, mar 2013.

[34] Joseph A Dimasi, Henry G Grabowski, and Ronald W Hansen. Innovation in the pharmaceutical industry: New estimates of R&D costs. *Journal of Health Economics*, 47:20–33, 2016.

[35] Annette J Dobson and Adrian Barnett. *An introduction to generalized linear models*. CRC press, 2008.

[36] Irini A Doytchinova, Pingping Guan, and Darren R Flower. Identifying human MHC supertypes using bioinformatic methods. *The Journal of Immunology*, 172(7):4314–4323, 2004.

[37] Federica Eduati, Lara M Mangravite, Tao Wang, et al. Prediction of human population responses to toxic compounds by a collaborative competition. *Nature biotechnology*, 33(9):933–40, 2015.

[38] Bradley Efron, Trevor Hastie, Iain Johnston, and Robert Tibshirani. Least Angle Regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[39] William E Evans and Howard L McLeod. Pharmacogenomics–drug disposition, drug targets, and side effects. *The New England journal of medicine*, 348(6):538–49, March 2003.

[40] Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, pages 615–637, 2005.

[41] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi-task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004.

[42] Jianqing Fan, Jinchi Lv, and Lei Qi. Sparse high dimensional models in economics. *Annual review of economics*, 3:291–317, 2011.

[43] D. R. Flower. On the properties of bit string-based measures of chemical similarity. *Journal of Chemical Information and Computer Sciences*, 38(3):379–386, 1998.

[44] Hirokazu Furuya, Pedro Fernandez-Salguero, Wendy Gregory, et al. Genetic polymorphism of CYP2C9 and its effect on warfarin maintenance dose requirement in patients undergoing anticoagulation therapy. *Pharmacogenetics and Genomics*, 5(6):389–392, 1995.

[45] Joumana Ghosn and Yoshua Bengio. Multi-Task Learning for Stock Selection. *Advances in Neural Information Processing Systems 9 (NIPS'96)*, pages 946–952, 1997.

[46] Assaf Gottlieb, Gideon Y Stein, Yoram Oron, Eytan Ruppin, and Roded Sharan. INDI: a computational framework for inferring drug interactions and their associated recommendations. *Molecular systems biology*, 8(592):592, January 2012.

[47] Dominik G Grimm, Chloé Agathe Azencott, Fabian Aicheler, et al. The evaluation of tools used to predict the impact of missense variants is hindered by two types of circularity. *Human Mutation*, 36(5):513–523, 2015.

[48] David Heckerman, Carl Kadie, and Jennifer Listgarten. Leveraging information across hla alleles/supertypes improves epitope prediction. *Journal of Computational Biology*, 14(6):736–746, 2007.

[49] Arthur E Hoerl and Robert W Kennard. Ridge Regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[50] Jialiang Huang, Chaoqun Niu, Christopher D. Green, et al. Systematic prediction of pharmacodynamic drug-drug interactions through protein-protein-interaction network. *PLoS Computational Biology*, 9(3), 2013.

[51] Liang-Chin Huang, Xiaogang Wu, and Jake Y Chen. Predicting adverse side effects of drugs. *BMC Genomics*, 12(Suppl 5):S11, 2011.

[52] IEEE. *SNP sets selection under mutual information criterion, application to F7/FVII dataset*, 2008.

[53] Srinivasan V. Iyer, Rave Harpaz, Paea LePendu, Anna Bauer-Mehren, and Nigam H. Shah. Mining clinical text for signals of adverse drug-drug interactions. *Journal of the American Medical Informatics Association*, 21(2):353–362, 2014.

[54] Laurent Jacob, Brice Hoffmann, Véronique Stoven, and Jean-Philippe Vert. Virtual screening of GPCRs: an in silico chemogenomics approach. *BMC bioinformatics*, 9:363, January 2008.

[55] Laurent Jacob and Jean-Philippe Vert. Efficient peptide-MHC-i binding prediction for alleles with few known binders. *Bioinformatics*, 24(3):358–366, 2008.

[56] Laurent Jacob and Jean-Philippe Vert. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics (Oxford, England)*, 24(19):2149–56, October 2008.

[57] Ali Jalali, Sujay Sanghavi, Chao Ruan, and Pradeep K Ravikumar. A dirty model for multi-task learning. *Advances in Neural Information Processing Systems 23*, 23:964–972, 2010.

[58] W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics (Oxford, England)*, 8(1):118–27, January 2007.

[59] Iain M. Johnstone and D. Michael Titterington. Statistical challenges of high-dimensional data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906):4237–4253, 2009.

[60] Daniël M Jonker, Sandra AG Visser, Piet H van der Graaf, Rob A Voskuyl, and Meindert Danhof. Towards a mechanism-based analysis of pharmacodynamic drug–drug interactions in vivo. *Pharmacology & therapeutics*, 106(1):1–18, 2005.

[61] Richard Judson, Ann Richard, David J. Dix, et al. The toxicity data landscape for environmental chemicals. *Environmental Health Perspectives*, 117(5):685–695, 2009.

[62] Minoru Kanehisa, Yoko Sato, Masayuki Kawashima, Miho Furumichi, and Mao Tanabe. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Research*, 44(D1):D457–D462, 2016.

[63] Alla Katsnelson. Momentum grows to make'personalized'medicine more'precise'. *Nature medicine*, 19(3):249–249, 2013.

[64] Michael Kuhn, Mumna Al Banchaabouchi, Monica Campillos, et al. Systematic identification of proteins that elicit drug side effects. *Molecular systems biology*, 9(663):663, January 2013.

[65] Prateek Kumar, Steven Henikoff, and Pauline C Ng. Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. *Nature protocols*, 4(7):1073–81, January 2009.

[66] Vivian Law, Craig Knox, Yannick Djoumbou, et al. DrugBank 4.0: Shedding new light on drug metabolism. *Nucleic Acids Research*, 42(D1):1091–1097, 2014.

[67] Jason Lazarou, Bruce H Pomeranz, and Paul N Corey. Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies. *Jama*, 279(15):1200–1205, 1998.

[68] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, et al. Deep learning. *Nature*, 521(7553):436–444, 2015.

[69] Aurelie C Lozano and Grzegorz Swirszcz. Multi-level lasso for sparse multi-task regression. *Proceedings of the 29th International Conference on Machine Learning*, 29:361–368, 2012.

[70] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[71] Pierre Mahé, Liva Ralaivola, Véronique Stoven, and Jean-Philippe Vert. The pharmacophore kernel for virtual screening with support vector machines. *Journal of chemical information and modeling*, 46(5):2003–14, 2014.

[72] Gil A. McVean, David M. Altshuler (Co-Chair), Richard M. Durbin (Co-Chair), et al. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65, 2012.

[73] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

[74] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[75] Ana Miguel, Luís Filipe Azevedo, Manuela Araújo, and Altamiro Costa Pereira. Frequency of adverse drug reactions in hospitalized patients: a systematic review and meta-analysis. *Pharmacoepidemiology and drug safety*, 21(11):1139–1154, 2012.

[76] Seonwoo Min, Byunghan Lee, and Sungroh Yoon. Deep learning in bioinformatics. *Briefings in Bioinformatics*, 2016.

[77] Marvin Minsky and Seymour Papert. Perceptrons. 1969.

[78] Sayaka Mizutani, Edouard Pauwels, Véronique Stoven, Susumu Goto, and Yoshihiro Yamanishi. Relating drug-protein interaction network with drug side effects. *Bioinformatics*, 28(18):522–528, 2012.

[79] Keerthiram Murugesan and Jaime Carbonell. Multi-task multiple kernel relationship learning. *arXiv preprint arXiv:1611.03427*, 2016.

[80] MR Nelson, SA Bacanu, M Mosteller, et al. Genome-wide approaches to identify pharmacogenetic contributions to adverse drug reactions. *The pharmacogenomics journal*, 9(1):23–33, 2009.

[81] Yurii Nesterov. *Introductory Lectures on Convex Optimization*, volume 87. Springer Science & Business Media, 2004.

[82] Guillaume Obozinski, Ben Taskar, and Michael Jordan. Multi-task feature selection. *Statistics Department, UC Berkeley, Tech. Rep*, 2006.

[83] Hiroyuki Ogata, Susumu Goto, Kazushige Sato, et al. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 27(1):29–34, 1999.

[84] Igho J Onakpoya, Carl J Heneghan, and Jeffrey K Aronson. Post-marketing withdrawal of 462 medicinal products because of adverse drug reactions: a systematic review of the world literature. *BMC Medicine*, pages 1–11, 2016.

[85] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[86] D A Pappas, J M Kremer, G Reed, J D Greenberg, and J R Curtis. Design characteristics of the CORRONA CERTAIN study: a comparative effectiveness study of biologic agents for rheumatoid arthritis patients. *BMC Musculoskelet Disord*, 15:113, 2014.

[87] Edouard Pauwels, Véronique Stoven, and Yoshihiro Yamanishi. Predicting drug side-effect profiles: a chemical fragment-based approach. *BMC bioinformatics*, 12(1):169, January 2011.

[88] Anastasia Pentina and Christoph H Lampert. Active task selection for multi-task learning. *arXiv preprint arXiv:1602.06518*, 2016.

[89] Alexandre Perera, Alfonso Buil, Maria Chiara Di Bernardo, et al. Clustering of individuals given SNPs similarity based on normalized mutual information: F7 SNPs in the GAIT sample. *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, pages 123–126, 2007.

[90] Aritz Pérez. *Supervised classification in continuous domains with Bayesian networks*. PhD thesis, Universidad del Pais Vasco, 2010.

[91] Bjoern Peters, Huynh-Hoa Bui, Sune Frankild, et al. A community resource benchmarking predictions of peptide binding to MHC-I molecules. *PLoS Comput Biol*, 2(6):e65, 2006.

[92] MLL Prevoo, MA Van't Hof, HH Kuper, et al. Modified disease activity scores that include twenty-eight-joint counts development and validation in a prospective longitudinal study of patients with rheumatoid arthritis. *Arthritis & Rheumatism*, 38(1):44–48, 1995.

[93] Kriti Puniyani, Seyoung Kim, and Eric P Xing. Multi-population GWAS mapping via multi-task regularized regression. *Bioinformatics*, 26(12):i208–i216, 2010.

[94] Yanjun Qi, Merja Oja, Jason Weston, and William Stafford Noble. A unified multitask architecture for predicting local protein properties. *PLoS ONE*, 7(3), 2012.

[95] Liva Ralaivola, Sanjay J Swamidass, Hiroto Saigo, and Pierre Baldi. Graph kernels for chemical informatics. *Neural networks : the official journal of the International Neural Network Society*, 18(8):1093–110, October 2005.

[96] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning.*, volume 14. MIT Press, apr 2006.

[97] Daniele Ravi, Charence Wong, Fani Deligianni, et al. Deep learning for health informatics. *IEEE Journal of Biomedical and Health Informatics*, 2016.

[98] Mary V Relling and William E Evans. Pharmacogenomics in the clinic. *Nature*, 526(7573):343–50, 2015.

[99] J Robinson, A Malik, P Parham, JG Bodmer, and SGE Marsh. IMGT/HLA database–a sequence database for the human major histocompatibility complex. *Tissue antigens*, 55(3):280–287, 2000.

[100] Guilherme V Rocha, Xing Wang, and Bin Yu. Asymptotic distribution and sparsistency for l1-penalized parametric m-estimators with applications to linear SVM and logistic regression. *arXiv preprint arXiv:0908.1940*, 2009.

[101] David Rogers, Robert D Brown, and Mathew Hahn. Using extended-connectivity fingerprints with Laplacian-modified Bayesian analysis in high-throughput screening follow-up. *Journal of biomolecular screening*, 10(7):682–6, October 2005.

[102] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–54, May 2010.

[103] Josef Scheiber, Jeremy L Jenkins, Sai Chetan K Sukuru, et al. Mapping adverse drug reactions in chemical space. *J Med Chem*, 52(9):3103–3107, 2009.

[104] Sebastian Schneeweiss, Joerg Hasford, Martin Göttler, et al. Admissions caused by adverse drug events to internal medicine and emergency departments in hospitals: a longitudinal population-based study. *European Journal of Clinical Pharmacology*, 58:285–291, 2002.

[105] Bernhard Schölkopf, Koji Tsuda, and Jean-Philippe Vert. *Kernel methods in computational biology*. MIT press, 2004.

[106] Jana Marie Schwarz, Christian Rödelsperger, Markus Schuelke, and Dominik Seelow. MutationTaster evaluates disease-causing potential of sequence alterations. *Nature methods*, 7(8):575–6, August 2010.

[107] Vincent Segura, Bjarni J Vilhjálmsson, Alexander Platt, et al. An efficient multi-locus mixed-model approach for genome-wide association studies in structured populations. *Nature Genetics*, 44(7):825–830, 2012.

[108] Giovanni Severino and Maria Del Zompo. Adverse drug reactions: role of pharmacogenomics. *Pharmacological Research*, 49(4):363–373, 2004.

[109] Itay Shaked, Matthew A Oberhardt, Nir Atias, Roded Sharan, and Eytan Ruppin. Metabolic network prediction of drug side effects. *Cell systems*, 2(3):209–213, 2016.

[110] Solveig K Sieberts, Fan Zhu, Javier García-García, et al. Crowdsourced assessment of common genetic contribution to predicting anti-TNF treatment response in rheumatoid arthritis. *Nature communications*, 7:12460, 2016.

[111] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[112] AJ Smola and B Schölkopf. A tutorial on support vector regression. *Statistics and computing*, pages 199–222, 2004.

[113] Brian B. Spear, Margo Heath-Chiozzi, and Jeffrey Huff. Clinical application of pharmacogenetics. *Trends in Molecular Medicine*, 7(5):201–204, 2001.

[114] Paul P. Tak. A personalized medicine approach to biologic treatment of rheumatoid arthritis: A preliminary treatment algorithm. *Rheumatology*, 51(4):600–609, 2012.

[115] Fumihiko Takeuchi, Ralph McGinnis, Stephane Bourgeois, and et al. Barnes. A genome-wide association study confirms VKORC1, CYP2C9, and CYP4F2 as principal genetic determinants of warfarin dose. *PLoS genetics*, 5(3):e1000433, March 2009.

[116] Caroline F Thorn, Teri E Klein, and Russ B Altman. Pharmgkb: the pharmacogenomics knowledge base. *Pharmacogenomics: Methods and Protocols*, pages 311–320, 2013.

[117] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[118] Raymond R. Tice, Christopher P. Austin, Robert J. Kavlock, and John R. Bucher. Improving the human hazard characterization of chemicals: A Tox21 update. *Environmental Health Perspectives*, 121(7):756–765, 2013.

[119] AM van Gestel, ML Prevoo, MA van 't Hof, et al. Development and validation of the European League Against Rheumatism response criteria for rheumatoid arthritis. Comparison with the preliminary American College of Rheumatology and the World Health Organization/International League Against Rheumatism cri. *Arthritis and Rheumatism*, 39(1):34–40, 1996.

[120] Katharina Wagner, Frederik Damm, Gudrun Göhring, et al. Impact of idh1 r132 mutations and an idh1 single nucleotide polymorphism in

cytogenetically normal acute myeloid leukemia: Snp rs11554137 is an adverse prognostic factor. *Journal of clinical oncology*, 28(14):2356–2364, 2010.

[121] Sijian Wang, Bin Nan, Saharon Rosset, and Ji Zhu. Random lasso. *Annals of Applied Statistics*, 5(1):468–485, 2011.

[122] Xin Wang, Jinbo Bi, Shipeng Yu, and Jiangwen Sun. On multiplicative multitask feature learning. *Advances in Neural Information Processing Systems 27*, 27:2411–2419, 2014.

[123] Richard Weinshilboum. Inheritance and Drug Response. *The New England journal of medicine*, 348:529–537, 2003.

[124] Christian Widmer, Nora C. Toussaint, Yasemin Altun, and Gunnar Rätsch. Inferring latent task structure for Multitask Learning by Multiple Kernel Learning. *BMC Bioinformatics*, 11(Suppl 8):S5, 2010.

[125] Larry C Wienkers and Timothy G Heath. Predicting in vivo drug interactions from in vitro drug discovery data. *Nature reviews. Drug discovery*, 4(10):825–833, 2005.

[126] Russell A Wilke, Debbie W Lin, Dan M Roden, et al. Identifying genetic risk factors for serious adverse drug reactions : current progress and challenges. *Nature reviews. Drug discovery*, 6(november):904–916, 2007.

[127] CKI Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. *Learning in graphical models*, 1998.

[128] World Health Organization. International drug monitoring. The role of the hospital. Technical report, World Health Organization, 1969.

[129] Menghang Xia, Ruili Huang, Kristine L Witt, et al. Compound cytotoxicity profiling using quantitative high-throughput screening. *Environmental health perspectives*, 116(3):284, 2008.

[130] Jack Y Yang, Guo-Zheng Li, Hao-Hua Meng, Mary Qu Yang, and Youping Deng. Improving prediction accuracy of tumor classification by reusing genes discarded during gene selection. *BMC genomics*, 9(1):1, 2008.

[131] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[132] Feng Zhang and James R. Lupski. Non-coding genetic variants in human disease. *Human Molecular Genetics*, 24(R1):R102–R110, 2015.

[133] Lei Zhang, Yuanchao Derek Zhang, Ping Zhao, and Shiew-Mei Huang. Predicting drug-drug interactions: an FDA perspective. *The AAPS journal*, 11(2):300–6, June 2009.

[134] H Zou and T Hastie. Regularization and variable selection via the elastic-net. *Journal of the Royal Statistical Society*, 67:301–320, 2005.

[135] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

Résumé

Les effets indésirables médicamenteux (EIM) ont des répercussions considérables tant sur la santé que sur l'économie. De $1,9\%$ à $2,3\%$ des patients hospitalisés en sont victimes, et leur coût a récemment été estimé aux alentours de 400 millions d'euros pour la seule Allemagne. De plus, les EIM sont fréquemment la cause du retrait d'un médicament du marché, conduisant à des pertes pour l'industrie pharmaceutique se chiffrant parfois en millions d'euros.
De multiples études suggèrent que des facteurs génétiques jouent un rôle non négligeable dans la réponse des patients à leur traitement. Cette réponse comprend non seulement les effets thérapeutiques attendus, mais aussi les effets secondaires potentiels. C'est un phénomène complexe, et nous nous tournons vers l'apprentissage statisitque pour proposer de nouveaux outils permettant de mieux le comprendre.
Nous étudions différents problèmes liés à la prédiction de la réponse d'un patient à son traitement à partir de son profil génétique. Pour ce faire, nous nous plaçons dans le cadre de l'apprentissage statistique multitâche, qui consiste à combiner les données disponibles pour plusieurs problèmes liés afin de les résoudre simultanément. Nous proposons un nouveau modèle linéaire de prédiction multitâche qui s'appuie sur des descripteurs des tâches pour sélectionner les variables pertinentes et améliorer les prédictions obtenues par les algorithmes de l'état de l'art. Enfin, nous étudions comment améliorer la stabilité des variables sélectionnées, afin d'obtenir des modèles interprétables.

Mots Clés

apprentissage statistique, médecine personalisée, prédiction d'effets secondaires, effets secondaires indésirables, apprentissage multitâche

Abstract

Adverse drug reaction (ADR) is a serious concern that has important health and economical repercussions. Between $1.9\% - 2.3\%$ of the hospitalized patients suffer from ADR, and the annual cost of ADR have been estimated to be of 400 million euros in Germany alone. Furthermore, ADRs can cause the withdrawal of a drug from the market, which can cause up to millions of dollars of losses to the pharmaceutical industry.
Multiple studies suggest that genetic factors may play a role in the response of the patients to their treatment. This covers not only the response in terms of the intended main effect, but also in terms of potential side effects. The complexity of predicting drug response suggests that machine learning could bring new tools and techniques for understanding ADR.
In this doctoral thesis, we study different problems related to drug response prediction, based on the genetic characteristics of patients. We frame them through multitask machine learning frameworks, which combine all data available for related problems in order to solve them at the same time. We propose a novel model for multitask linear prediction that uses task descriptors to select relevent features and make predictions with better performance as state-of-the-art algorithms. Finally, we study strategies for increasing the stability of the selected features, in order to improve interpretability for biological applications.

Keywords

machine learning, personalized medicine, side effect prediction, adverse drug reaction, multitask learning,