Learning from genomic data : efficient representations and algorithms.

Marine Le Morvan

THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres
PSL Research University

Préparée à MINES ParisTech

Learning from genomic data: efficient representations and algorithms.

Développement de représentations et d'algorithmes efficaces pour l'apprentissage statistique sur des données génomiques.

École doctorale nº432

SCIENCES ET MÉTIERS DE L'INGÉNIEUR

Spécialité BIO-INFORMATIQUE

Soutenue par **Marine LE MORVAN**
le 3 Juillet 2018

Dirigée par **Jean-Philippe VERT**
Andrei Zinovyev

COMPOSITION DU JURY :

M Alexandre GRAMFORT
Université Paris Saclay, Rapporteur, Président

M Fabio VANDIN
University of Padova, Rapporteur

Mme Chloé-Agathe Azencott
MINES ParisTech, Membre du jury

Mme Florence d'Alché-Buc
Télécom ParisTech, Membre du jury

M Andrei ZINOVYEV
Institut Curie, Membre du jury

M Jean-Philippe VERT
MINES ParisTech, Membre du jury

Abstract

Since the first sequencing of the human genome in the early 2000s, large endeavours have set out to map the genetic variability among individuals, or DNA alterations in cancer cells. They have laid foundations for the emergence of precision medicine, which aims at integrating the genetic specificities of an individual with its conventional medical record to adapt treatment, or prevention strategies. Translating DNA variations and alterations into phenotypic predictions is however a difficult problem. DNA sequencers and microarrays measure more variables than there are samples, which poses statistical issues. The data is also subject to technical biases and noise inherent in these technologies. Finally, the vast and intricate networks of interactions among proteins obscure the impact of DNA variations on the cell behaviour, prompting the need for predictive models that are able to capture a certain degree of complexity. This thesis presents novel methodological contributions to address these challenges. First, we define a novel representation for tumour mutation profiles that exploits prior knowledge on protein-protein interaction networks. For certain cancers, this representation allows improving survival predictions from mutation data as well as stratifying patients into meaningful subgroups. Second, we present a new learning framework to jointly handle data normalisation with the estimation of a linear model. Our experiments show that it improves prediction performances compared to handling these tasks sequentially. Finally, we propose a new algorithm to scale up sparse linear models estimation with two-way interactions. The obtained speed-up makes this estimation possible and efficient for datasets with hundreds of thousands of main effects, thereby extending the scope of such models to the data from genome-wide association studies.

Résumé

Depuis le premier séquençage du génome humain au début des années 2000, de grandes initiatives se sont lancé le défi de construire la carte des variabilités génétiques inter-individuelles, ou bien encore celle des altérations de l'ADN tumoral. Ces projets ont posé les fondations nécessaires à l'émergence de la médecine de précision, dont le but est d'intégrer aux dossiers médicaux conventionnels les spécificités génétiques d'un individu, afin de mieux adapter les traitements et les stratégies de prévention. La traduction des variations et des altérations de l'ADN en prédictions phénotypiques constitue toutefois un problème difficile. Les séquenceurs ou puces à ADN mesurent plus de variables qu'il n'y a d'échantillons, posant ainsi des problèmes statistiques. Les données brutes sont aussi sujettes aux biais techniques et au bruit inhérent à ces technologies. Enfin, les vastes réseaux d'interactions à l'échelle des protéines obscurcissent l'impact des variations génétiques sur le comportement de la cellule, et incitent au développement de modèles prédictifs capables de capturer un certain degré de complexité. Cette thèse présente de nouvelles contributions méthodologiques pour répondre à ces défis. Tout d'abord, nous définissons une nouvelle représentation des profils de mutations tumorales, qui exploite leur position dans les réseaux d'interaction protéine-protéine. Pour certains cancers, cette représentation permet d'améliorer les prédictions de survie à partir des données de mutations, et de stratifier les cohortes de patients en sous-groupes informatifs. Nous présentons ensuite une nouvelle méthode d'apprentissage permettant de gérer conjointement la normalisation des données et l'estimation d'un modèle linéaire. Nos expériences montrent que cette méthode améliore les performances prédictives par rapport à une gestion séquentielle de la normalisation puis de l'estimation. Pour finir, nous accélérons l'estimation de modèles linéaires parcimonieux, prenant en compte des interactions deux à deux, grâce à un nouvel algorithme. L'accélération obtenue rend cette estimation possible et efficace sur des jeux de données comportant plusieurs centaines de milliers de variables originales, permettant ainsi d'étendre la portée de ces modèles aux données des études d'associations pangénomiques.

“ *Theory is when you know everything but nothing works. Practice is when everything works but no one knows why. In our lab, theory and practice are combined : nothing works and no one knows why.* ”

---

attributed to Albert Einstein

Contents

Abstract i

Résumé iii

Contents vii

1 Introduction 1
- 1.1 Contextual setting . . . 3
- 1.2 Cancer Genomics . . . 5
- 1.3 GWAS . . . 10
- 1.4 Statistical learning . . . 12
- 1.5 Learning in high dimension . . . 15
- 1.6 Computational challenges . . . 22
- 1.7 Contributions . . . 25

2 NetNorM 31
- 2.1 Introduction . . . 33
- 2.2 Overview of NetNorM . . . 34
- 2.3 Survival prediction . . . 36
- 2.4 Patient stratification . . . 46
- 2.5 Discussion . . . 51
- 2.6 Materials and Methods . . . 53

3 Supervised Quantile Normalisation 59
- 3.1 Introduction . . . 61
- 3.2 Quantile normalisation (QN) . . . 62
- 3.3 Supervised quantile normalisation (SUQUAN) . . . 63
- 3.4 SUQUAN as a matrix regression problem . . . 64
- 3.5 Algorithms . . . 65
- 3.6 Experiments . . . 68
- 3.7 Discussion . . . 73

4 WHInter 75
- 4.1 Introduction . . . 77
- 4.2 Preliminaries . . . 79
- 4.3 The WHInter algorithm . . . 81
- 4.4 Simulation study . . . 86
- 4.5 Results on real world data . . . 87
- 4.6 Related work . . . 89

4.7 Discussion 91

5 Conclusion **93**

A Supplementaries for NetNorM **97**
A.1 Supplementary figures 97
A.2 Supplementary tables 104

B Supplementaries for WHInter **107**
B.1 Proof of Lemma 4.3.1 107
B.2 Computing η_{min} 107
B.3 Alternative solver for working set updates 111
B.4 SPP: depth-first vs breadth-first 114
B.5 Suppplementary figures 115

Bibliography **119**

Chapter 1

Introduction

Contents

1.1 Contextual setting . . . 3
1.1.1 Human genome sequencing: a bit of recent history . . . 3
1.1.2 Precision medicine . . . 5
1.2 Cancer Genomics . . . 5
1.2.1 What is cancer? . . . 5
1.2.2 Mutations in cancer . . . 6
1.2.3 Cancer as a pathway disease . . . 7
1.2.4 The development of targeted therapies . . . 8
1.3 GWAS . . . 10
1.3.1 GWAS for understanding the biology of complex diseases . . . 10
1.3.2 The notion of heritability . . . 10
1.3.3 The search for missing heritability . . . 11
1.3.4 Polygenic Risk Scores . . . 12
1.4 Statistical learning . . . 12
1.4.1 Statistical learning framework . . . 13
1.4.2 Generalisation . . . 13
1.4.3 The tradeoff between estimation and approximation errors . . . 14
1.4.4 The curse of dimensionality . . . 14
1.5 Learning in high dimension . . . 15
1.5.1 Regularisation techniques . . . 15
1.5.2 Transformation of the feature space . . . 17
1.5.3 Link between feature space transformation and regularisation . . . 20
1.6 Computational challenges . . . 22
1.6.1 The need for solvers dedicated to ℓ_1-regularised problems . . . 23
1.6.2 Available algorithms . . . 23
1.6.3 Active set algorithms and Screening techniques . . . 24
1.7 Contributions . . . 25
1.7.1 NetNorM . . . 25
1.7.2 Suquan . . . 27
1.7.3 WHInter . . . 29
1.7.4 Published work appearing in this thesis . . . 29

Abstract

This chapter introduces background relevant to the contributions presented in this thesis, both from the point of view of intended applications, and from a methodological point of view. In a first part, we give an overview of cancer genomics and of genome wide association studies (GWAS). These are the application fields towards which our contributions are mainly geared. In particular, we give insights into the molecular underpinnings of cancer and focus on the role played by tumour mutations, how they may impact important biological pathways, and how they may be targeted by new therapies. We also discuss central questions in GWAS, such as the missing heritability mystery, and we briefly highlight how the type of statistical methods used in the field has evolved. In a second part, we provide some statistical and computational background that is relevant to our contributions. We first introduce fundamental principles in machine learning, and then focus on frameworks to tackle high-dimensional learning problems. In particular, we introduce regularisation and feature transformation strategies. We also review algorithms and computational frameworks that allow to efficiently learn sparse models in high dimensions. We conclude this chapter by a presentation of our contributions.

Résumé

Ce chapitre permet de contextualiser les contributions de cette thèse, à la fois du point de vue méthodologique et du point de vue des applications envisagées. Une première partie introduit quelques notions relatives à la génomique du cancer et aux études d'associations pangénomiques (GWAS). Ce sont les deux domaines d'application majeurs vers lesquels nos contributions sont tournées. La génomique du cancer cherche à décrire les mécanismes moléculaires propres aux cellules cancéreuses. Nous nous concentrerons en particulier sur les mutations tumorales, sur leur impact au sein des voies biochimiques, et sur les thérapies qui permettent de les cibler. Nous discuterons également quelques questions centrales au GWAS, comme le mystère de l'héritabilité manquante, tout en soulignant comment les méthodes statistiques utilisées dans ce domaine tendent à évoluer. Dans un second temps, nous introduirons quelques principes fondamentaux de l'apprentissage statistique en général, avant de se pencher sur les techniques d'apprentissage en grande dimension, en particulier la régularisation et la transformation de l'espace de représentation des données. Pour terminer, nous nous intéresserons aux algorithmes qui permettent d'apprendre efficacement en grande dimension, notamment pour les modèles parcimonieux. Ce chapitre se termine par une présentation de nos contributions.

1.1 Contextual setting

1.1.1 Human genome sequencing: a bit of recent history

In year 2000, humankind crossed a new frontier. For the first time in history, a map of the human genome was revealed [Lander et al., 2001; Venter et al., 2001]. This achievement was the fruit of two concomitant endeavours, one of the publicly funded Human Genome Project (HGP) and one of the private company Celera Genomics led by Craig Venture. The Human Genome Project was a large international academic effort that started in 1990, and whose goal was to sequence the 3 billion base pairs of our DNA. These base pairs, also called nucleotides, are of four types, A, T, C and G, and are the building blocks of DNA. This goal was partially reached in 2000, with roughly 90% of the genome sequenced, and officially finished in 2003, thirteen years after its commencement, for a total cost of approximately 3 billion dollars. The first draft of the human genome was assembled from the DNA of a few donors. It was not the genome of one particular individual, nor was it representative of the genetic diversity of our species. It was the first *reference genome* for *Homo Sapiens*.

While most of the DNA sequence between any two humans is identical, no two individuals have the same genome, except maybe monozygotic twins. Single Nucleotide Polymorphisms (SNPs) are the most common type of sequence variation. They are positions in DNA at which nucleotides vary according to individuals. In principle, SNPs could be bi-, tri- or tetra-allelic, depending on how many variants exist in a population. However, the vast majority of SNPs are bi-allelic, i.e, only two alleles are frequently found in the population. As markers defining one's unique genetic identity, SNPs constitute essential keys to understand variations among individuals, such as susceptibility to disease. For this reason, the HapMap project [The International HapMap Consortium, 2003, 2005] was launched in 2002 with the purpose of mapping the genetic diversity between individuals. The International HapMap Consortium notably produced a database with more than 1 millions SNPs genotypes, some of them already known and some new ones, identified from the DNA of 269 individuals. This database notably allowed to characterise the correlation structure between neighbouring SNPs, known as *linkage disequilibrium*, with unprecedented accuracy. The HapMap project was followed by the 1000 genomes project, launched in 2008 and completed in 2015, which further described the human genetic variations map [The 1000 Genomes Project Consortium et al., 2015]. Based on genome sequencing and genotyping array experiments, this project describes more than 88 millions variations (80 millions of which are rare, i.e, occur at a frequency under 5% in the population) in human DNA in roughly 2500 individuals from many different ethnic origins.

The fast pace at which our genomes have been explored and catalogued in the last twenty years is the result of technological breakthroughs that have accelerated DNA sequencing, while reducing costs (see Mardis [2017] for a review). The cost for sequencing a human genome has been decreasing since 2001 faster than Moore's law (Fig. 1.1). It has notably plummeted around 2008 with the mass arrival of the second generation sequencing, also known as the *next-generation sequencing* (NGS), which relies on massively parallel sequencing of short DNA fragments. The first generation sequencing technologies, with which The Human Genome Project has been conducted, are based on Sanger's chain termination sequencing method. To date, there is a tough competition between various private companies to develop faster, cheaper and better sequencers. A third generation of sequencing technologies is under development, although it

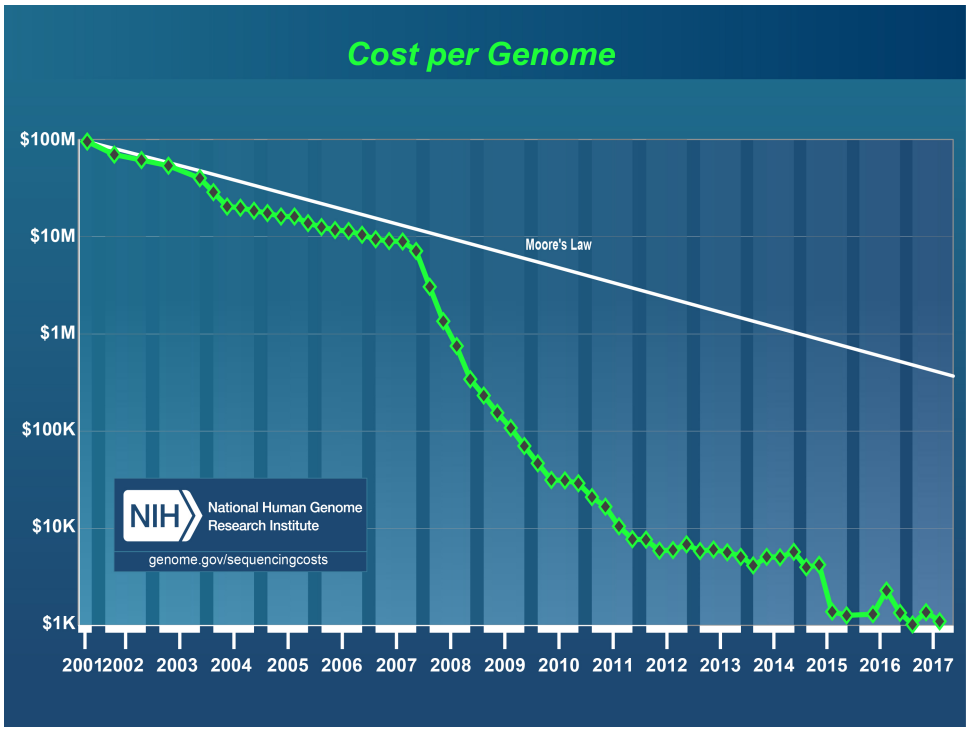


Figure 1.1 – Cost for sequencing a human-sized genome at a given coverage according to the National Human Genome Research Institute (NHGRI). The assumed coverage depends on the platform, a lower coverage is assumed for platforms that output longer reads. The reported cost is intended to comprehensively represent the total sequencing cost, including the depreciation of sequencing machines, the consumables and reagents, the bioinformatics post-processing, and management.

cannot be broadly used yet because of high error rates. This third generation relies on different technology breakthroughs according to the companies but is basically characterised by a direct sequencing of single DNA molecules where previous technologies needed to amplify, i.e, replicate many times the DNA fragments. These new technologies promise to sequence much longer fragments of DNA than previous technologies at a fast pace.

1.1.2 Precision medicine

Precision medicine takes into consideration the genetic background of an individual, together with its usual medical record, to better guide the choice of treatments and dosage. As sequencing costs continue to descrease, the onset of precision medicine carries great hopes to reduce the pervasiveness of ineffective treatments and adverse effects (see Ashley [2016] for a review).

Precision medicine has already shown great promise in oncology where targeted therapies, designed to be effective on cancer cells that carry specific genomic alterations, have improved survival for a number of cancer types. Its adoption in routine clinical practice is however only incipient in general. For example, the choice of drug dosage in prescriptions is mostly based on a patient's weight. While it may be appropriate for some drugs, it has now been known for decades that the ability of an individual to metabolise a drug is influenced by its genetic background. The same applies for the onset of side effects, or the efficacy of a drug. In the long run, the use of refined diagnostic testing could reduce the proportion of inapropriate treatments, which are both deleterious for a person's health and costly for the society.

Precision medicine is not the only perspective offered by the vast amount of biological data now available. Prevention, i.e, the evaluation of one's risk to develop a disease and the ability to pose early diagnosis, is another important aspect of the transformation incurred by the 'omics' technologies. The aim in this case is to extract predictive signatures from the data, indicative of certain diseases, so as to adapt patient monitoring if appropriate, or perhaps consider preventive care if the level of risk justifies it.

1.2 Cancer Genomics

Cancer genomics aims at identifying in a given tumour the genetic alterations that are responsible for the onset and development of cancer, and to understand how two cancers are molecularly related. The ability to read and understand cancer genomes is key to identify, within and across cancer types, the subgroups of patients who are likely to benefit from a therapy. This is important for both improving treatments efficacy and increasing the success rates of clinical trials by targeting the right patients.

1.2.1 What is cancer?

Cancer is among the leading causes of premature death in the world. In France, the estimated number of new cancer cases in 2017 is around 400,000. The mechanisms that lead to the onset of cancer are not fully understood yet. However it is widely agreed that genome instability is a common denominator to all cancer types, and that the disease is driven by genetic alterations in cancer cells that induce uncontrolled cell proliferation. This fast and abnormal multiplication of cancer cells in turn leads to the formation of tumours and allows them to invade neighbouring

tissues. It is usually the dysfunction of vital organs due to the invasion of cancer cells that can lead to death.

Cancers are usually classified depending on their tissue of origin, such as breast, lung or colon-rectum to cite the most frequent ones to date. However this rough classification does not reflect the highly heterogeneous nature of the disease. Indeed, the fine characterisation of tumours at a molecular level has underlined a wide spectrum of genetic alterations within cancer types. During the last decade, large initiatives involving thousands of cancer patients across many different cancer localisations, such as The Cancer Genome Atlas (TCGA) and the International Cancer Genome Consortium (ICGC), have shown that many cancer types can be further divided into several molecular subtypes. Some of these subtypes are already taken into account in clinical practice. This is for example the case of breast cancer, which early studies [Perou et al., 2000; Sorlie et al., 2001] have subtyped into 5 different groups, correlated with significantly different overall survival, based on microarray gene expression data. To date, three biomarkers related to these subtypes are routinely tested in clinical practice to guide the choice of an adapted treatment, i.e, the presence of estrogen and progesterone receptors at the surface of the cancer cells, and the excess of HER2 proteins. While the within cancer type heterogeneity at the molecular level is well established, recent studies have also highlighted that shared genomic alterations exist across cancer types, independently of the tissue of origin [Ciriello et al., 2013; Hoadley et al., 2014]. These observations underpin the importance of molecular profiling for better understanding cancer aetiology and defining better clinical strategies.

1.2.2 Mutations in cancer

Cancer results from a series of genetic alterations, and in particular somatic mutations, that accumulate in healthy cells. Somatic mutations, by opposition to germline mutations, occur during one's lifetime and do not affect germ cells. Therefore they are not inherited from parents, and cannot be passed on to offspring. While the term mutation can be used to refer to large scale DNA alterations such as copy number variations and gene fusions, in the sequel we will use this term to specifically refer to small scale alterations of DNA, i.e point mutations, small insertions and small deletions. Point mutations correspond to the substitution of a nucleotide by another, while small insertions (resp. deletions) correspond to the insertion (resp. deletion) of one or more nucleotides in the original DNA sequence. When they occur in protein coding genes, mutations are classified into different types according to their impact on the protein:

- *Synonymous mutations* do not modify the protein.
- *Missense mutations* produce proteins where one amino acid has been substituted for another.
- *Nonsense mutations* create a premature stop codon, and therefore a truncated protein .
- *Frameshift mutations* change the reading frame, which result in proteins with a totally different amino acid sequence, and a different length.

Mutations naturally occur in a lifetime and accumulate with age. Fortunately, most of them will not transform normal cells into cancer cells. They are caused by both endogenous and exogenous processes, such as inaccurate DNA replication during mitosis, defective DNA repair

machinery, or exposition to mutagens. Our current understanding of the mutational processes at work in cancer cells is quite rudimentary. However some clear mutational signatures have been highlighted, such as for example that of UV light in melanoma tumours, or that of tobacco smoke in lung tumours [Alexandrov et al., 2013]. The number of mutations in protein coding genes widely varies across cancers, from a few dozens to thousands [Martinocorena and Campbell, 2015]. A central topic in cancer research is to distinguish, among these mutations, those which play a role in promoting the proliferation of cancer cells, called *drivers*, versus those which don't, called *passengers* (see Raphael et al. [2014] for a review). Driver mutations typically occur in genes that promote cell division, known as *proto-oncogenes*, or genes that inhibit cell division, known as *tumour-suppressor* genes. Tumour-suppressor genes are generally deactivated via *loss-of-function* driver mutations that make the protein non-functional. By contrast, proto-oncogenes are generally activated into oncogenes via *gain-of-function* mutations, which confers a new or enhanced function to the protein. Since the alteration of most positions in a gene results in a non-functional protein, mutations in oncogenes tend to be clustered into hotspots, while they are more uniformly spread in tumour-suppressor genes (Fig. 1.2) [Vogelstein et al., 2013]. Recent studies have estimated that cancer cells have on average between 1 and 10 driver mutations depending on cancer types [Martincorena et al., 2017; Tomasetti et al., 2015]. This means that the vast majority of mutations are actually passengers. A Cancer Gene Census [Futreal et al., 2004] has been established to catalogue *cancer genes*, defined as the genes carrying driver mutations. To date, it contains around 300 genes for which there is strong evidence that they are drivers. Most of them are mutated at intermediate frequency across tumours, between 2% and 20%. However, it is expected that as more and more tumours are sequenced, many more driver genes will be discovered [Lawrence et al., 2014; Martincorena et al., 2017]. The fact that drivers occur in many different genes at intermediate or low frequency poses important challenges for the identification of drivers in an individual tumour, and consequently, for mutation based stratification of patients and the identification of the causal mechanisms implied in any given tumour.

1.2.3 Cancer as a pathway disease

The heterogeneity of driver mutations across tumours, even within one tissue, highlights the complexity of cancer aetiology and the challenges ahead to design molecular therapies effective in a sufficiently large number of patients. From a bird's-eye perspective, it is nonetheless possible to capture patterns in the hodgepodge of driver mutations, by taking into account the pathways within which they interact. Biological pathways describe biochemical cascades, mediated by a number of proteins, that convert stimuli (such as hormones or growth factors) into the appropriate cellular responses. The proper functioning of a pathway can be seen a 'teamwork', i.e, one non-functional protein in the cascade suffices to produce an aberrant cellular response. As a consequence, several driver mutations in different cancer genes, but in the same pathway, can have similar downstream effects. In practice, several well studied signalling cascades have been shown to be frequently mutated in cancer cells. One example is the PI3K pathway, depicted in Fig. 1.3, which notably controls cell growth and survival. This pathway includes a number of recurrently mutated genes in cancer, marked by red asterisks, among which the commonly mutated oncogene PIK3CA (coding for the protein p110α) and tumour suppressor gene PTEN. The gain-of-function mutations in PIK3CA (see Fig. 1.2) and the loss-of-function mutations in

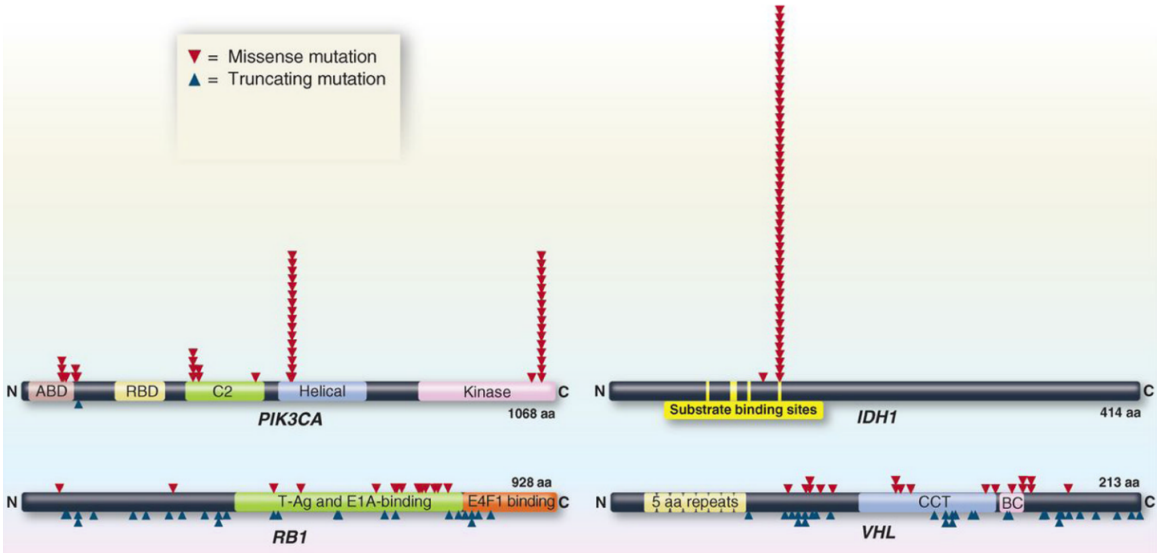


Figure 1.2 – Patterns of somatic mutations in oncogenes (PIK3CA, IDH1) and tumour suppressor genes (RB1, VHL). The figure was taken from [Vogelstein et al., 2013].

PTEN have both been shown to incur a constitutive activation of the PI3K pathway [Samuels and Waldman, 2010], promoting neoplasia. Characterised pathways usually involve a few dozens of genes whose biochemical interactions have been characterised extensively. They therefore reflect small scale but highly confident and well characterised interactions between genes. Pathways represent precious units of biological knowledge, however they should not be seen as independent units: they overlap and communicate with each other within a vast intricate network of genes. Protein-protein interaction networks overcome this limitation, at the price of being incomplete and noisy. These networks represent general interaction between genes, without precise knowledge of the nature of each interaction. They are derived from one or several sources including classical experimental results, high-throughput experiments, and literature curation. A number of recent works have leveraged pathways and gene networks to identify new cancer drivers (see Creixell et al. [2015]; Dimitrakopoulos and Beerenwinkel [2017]; Raphael et al. [2014] for a review) and to stratify cohorts of patients into relatively homogeneous subtypes [Hofree et al., 2013].

1.2.4 The development of targeted therapies

The fine characterisation of tumours at a molecular level has already started to improve the standard-of-care in oncology. Since 2000, many targeted therapies have been introduced into clinical practice. These therapies represent a paradigm shift compared to conventional chemotherapies. Indeed, chemotherapies are aimed at killing all rapidly dividing cells, while targeted therapies act on proteins that are specific to cancer cells, such as mutated proteins, fusion proteins, or proteins that are over-expressed in cancer cells. Targeted therapies are in general small molecules or monoclonal antibodies. The former can act on proteins inside the cells, while the latter cannot pass the cell membrane but act on proteins at the cell surface. The biological activity of these drugs is mediated by a variety of mechanisms of action. Let's take an example

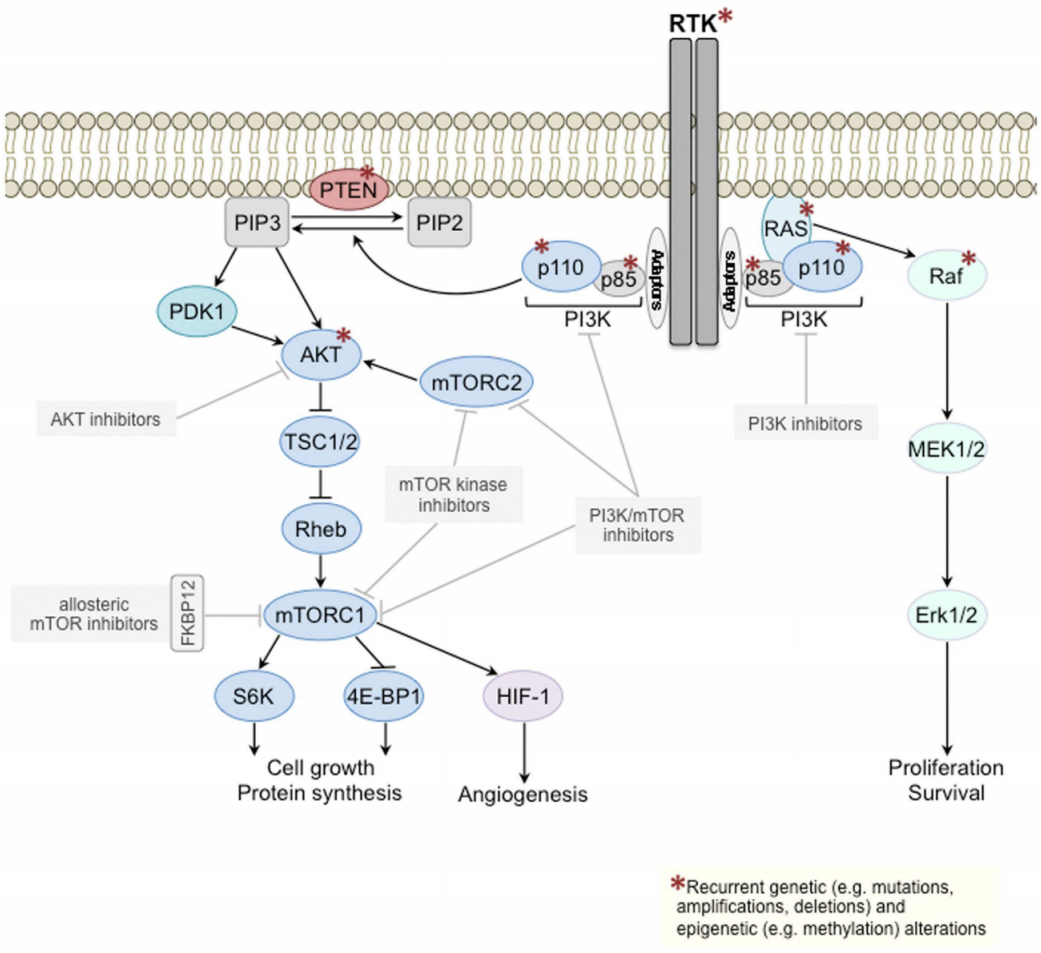


Figure 1.3 – **Partial view of the PI3K signalling pathway.** The figure was taken from Weigelt and Downward [2012].

for illustration purposes. In 2002, Davies et al. [2002] discovered that approximately half of melanomas were characterised by a mutation in the BRAF gene, and that most of these mutations were a substitution of valine with glutamic acid at position 600 (V600E). This mutation was shown to constitutively activate the MAPK pathway which in turn spurs cellular proliferation. After years of research and development, a targeted therapy called Vemurafenib (commercial name Zelboraf) [Chapman et al., 2011] eventually reached the Food and Drug Administration (FDA) approval in 2011. This molecule specifically targets the protein kinase BRAF with V600E mutation. It acts by binding to the Adenosine triphosphate (ATP) binding site of the mutant BRAF, thereby preventing it from activating downstream signalling. To date, many targeted therapies (around a hundred) have already been approved by the FDA.

1.3 GWAS

Genome-wide association studies (GWASs) attempt to identify SNPs that contribute to a given phenotype in a population, for example a disease. The development of relatively cheap SNPs arrays, which allow to genotype an individual for a predefined set of loci, has been crucial to the onset of GWASs. These arrays, to date, typically contain from 200,000 to 2,000,000 SNPs. The first large GWAS study, involving 14, 000 individual with one of seven common diseases such as type 1 and type 2 diabetes, as well as 3, 000 control individuals, dates back to 2007 [The Wellcome Trust Case Control Consortium, 2007]. Since then, many GWASs have been conducted for thousands of complex traits (height, schizophrenia, ...) and over 10,000 associations have been reported [Welter et al., 2014].

1.3.1 GWAS for understanding the biology of complex diseases

The purposes of GWAS studies, beyond the search for associated loci, are multiple. Primarily, these studies are aimed at facilitating the identification of the underpinnings of complex diseases and ultimately driving translational advances. In the last decade, GWASs have successfully facilitated the discovery of biological mechanisms involved in several diseases (see Visscher et al. [2017] for a review). One famous example is the discovery, through GWAS, of a SNP within the *Complement Factor H* (CMH) gene that conveys a significant increased risk in developing age-related macular degeneration (AMD) [Klein et al., 2005]. The biological insight gained through this discovery has fuelled the development of a number of therapeutics that are today at preclinical or clinical stages (see Black and Clark [2015] for a review). Nonetheless, GWAS are today facing criticisms regarding its primary purpose. These criticisms notably point the difficulty to go from GWAS results to the identification of causal SNPs, and the fact that the vast majority of the discovered associations have small effects, i.e, correspond to small increased risk to develop a disease.

1.3.2 The notion of heritability

This last criticism is linked to another important purpose of GWAS studies, i.e, disentangling the proportion of the total phenotypic variance that is due to the genotype, as opposed to the environment. This is the *nature versus nurture* debate. In genetics, the proportion of total phenotypic variance that is due to the genotype is called the *broad-sense heritability* H^2, while the proportion of the total phenotypic variance that is due to *additive genetic effects* is called

the *narrow-sense heritability* h^2 [Visscher et al., 2008]. While H^2 would be the quantity of most interest, h^2 is the quantity that is actually manipulated in genetics, mostly for practical reasons. Models for estimating heritability have existed for decades, long before the advent of SNP chips. These models typically estimate h^2 based on the observed resemblance, or phenotype correlation, between relatives [Tenesa and Haley, 2013]. Since h^2 is a proportion of the total phenotypic variance, it is necessarily a value between 0 and 1. An idea of typical values obtained for h^2 can be grasped from a recent study [Wang et al., 2017] which, based on insurance claims from 128,989 american families (parents and children), gives estimates of heritability for 149 diseases. Using a multivariate, generalised, linear mixed model taking into account shared environmental factors, they estimate for example $h^2 = 0.56$ for type 2 diabetes, or $h^2 = 0.46$ for general hypertension.

1.3.3 The search for missing heritability

At the time of the first large scale GWASs, researchers expected to pinpoint a few genetic variants that would explain a sizeable proportion of the heritability observed in family studies. However, except for a few phenotypes such as age-related macular degeneration, the results obtained fell short of expectations. In most studies, the variants identified as significantly associated with the phenotype could explain only a small proportion of the heritability. This observation gave birth to the concept of *missing heritability* [Maher, 2008; Manolio et al., 2009], which refers to the gap between h^2 estimated from family studies and h^2_{SNP} estimated from the SNPs. For example, a series of studies reported a proportion of heritability explained of 5% for height [Gudbjartsson et al., 2008; Lettre et al., 2008; Weedon et al., 2008] or 6% for type 2 diabetes [Zeggini et al., 2008]. The mystery of missing heritability has been the subject of much research since then, and a series of possible explanations and hypothesis have been formulated to solve it. Not long after the missing heritability problem was raised, Yang et al. [2010] posited that missing heritability was partly due to common SNPs whose effects are too small to reach statistical significance with 'traditional' GWAS methods. Indeed, traditional GWAS methods implement a battery of statistical tests, one for each locus, and selects a significance threshold that accounts for multiple testing, knowing that the number of tests to be performed is equal to the number of SNPs which can easily reach millions. To overcome this issue, they proposed the first method that jointly models the additive influence of all variants simultaneously, and reported with this new method that 45% of height heritability could actually be explained with common SNPs additive effects, compared to the 5% reported so far. This method has been applied to many traits since its publication and extensions and refinements are an active area of research. While the above mentioned method look for common variants with small effects, it has also been conjectured that the missing heritability would be due to rare variants with larger effects [Pritchard, 2001], where a variant is usually considered as rare if its frequency in a population is below 1%. Indeed, rare variants have been understudied because they are not tagged on conventional SNP chips. Recent findings concerning rare variants however suggest that these would also have small effects in general (see Auer and Lettre [2015] for a review), although the assessment of the rare variant hypothesis is clearly still underway. From a different point of view, it has also been proposed that missing heritability would be due to other types of genetic variations, and in particular copy number variations and epigenetic factors that are passed on from parent to children. Last but not least, one hypothesis states that the estimates of heritability could well be inflated,

thus creating more missing heritability than there really is. Indeed, estimates of h^2 could be inflated by the existence of non-additive effects such as epistasis [Hemani et al., 2013; Zuk et al., 2012], by shared familial environment if not properly taken into account, or by gene-environment interaction or correlation. Epistasis [Phillips, 2008] refers to genetic interactions among loci, i.e, to events whereby the effect of one locus depends on the genotype at another locus. Epistasis phenomena have been widely observed at the molecular level, where gene products are known to act within pathways. Overall, it is difficult to date to draw a consensus about where the missing heritability lies, or if it is even missing.

1.3.4 Polygenic Risk Scores

In order to further assess the explanatory power of SNPs, models that shift the objective of loci identification to accurate phenotype predictions have also emerged. These models, called *polygenic risk scores* (PRS), are constructed as weighted linear combinations of SNPS and are aimed at accurately predicting one's phenotype based on its genotype. PRS generate a growing interest since compared to more traditional approaches, they offer another way of measuring how much genetic signal there is in a dataset, whether or not variants could be significantly associated to the phenotype [Dudbridge, 2013]. Moreover, if sufficiently accurate, they also provide unprecedented tools to clinically evaluate one's risk to develop a certain disease and to set up more informed, and more personalised medical care. For example, in a study gathering more than 30,000 breast cancer cases and as many controls, Mavaddat et al. [2015] calculated the genetic risk of developing breast cancer in a lifetime based on a 77-SNPs polygenic risk score. They showed that this risk was 3.5% for women below the 1^{st} percentile of the PRS and 29% for those above the 99^{th} percentile. These are to be compared with the lifetime risk of a women to be diagnosed with breast cancer, which is 12% according to recent statistics [Howlader et al., 2017]. In France for example, a mammography screening is systematically proposed every two years to all women aged between 50 and 74. One could imagine to propose this screening not only based on the age but also on the PRS in order to achieve a better harms-benefit balance between undesirable side effects, the importance of an early diagnosis, and the screening costs.

1.4 Statistical learning

Mutations, SNPs, gene expression and other 'omics' data types are naturally represented by many variables. Mutation and expression data are typically represented by around 20,000 variables, one for each gene, representing the mutation status of a gene or its expression level. SNPs datasets come with hundreds of thousands or millions of variables, one per position assessed in the genome. This is generally much more than the number of samples available. Indeed, in cancer genomics the number of tumours per cancer type for which there is molecular data available is generally in the hundreds, sometimes in the thousands. For example, The Cancer Genome Atlas (TCGA) has characterised 11,000 tumours from 33 cancer types. For GWAS studies, the cohorts are usually larger, and easily in the thousands or dozens of thousands. Prediction problems based on such data are therefore high dimensional, i.e, there are more parameters to be estimated than samples. In machine learning, this setting is known as the small n large p problem, and a number of frameworks have been developed to handle it. In this section, we aim at introducing this problem from a statistical learning perspective. We then present an overview

of techniques that are useful in high dimensional problems, and finally we briefly bring to light computational challenges relative to these techniques. Moreover, in this section we will mainly focus on *supervised* learning (by opposition to unsupervised learning, semi-supervised learning, ...) since the contributions of this thesis mostly involve supervised problems. In supervised problems, the learner is provided with both examples and corresponding outputs, and its goal is to find a relationship between the two. It is said to be supervised since the predictions of the learner can be compared to the observed outputs and improved based on this feedback.

1.4.1 Statistical learning framework

We will denote by $\mathcal{X}$ the input space and $\mathcal{Y}$ the output space. In statistical learning we assume that the data is generated following a joint probability distribution $\mathcal{P}$ on $\mathcal{X} \times \mathcal{Y}$, which is unknown. What we do observe is the realisation of n pairs of independent and identically distributed (in short, *iid*) random variables $(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathcal{X} \times \mathcal{Y}$ following the probability distribution $\mathcal{P}$.

The goal of supervised learning is to find a mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ which estimates 'as well as possible' an output $y \in \mathcal{Y}$ given an observed input $x \in \mathcal{X}$. The notion of 'how good' a mapping is at predicting the output for a given input is quantified thanks to a *loss function* which has low values whenever the prediction is close to the observed output and high values otherwise. For regression problems, i.e., when $\mathcal{Y} = \mathbb{R}$, the squared loss is widely used and is defined as:

$$\forall (y, y') \in \mathbb{R}^2, \quad l(y, y') = \frac{1}{2}(y - y')^2.$$

Given a loss function l, the best mapping that can be obtained is the one that minimises the *risk*:

$$R(f) = \mathbb{E}(l(f(X), Y)),$$

i.e, the one that minimises the expected loss over all points following the distribution $\mathcal{P}$. However, since $\mathcal{P}$ is unknown, it is practically impossible to find the mapping that minimises the risk. This is why in practice the *empirical risk* is considered instead of the true risk. It is an approximation of the true risk which is defined as the average loss over all observed data points, i.e:

$$R_{emp}(f) = \frac{1}{n} \sum_{i=1}^{n} l(f(X_i), Y_i).$$

Looking for a mapping f which minimises the empirical risk $R_{emp}(f)$ is known as the *empirical risk minimisation* (ERM) principle.

1.4.2 Generalisation

Let $\mathcal{F}$ denote a set of functions with input space $\mathcal{X}$ and output space $\mathcal{Y}$. We will denote by $f_{\mathcal{F}}$ an optimal mapping in $\mathcal{F}$, i.e,

$$f_{\mathcal{F}} \in \operatorname*{argmin}_{f \in \mathcal{F}} R(f)\,.$$

For a given set of training points $(X_i, Y_i)_{i=1,\ldots,n}$, we define f_n as a minimiser of the empirical risk, i.e,

$$f_n \in \operatorname*{argmin}_{f \in \mathcal{F}} R_{emp}(f).$$

While f_n minimises the empirical risk based on the available training data, what we are really interested in is the quality of the predictions on unseen data, i.e, the generalisation ability of f_n. Formally, a predictor f_n *generalises* well if its risk $R(f_n)$ is small. If $\mathcal{F}$ is taken as $\mathcal{F}_{all}$, i.e, the space of all measurable functions with input space $\mathcal{X}$ and output space $\mathcal{Y}$, then the empirical risk minimisation principle does not yield predictors that generalise well. For example, in the case of regression, it is always possible to construct a mapping f_n:

$$f_n(x) = \begin{cases} Y_i \text{ if } x = X_i \\ \text{any value otherwise}\,, \end{cases} \tag{1.1}$$

such that the empirical risk is exactly equal to zero for any number of samples n while the true risk is nonzero and arbitrarily high. Such functions only *memorise* the training points without learning anything about the underlying distribution. This illustrates why the complexity of the space $\mathcal{F}$ of functions from which f_n will be chosen needs to be controlled. The choice of $\mathcal{F}$ will typically encode assumptions we are willing to make about the data, or some prior knowledge. The most general assumption that the vast majority of learners rely on is smoothness, with the idea that small changes in the input should lead to small changes in the output. It is also common to rely on stronger assumptions, and for example choose $\mathcal{F}$ as the set of all linear functions. Overall, it is not possible to build a successful predictor without making any assumptions about the underlying probability distribution $P(\mathcal{X}, \mathcal{Y})$. This is in essence the message conveyed by the *no free lunch theorem* [Wolpert, 2002; Wolpert and Macready, 1997].

1.4.3 The tradeoff between estimation and approximation errors

In practice, the complexity of the function space $\mathcal{F}$ should be chosen small enough so that the learned predictor generalises well, but big enough so that the relationship between the inputs and the outputs can be well approximated. This tradeoff can be highlighted by decomposing the overall error $R(f_n) - R(f_{\mathcal{F}_{all}})$ as:

$$R(f_n) - R(f_{\mathcal{F}_{all}}) = \underbrace{R(f_n) - R(f_{\mathcal{F}})}_{\text{estimation error}} + \underbrace{R(f_{\mathcal{F}}) - R(f_{\mathcal{F}_{all}})}_{\text{approximation error}}\,.$$

The *approximation error* corresponds to the error made by approximating the best possible predictor $f_{\mathcal{F}_{all}}$ by a predictor $f_{\mathcal{F}}$ which is the best one in a relatively simple function space $\mathcal{F}$, while the estimation error reflects the difficulty of estimating the best possible predictor in $\mathcal{F}$ based on a limited number of training points. This tradeoff between estimation and approximation error is similar to the *bias-variance* tradeoff in statistics.

1.4.4 The curse of dimensionality

High-dimensional data is commonplace in computational biology, but also in other fields such as astronomy or finance. High-dimensional datasets are characterised by a number of predictors exceeding the number of samples, i.e, $n << p$. In such settings, learning can be impaired by a phenomenon commonly referred to as *the curse of dimensionality* [Bellman, 1961]. The curse of dimensionality stems from the fact that a high-dimensional space is very sparsely populated by the training points. To see this, let's consider a unit hypercube in dimension p over which training points are drawn from a uniform distribution. Assume we bin each coordinate in n_b

intervals of equal length. This corresponds to binning the p-dimensional space according to n_b^p hypercubes. Therefore, in order to have on average one training point falling in each small hypercube, we would need a number of samples that grows exponentially with p, i.e, n_b^p. This is practically infeasible even if the bins are large. Indeed, if each coordinate were binned into two equal parts for a problem where $p = 30$, we would need 2^{30} samples, i.e, more than one billion, to have on average one sample per small hypercube. In fact, the edge length of a small hypercube which would cover in expectation 1% of the training points would be equal to $e(p) = \sqrt[p]{0.01}$. For example in dimension 30, $e(30) \approx 0.86$, i.e, a bin should span 86% of the values over which coordinates vary. As a consequence, a model in a high dimensional space is doomed to make predictions based on neighbourhoods that are necessarily large in at least one dimension.
Despite of this, learning is still possible in most cases. This is true for mostly two reasons. First, real data tends to exhibit a certain degree of smoothness which implies that the output cannot vary too quickly with the input. Second, real data usually lies on a manifold of lower effective dimensionality than the ambiant space, i.e, the probability mass of the data points is not uniformy spread over the whole space but rather concentrated near a manifold. In the following sections, we present learning techniques that take advantage of these properties to produce good predictors.

1.5 Learning in high dimension

The curse of dimensionality raises important issues in statistical learning. Indeed, when the input space is very sparsely populated by training points, interpolation like methods are prone to overfitting. There are however techniques designed to overcome this issue. These techniques are roughly based on two complementary ideas. One is to reduce the complexity of the function space $\mathcal{F}$ sufficiently so that interpolation and extrapolation based on example points lead to some generalisation. Another is to find a transformation of the input space in which simple predictors such as linear predictors perform well. Hereafter we review classical methods from these two categories.

1.5.1 Regularisation techniques

Formulation

Playing on definition of the function space $\mathcal{F}$ over which empirical risk minimisation is performed is the most obvious lever to control the complexity of $\mathcal{F}$. For example, the space of linear functions is a subspace of the space of polynomial functions of degree $k \in \mathbb{N}$ which can fit more complex relationships between inputs and outputs. However, playing on the family of functions over which to optimise is usually not very practical and do not offer a precise control over the level of complexity. Instead, *regularisation techniques* are most often used. It consists in minimising the regularised empirical risk:

$$R_{reg}(f) = R_{emp}(f) + \lambda\Omega(f) \ ,$$

where the function $\Omega(f)$ is called the regulariser. The regulariser is designed so as to penalise complex functions, such as for example functions that vary too rapidly over the input space. It is typically a norm. The parameter $\lambda \in \mathbb{R}^+$ is called the regularisation parameter and controls the

balance between the approximation and estimation error. The larger λ the more emphasis there is on minimising the complexity of the model rather than the empirical risk. Said differently, increasing λ amounts to learning a predictor f_n from a function space $\mathcal{F}$ with smaller complexity.

Controlling the complexity of a model via a regulariser presumes making assumptions about the true predictor. For example, using an ℓ_1 norm regularisation presumes that only a fraction of the features are predictive. The LASSO and ridge regression encode assumptions that are rather generic and that can be appropriate for many problems. However in some cases, we have more specific prior knowledge about the properties that a good predictor should satisfy. If such is the case, the regulariser is a good place to encode this prior knowledge.

Ridge regression and the LASSO

Ridge regression [Hoerl and Kennard, 1970], initially invented under the name Tikhonov regularisation [Tikhonov, 1943], and the Least Absolute Shrinkage and Selection Operator (LASSO) [Tibshirani, 1996] are two popular regularised linear models. Let $(\boldsymbol{x}_i, y_i)_{i\in[\![n]\!]} \in (\mathbb{R}^p \times \mathbb{R})$ be n training points. The ridge estimate solves:

$$\hat{\boldsymbol{w}}^{ridge}(\lambda) = \underset{\boldsymbol{w}\in\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i - b\right)^2 + \lambda \| \boldsymbol{w} \|_2^2 , \tag{1.2}$$

where $\mathcal{F}$ is chosen as the set of linear functions $f : \boldsymbol{x} \mapsto \boldsymbol{w}^\top \boldsymbol{x} + b$, $\boldsymbol{w} \in \mathbb{R}^p, b \in \mathbb{R}$, the loss function is the squared loss and the penalty term is the ℓ_2 norm of $\boldsymbol{w}$, i.e, $\| \boldsymbol{w} \|_2^2 = \sum_{j=1}^p w_j^2$. The ridge penalty forces the coefficients of $\hat{\boldsymbol{w}}^{ridge}$ to be shrunk towards zero when λ increases. The LASSO differs from ridge regression in that the regulariser is the ℓ_1 norm of $\boldsymbol{w}$, i.e, $\| \boldsymbol{w} \| = \sum_{j=1}^p | w_j |$, instead of the ℓ_2 norm:

$$\hat{\boldsymbol{w}}^{LASSO}(\lambda) \in \underset{\boldsymbol{w}\in\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i - b\right)^2 + \lambda \| \boldsymbol{w} \|_1 . \tag{1.3}$$

If λ is sufficiently large, this penalty forces coefficients in $\hat{\boldsymbol{w}}^{LASSO}$ to be equal to zero. To see how the penalty terms control the complexity in these two models, we can derive the expression of the degrees of freedom for both models as a function of λ. The degrees of freedom of a model are the 'effective number of parameters in the model', i.e, the number of dimensions over which the predictions can vary. For this purpose, let us introduce some notations. Let $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ be the data matrix where sample $\boldsymbol{x}_i \in \mathbb{R}^p$ is the i^{th} row of the matrix $\boldsymbol{X}$. Let $(d_j)_{j\in[\![p]\!]}$ be the set of singular values of $\boldsymbol{X}$, and consider a fixed regularisation parameter λ. Then the degrees of freedom for ridge regression is [Hastie et al., 2001, chap. 3]:

$$df_{Ridge}(\lambda) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda} . \tag{1.4}$$

For the LASSO, we assume that the response vector $y \in \mathbb{R}^n$ follows a normal distribution $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$ with mean $\mu \in \mathbb{R}_n$ and spherical covariance. Based on this assumption, theorem 2 in [Tibshirani and Taylor, 2012] state that:

$$df_{LASSO}(\lambda) = \mathbb{E}(\operatorname{rank}(\boldsymbol{X}_{\mathcal{A}})) , \tag{1.5}$$

where $\mathcal{A}$ is the active set corresponding to any solution $\hat{\boldsymbol{w}}^{LASSO}$, i.e, $\mathcal{A} = \{i \in [\![p]\!] : \hat{w}_i^{LASSO} \neq 0\}$. In the case of ridge regression, it is easy to see that the degrees of freedom decrease with λ. In the case of the LASSO, the degrees of freedom equals the expected dimension of the subspace spanned by the columns of $\boldsymbol{X}$ in $\mathcal{A}$. Since the larger λ the smaller $\mathcal{A}$, the degrees of freedom is also expected to decrease with λ. In practice the choice of λ is a difficult problem. It is most often chosen by cross-validation. Expressions (1.4) and (1.5) are interesting since they allow to explicitly quantify how the complexity of a function space can be controlled by varying λ. Of note, we focused here on the degrees of freedom to measure complexity but there are a number of other possibilities. In fact, there is not one universal measure of complexity that is used throughout all machine learning problems, but several, and the choice of which one to use depends on the problem at hand.

Kernel methods

Kernel methods are a group of algorithms that rely on the definition of a kernel function to learn predictors. The choice of a kernel function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ defines a similarity measure between any two samples from the input space. It can be shown that a positive definite kernel can be thought of as an inner product between two example points after they have been embedded in some Hilbert space $\mathcal{H}_k$ [Aronszajn, 1950]. As a consequence, learning with a kernel can be thought of as learning in some feature space $\mathcal{H}_k$, in which the inner product between two samples x and x' equals $k(x, x')$. Of note, the feature space does not need to be explicitly computed since kernel methods only deal with inner products in $\mathcal{H}_k$: this is the kernel trick. In fact, $\mathcal{H}_k$ can even be infinite dimensional. In supervised settings, kernel methods find the predictor f_n that solves the following problem:

$$\min_{f \in \mathcal{H}_k} R_{emp}(f) + \lambda \| f \|_{\mathcal{H}_k}^2 , \tag{1.6}$$

where k is a predefined positive definite kernel. The regulariser $\| f \|_{\mathcal{H}_k}^2$ controls the complexity of the function space $\mathcal{H}_k$ by encouraging the learned model f_n to be smooth in the implicit feature space $\mathcal{H}_k$ defined by the kernel. This means that two inputs that are close in $\mathcal{H}_k$ should have similar values $f_n(x)$. The predictor f_n will be a linear combination of the features in $\mathcal{H}_k$, corresponding to a non linear model in the original feature space. The possibility offered by kernel methods to incorporate prior knowledge via a kernel and learn non linear models while efficiently controlling the complexity of the model space explain their great popularity for high dimensional learning problems.

1.5.2 Transformation of the feature space

Penalised linear regression or kernel machines purely rely on interpolation between example points for generalisation. They compensate for the lack of example points in high dimensions by assuming that there exists good prediction functions that are sufficiently simple, be it in the original input space or in some transformation of the input space predefined by a kernel. Of course, such assumptions may not be verified and these models may not be able to capture the complexity of the relationships between inputs and outputs sufficiently well. In order to overcome this drawback, one idea is to find a transformation of the input space in which example points are related to their labels through a simple function, typically a linear function. This

is in fact equivalent to learning a new similarly measure between samples according to which simple methods such as penalised linear regression or kernel machines would be more effective. In particular, in such a transformed space one expects that points with similar labels are quite close to each other while points with dissimilar labels are further away. While the means to achieve such a goal differ according to the techniques, one can notice they all tend to extract *underlying explanatory factors* hidden in low level data, corresponding to *concepts* or *higher-level abstractions*. In the following paragraphs we describe three main categories of techniques for transforming a feature space: feature engineering, dimensionality reduction and manifold learning, and representation learning. Our goal here is not to provide an exhaustive catalogue of all existing methods in these three categories, but rather to describe the pursued objectives in all three cases and to highlight classical examples.

Feature engineering

Feature engineering consists in manually crafting new features, based on the original ones, that are thought to be predictive for the task at hand. It requires the intervention of a human expert who has a prior knowledge of the underlying explanatory factors hidden in the data, and who is going to design new features aimed at representing these underlying factors.

One classical example of successful feature engineering is the Scale-Invariant Feature Transform (SIFT) method [Lowe, 2004]. The development of SIFT descriptors, or SIFT features, has provoked a small revolution in the early 2000s in the computer vision field. The SIFT method starts by identifying, via the comparison of a pixel with its neighbours across different scales, interesting *keypoints* (i.e locations in the image) that are invariant to scale, and robust to small amounts of noise. These keypoints, which are already invariant to scale, are then assigned an orientation based on local gradients directions, so as to make them invariant to rotation. In a last step, a 128-dimensional feature is computed to describe each keypoint, again based on local gradient directions and magnitudes around the keypoint, and transformed to be partially invariant to variations in illumination and shape distortion. As can be seen from the description of the method, SIFT features were built with at their heart the idea that they should be invariant to the factors that make two different views of an object different. At the time it was published, this feature engineering approach dramatically improved state-of-the-art performances for tasks such as image matching, object recognition or motion tracking.

In genomics, the question of how to translate gene level measurements into pathway activation or deregulation scores is a hot topic. This question stems from the fact that many diseases are thought to be pathway diseases, i.e, diseases arising from the deregulation of pathways rather than from the alteration of specific genes. In such a context it is reasonable to assume that pathway-based predictors should outperform gene based ones. Feature engineering proposals have thus been imagined to combine gene level measurements into pathway activities features. A recent example is the Canonical Circuit Activity Analysis (CCAA) [Hidalgo et al., 2017]. This method transforms gene expression measurements into *circuits* scores, where a circuit comprises all paths between an input and an output node of a pathway, via a signal propagation algorithm that quantifies the amount of signal that can travel from the input to the output node. Jiao et al. [2017] have shown that the features generated via this approach could outperform conventional gene-level expression features for supervised breast cancer prognosis, which is quite encouraging .

Dimensionality reduction and manifold learning

Contrary to feature engineering, dimensionality reduction and manifold learning techniques are data driven and do not require the intervention of a human expert. They rely on the assumption that observed high-dimensional data points are not uniformly distributed over the input space, but instead concentrated near a manifold with much lower dimensionality than the ambient space. This assumption is most often satisfied for many types of high-dimensional data. This can be easily intuited with images. Indeed, if pixel values are drawn uniformly at random from the input space, then the resulting image is very likely to look like noise and very unlikely to represent anything close to a photograph. Manifold learning techniques are aimed at discovering a relatively small number of features, compared to the original dimensionality of the problem, which capture the the intrinsic structure of the data. Principal Components Analysis (PCA) is the most simple example of a manifold learning algorithm. It finds the linear subspace that best approximate a cloud of data points. PCA, together with other linear dimensionality reduction techniques such as Independent Component Analysis (ICA) or Non-negative Matrix Factorisation (NMF) are widely used for data exploration purposes, i.e, to identify underlying explanatory factors in the data. Zinovyev et al. [2013] review the applications of such methods to gene expression profiles in cancer. The features extracted from these profiles typically represent biological functions or technical bias, and can be used to enhance tumour subtype classification, diagnosis or prognosis.

While these techniques can be referred to as manifold learning techniques, this terminology most often refers to techniques that learn non linear manifolds. A majority of these techniques are unsupervised and try to project the data in a lower dimensional space while preserving local distances between data points, i.e, nearby points on the manifold are mapped to nearby points in the lower dimensional space. Pathifier [Drier et al., 2013] is one example of a method for the analysis of tumour gene expression profiles that rely on a non-linear dimensionality reduction technique, i.e, principle curves [Hastie and Stuetzle, 1989]. Principle curves are a non-linear generalisation of PCA. More precisely, Pathifier transforms a dataset where samples are represented by gene expression measurements into a pathway based representation of samples. For each pathway P defined by d_p genes, samples points are represented in the corresponding d_P-dimensional space, and the resulting cloud of points is then summarised by a principal curve, on which samples are projected. The Pathway Deregulation Score (PDS) of a sample is defined as the distance, computed along the principal curve, between the projection of the sample and the projection of normal reference samples.

Representation learning

A popular trend in machine learning consists in learning representations instead of using predefined feature spaces (e.g. kernels) or manually engineered features. The goal of representation learning is to automatically extract features from the raw data that make the subsequent learning task easy. The learned features are expected to identify and disentangle the underlying explanatory factors of the data without relying on costly and time consuming human intervention.

During the past decade, representation learning techniques and in particular deep learning have allowed several breakthroughs in terms of performance in domains such as object recognition, speech recognition and natural language processing (NLP) (see Bengio et al. [2013a] for

a review). For example, in object recognition tasks, deep convolutional neural networks have largely superseded the manually engineered SIFT features. The representations learned by these deep convolutional architectures were shown to capture concepts such as edges or textures in the first layers, and objects or parts of objects such as eyes or cats in deeper layers. In NLP, representation learning techniques have also imposed their supremacy over more traditional models such as N-grams or latent semantic analysis (LSA). Today, Word2Vec [Mikolov et al., 2013] yields among the best vector representations for words. It refers to one of two models, the continuous bag-of-words (CBOW) model or the skip-gram model. Both of these models are linear models whose architecture only contains one single hidden layer (no non-linearity applied). The CBOW model predicts which word is most likely to appear given a certain number of words which precedes and follows it in a sentence. By contrast, the skip-gram model predicts the words that are likely to surround a given word in a sentence. Both of these models take as input the one-hot encoding of words and implement a softmax regression to predict the output. The new vector representation of words learned by these models is the representation of words in the hidden layer. This representation was shown to be interesting since it not only encodes syntactic similarities between words but also semantic similarities. For example, simple arithmetic operations such as Vec('King') - Vec('Man') + Vec('Woman') yields a new vector which is closest to the vector that represents the word 'Queen' in the database. The representation learned with Word2Vec has been successfully used in various NLP tasks.

In general, learned representations are obtained by jointly optimising a simple predictor and the representation it is based on in a supervised fashion. The representation itself is modelled according to assumptions that one has about the data, for example, the assumption that the underlying explanatory factors can be hierarchically defined in terms of other lower level abstractions in the case of deep learning models. It has been shown experimentally that learned representations can help disentangle explanatory factors. Geometrically speaking, it has further been shown that this disentanglement corresponds to the unfolding of the manifold the data lies on and that, consequently, the distribution of the data in the transformed space is much closer to a uniform distribution than in the original input space [Bengio et al., 2013b]. In genomics, representation learning is not widely used as in some other application domains of machine learning. The two main uses of representation learning for biology related domains would be for biological images (histopathology, microscopy, ...) or the study of DNA sequences, for example, regulatory genomics (see Yue and Wang [2018] for a review). However, these are not the type of applications we are interested in in this thesis, since we aim at predicting phenotypes based on gene level measurements such as mutations, SNPs or gene expression. For these types of applications, the use of representation learning is only incipient. One of the main reasons for this stems from the fact that representation learning techniques need to be fed with relatively large amounts of data which are typically not available in such cases.

1.5.3 Link between feature space transformation and regularisation

While regularisation and feature transformation tackle high dimensional learning problems from two distinct perspectives, it is enlightening to draw links between the two approaches, and see how both of them can converge to similar predictors. For this purpose, we describe one example, relevant to computational biology applications, where regularisation and feature transformation are equivalent.

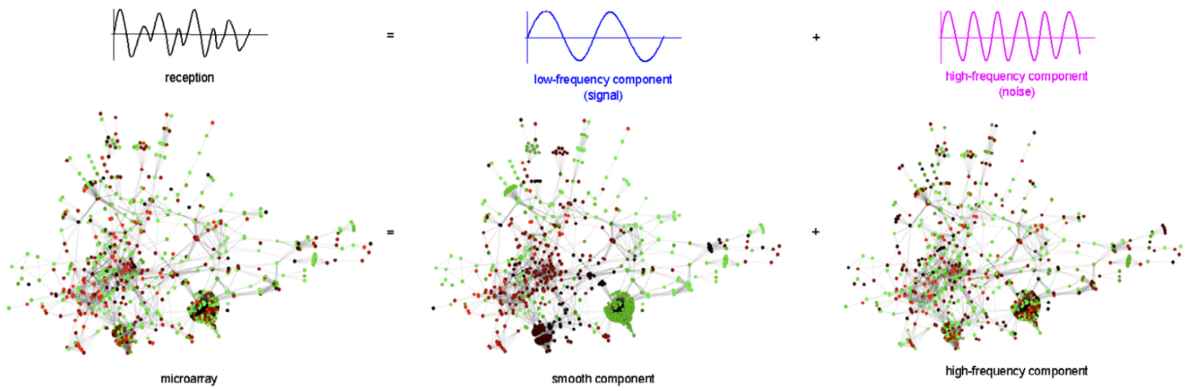


Figure 1.4 – Decomposition of a signal over a graph in smooth and high-frequency components. The figure was taken from [Rapaport et al., 2007]

Suppose that instead of using the data in the original input space, one is interested in considering the projections of the data points onto the columns of a given matrix $\boldsymbol{P} \in \mathbb{R}^{p \times p}$, i.e, one considers the set of data points $\{\boldsymbol{P}x_i\}_{i=1..n}$ instead of $\{x_i\}_{i=1..n}$. Learning a linear model with ℓ_2 regularisation in this new feature space amounts to solving:

$$\min_{\boldsymbol{w} \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} l(\boldsymbol{w}^\top \boldsymbol{P}\boldsymbol{x}_i,\, y_i) + \lambda \| \boldsymbol{w} \|_2^2 \,. \tag{1.7}$$

For any $\boldsymbol{w}$ in $\mathbb{R}^p$, let $\boldsymbol{v} = \boldsymbol{P}^\top \boldsymbol{w}$. We will denote by $\boldsymbol{P}^{-1}$ the inverse of $\boldsymbol{P}$ or its pseudo-inverse if need be. Simple calculations show that problem (1.7) is equivalent to:

$$\min_{\boldsymbol{v} \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} l(\boldsymbol{v}^\top \boldsymbol{x}_i, y_i) + \lambda \boldsymbol{v}^\top \boldsymbol{P}^{-2} \boldsymbol{v} \,, \tag{1.8}$$

i.e, to a linear model in the original input space, but with a regulariser that is more complex than the original ℓ_2 norm regulariser.
Let's now take an example to illustrate the relationship between these two formulations. Suppose features are related through an undirected graph $G = (V, E)$ which contains a set of p vertices V and a list of edges E. For example, if features were genes this graph could represent known interactions between genes [Rapaport et al., 2007], or if features were regions of fMRI images it could represent the correlation in the activity of different brain regions [Bullmore and Sporns, 2009]. For simplicity we assume that the weights associated to the edges are all equal to one. The graph G can be described by its normalised Laplacian $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ where $\boldsymbol{A} \in \mathbb{R}^{p \times p}$ is the adjacency matrix of the graph and $\boldsymbol{D} \in \mathbb{R}^{p \times p}$ is a diagonal matrix which represents the degree of each node in the graph. $\boldsymbol{L}$ is a symmetric and positive definite matrix [Chung, 1997] and can therefore be decomposed as $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Gamma}\boldsymbol{U}^\top$ where the matrix $\boldsymbol{U} \in \mathbb{R}^{p \times p}$ contains the normalised eigenvectors of the Laplacian as columns and $\boldsymbol{\Gamma} \in \mathbb{R}^{p \times p}$ is a diagonal matrix which contains its eigenvalues $0 \leq \gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_p$. The eigenvectors of the Laplacian have interesting properties since those associated to smaller eigenvalues (low frequency components) are smooth

over the graph while those associated to higher eigenvalues (high frequency components) tend to have values which vary more quickly over the graph. This notion is illustrated in Fig. 1.4. Given a graph, one could thus be interested in projecting the data on the eigenvectors of the Laplacian and choose $\boldsymbol{P} = \boldsymbol{L}^{-\frac{1}{2}}$ in (1.7), such that:

$$\forall i \in [\![n]\!]\,,\ \boldsymbol{P}\boldsymbol{x}_i = \boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{x}_i \tag{1.9}$$

$$= \sum_{\substack{j=1 \\ j:\gamma_j \neq 0}}^{p} \frac{\boldsymbol{U}_k^\top \boldsymbol{x}_i}{\sqrt{\gamma_j}} \boldsymbol{U}_k\,. \tag{1.10}$$

i.e, the high frequency components of x_i are attenuated compared to the low frequency ones. The projection of the data on $\boldsymbol{L}^{-\frac{1}{2}}$ therefore corresponds to a form of smoothing of the samples over the graph. If we now look at the induced regulariser in (1.8), it is easy to see that choosing $\boldsymbol{P} = \boldsymbol{L}^{-\frac{1}{2}}$ corresponds to choosing $\boldsymbol{P}^{-2} = \boldsymbol{L}$ which leads to a regulariser of the form:

$$\begin{aligned} \boldsymbol{v}^\top \boldsymbol{P}^{-2} \boldsymbol{v} &= \boldsymbol{v}^\top \boldsymbol{L} \boldsymbol{v} \\ &= \frac{1}{2} \sum_{i \sim j} \left(\frac{v_i}{\sqrt{\boldsymbol{D}_{ii}}} - \frac{v_j}{\sqrt{\boldsymbol{D}_{jj}}} \right)^2 , \end{aligned}$$

where the notation $i \sim j$ indicate the sum over all unordered pairs of nodes connected in the graph. From this expression one can see that this regulariser encourages the weights vector v to be smooth over the graph. This example illustrates the fact that regularisation and transformation of the feature space are two sides of the same coin. In particular in this example, ridge regression on smoothed data is equivalent to learning a linear model on the original data where the weight vector is encouraged to be smooth over a graph.

1.6 Computational challenges

ℓ_1 regularisation, and in particular the LASSO, enjoys a great popularity in high-dimensional statistics, and applications in genomics are no exception. This popularity is due to several factors. First, the sparsity promoted by the ℓ_1 penalty is coherent with the common expectation that only a few variables should be relevant to the problem at hand. Secondly, it provides interpretable models, which is essential if one wants to be able to explain the discriminative information captured by a predictor. Last but not least, the wide use of these models is fostered by the availability of fast and easy-to-use solvers, which have been developed over the last two decades. In this section, we would like to underline the computational advances that led to the current success of ℓ_1 penalties. For clarity, we will focus on the LASSO although much of the content could easily be translated to other losses than the quadratic loss. While much progress has been made since the original introduction of the LASSO in 1996, current research continues to provide improved solvers, which allow to deal with problems of ever larger scales.

1.6.1 The need for solvers dedicated to ℓ_1-regularised problems

The best subset regression problem seeks the subset of features, of cardinality at most k, that provides the best least squares fit, i.e, it solves:

$$\min_{\boldsymbol{w}\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n}\Big(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i - b\Big)^2 \text{ such that } \| \boldsymbol{w} \|_0 \leq k\,, \tag{1.11}$$

where the ℓ_0 (pseudo) norm counts the number of non-zero elements of a vector. It is defined as $\| w \|_0 = \sum_{j=0}^{p} 1\,(w_j \neq 0)$, with $1(.)$ an indicator function. This problem is a non-convex combinatorial problem which has been shown to be NP-hard [Natarajan, 1995]. It is therefore intractable in high-dimensions, although approximate solutions can be obtained with greedy algorithms such as Matching Pursuit [Mallat and Zhang, 1993]. The LASSO is a convex relaxation of (1.11), where the ℓ_0 norm is replaced by the ℓ_1 norm. It solves:

$$\min_{\boldsymbol{w}\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n}\Big(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i - b\Big)^2 \text{ such that } \| \boldsymbol{w} \|_1 \leq k\,, \tag{1.12}$$

This problem is equivalent to the penalised form (1.3) to which most solvers apply. While the LASSO is a convex problem, the non-differentiability of the ℓ_1 norm prevents the use of many classical algorithms. In the original paper which introduced the LASSO [Tibshirani, 1996], the minimisation problem was cast as a Quadratic Program (QP) and an off-the-shelf QP solver was used. This option however does not leverage the sparse structure of the solution, and proved to be slow. Since then, a number of dedicated algorithms have been developed and have largely accelerated computations. We provide below a brief overview of these methods, without claiming to be exhaustive.

1.6.2 Available algorithms

We distinguish three main classes of algorithms that are broadly used to solve LASSO problems, namely path following algorithms, proximal methods and coordinate descent algorithms. They rely on three different strategies to exploit the known sparsity of the solution on the one hand, and to overcome the non-differentiability of the penalty on the other hand.

Path following algorithms, of which the *homotopy* algorithm [Osborne et al., 2000] and the Least Angle Regression (LARS) [Efron et al., 2004] are two well known examples, provide LASSO solutions for all values of the regularisation parameter λ. In other words, they compute the entire *regularisation path* $\lambda \mapsto \hat{\boldsymbol{w}}^{LASSO}(\lambda)$. These methods exploit the piecewise linearity of the regularisation path to compute it efficiently, starting from a known solution and consequently following the path. The breakpoints of the path correspond to values of λ at which a variable leaves or enters the active set, and are of course not known in advance. In between two breakpoints, the slope of the regularisation path has a closed form expression that depends on the features included in the active set. It can be deduced from this slope what the next breakpoint will be, and which variables enters or leaves the active set at that point. From there the active set can be updated, and subsequently the slope, and the process can be repeated until the full path is characterised. These algorithms typically start with $\lambda_{max} = \| \boldsymbol{X}^\top \boldsymbol{y} \|_\infty$ for which the solution of the LASSO is zero, i.e, $\boldsymbol{w}^{\hat{LASSO}}(\lambda_{max}) = \boldsymbol{0}_p$. It has been shown quite recently that

these algorithms have a worst case complexity that is exponential in the number of variables [Mairal and Bin, 2012], although in most practical cases the complexity turns out to be linear in the number of variables.

Proximal algorithms [Parikh and Boyd, 2014] take a very different approach. They are generally useful in cases where one term in the objective function is non-differentiable. As their denomination indicates, they are characterised by the use of proximal operators, which can be interpreted as gradient steps with regards to a smoothed surrogate of the objective function. Evaluating the proximal operator involves solving a convex optimisation problem, and the capacity to solve it efficiently is critical for the efficiency of the proximal algorithm itself. Fortunately for the LASSO, the proximal operator of the ℓ_1 penalty have a closed form solution which can be evaluated efficiently coordinate by coordinate. It is known as the *soft thresholding operator*:

$$\left(\mathrm{Prox}_{\lambda\|\,.\,\|_1}(\boldsymbol{v})\right)_i = \begin{cases} v_i - \lambda & v_i \geq \lambda\,, \\ 0 & |\,v_i\,| \leq \lambda\,, \\ v_i + \lambda & v_i \leq \lambda\,. \end{cases} \tag{1.13}$$

ISTA [Daubechies et al., 2004, Iterative Soft Thresholding Algorithm] and it accelerated counterpart FISTA [Beck and Teboulle, 2009] are two widely used proximal gradient methods for the LASSO. Under mild conditions, they converge in $O(\frac{1}{k})$ and $O(\frac{1}{k^2})$ respectively, where k is the number of iterations.

Coordinate descent (CD) algorithms [Friedman et al., 2007] are a third efficient option to solve the LASSO. They minimise the objective function one coordinate (or a block of coordinates) at a time while the others are kept fixed. For the LASSO, this univariate optimisation problem admits a closed form solution involving the soft thresholding operator. The order according to which coordinates are updated can follow various schemes. For example, Friedman et al. [2007] cycle trough all coordinates until convergence, Osher and Li [2009] propose a greedy CD where the updated coordinate is the one that leads to the largest decrease of the objective function, while Nesterov [2012] randomly choose the next coordinate. Currently, the very popular libraries GLMNET [Friedman et al., 2010] in R and scikit-learn [Pedregosa et al., 2012] in Python rely on coordinate descent to solve the LASSO.

1.6.3 Active set algorithms and Screening techniques

The algorithms described above are at the core of modern solvers. To obtain ever faster solvers, the holy grail would be to identify as soon as possible in the optimisation process the optimal support, in order to avoid wasteful updates. Active set and screening strategies are two approaches tending towards this goal. They are often used in conjunction with the algorithms described above (although not with path following methods since these can already be seen as active set techniques) to obtain further speed-ups.

Active set strategies prioritise computational resources on small sets of features which are likely to be included in the optimal support. They iteratively solve a sequence of subproblems restricted to the active set. Different strategies can be defined to update the active set at each

iteration. A typical choice consists in adding the most violated features with regards to the optimality conditions. The solver used for the small subproblems is usually chosen among the algorithms presented in the previous section. While there is only little theory supporting active set strategies, they enjoy great success in practice. In particular, the popular GLMNET package relies on an active set strategy combined with cyclic coordinate descent. It is also interesting to note that very recent LASSO solvers with state-of-the-art performance, such as BLITZ [Johnson and Guestrin, 2015] or CELER [Massias et al., 2018], also implement active set strategies.

Compared to active set frameworks, screening techniques adopt an opposite perspective, starting with the entire set of features and gradually discarding irrelevant ones. In the sequel we distinguish *safe rules*, which eliminate features which are *guaranteed* to be inactive at the optimum, from the rules which may mistakenly discard features that would be active at the optimum. The latter category typically includes methods that select the features with highest correlation with the response or residual, such as Sure Independence screening (SIS) [Fan and Lv, 2008] or the Strong Rules. SIS enjoys the sure screening property while the strength and interest of the Strong Rules lies in the fact that they are very rarely violated in practice. The safe rules were first introduced by El Ghaoui et al. [2012] for l1- regularised problems. Technically, these rules rely on the identification of a *safe region* which is guaranteed to contain the dual optimal solution, and which is subsequently used in conjunction with the Karush-Kuhn-Tucker conditions to safely eliminate features. Safe screening rules mostly differ according to the definition of the safe region used, which is often a dome or a sphere for computational tractability. Screening rules can be used as a preprocessing step, or can be reevaluated several times as the solver proceeds, in which case they are called *dynamic screening rules*. Screening techniques have been shown to lead to significant acceleration of available solvers, and are also useful in combination with active set strategies. In particular, the GLMNET library relies on the Strong Rules [Tibshirani et al., 2012] to temporarily restrict the pool of features from which the active set is chosen.

1.7 Contributions

This thesis tackles the curse of dimensionality in genomics on several fronts. New feature spaces, or equivalently new representations, are proposed to enhance supervised prediction tasks from either mutation data (NetNorM) or expression data (Suquan). The new representations proposed result from both feature engineering processes and representation learning models. A regularisation perspective is also adopted by scaling up existing regularisers, and in particular the ℓ_1 norm, to feature spaces that include interactions.

1.7.1 NetNorM

My first contribution addresses the challenge of learning from somatic cancer mutation profiles and in particular, the challenge of designing a representation of mutations that is amenable to statistical learning. In what follows, I first briefly review why there is a pressing need to design features from low level mutation data, and then summarise the feature engineering process that led to NetNorM.

Somatic mutations are mutations that appeared in one's lifetime as opposed to germline mutations which are passed on from parents to children. DNA sequencing has revealed that in most cancer types, tumours exhibit many somatic mutations. This opened the way to many

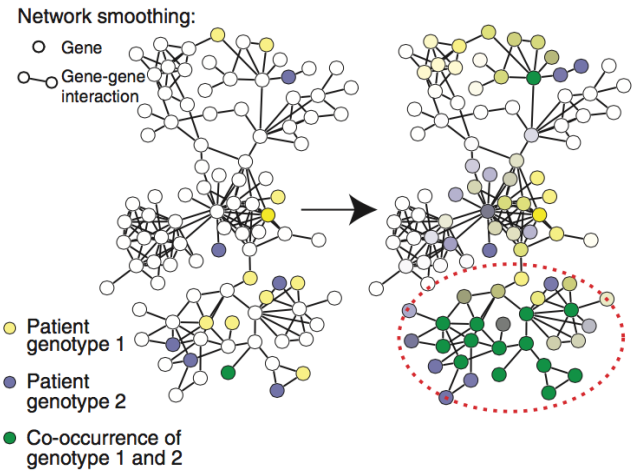


Figure 1.5 – Diffusion of mutations on a gene network. The figure was taken from [Hofree et al., 2013]. Two mutation profiles, a blue and a yellow one, are projected over a gene network. The mutation profiles appear in their original binary form on the left and in their smoothed version on the right. The creation of a similarity between them is well emphasised by the green region of the network where genes in both profiles have relatively high mutational loads inherited from close mutated neighbours in the network.

exciting research questions such as: what are the mutational processes behind the appearance of these mutations? which mutations drive the apparition and the development of cancer? can we design more effective personalised therapies based on mutational profiles? The hurdles that separate us from satisfying answers to these questions are numerous, and one of them relate to the nature of mutation data for which it is difficult to find a representation amenable to statistical learning. In our project we focused on somatic mutations obtained from whole exome sequencing, i.e, sequencing experiments only focused on parts of the genome that code for genes. Such data is naturally summarised as a patient by genes mutation matrix where a one stands for the presence of a mutation in a given gene. Typical datasets include a few hundreds tumour samples, in the best cases a few thousands, for around 20, 000 genes. Learning algorithms on such data will therefore be subject to the curse of dimensionality, but not only. In most cancers, the number of mutations found in a tumour ranges from 10 to 1000, among which a vast majority occur in genes that are only rarely mutated across patients. As a result, two mutation profiles are on average highly dissimilar, even within one cancer type. Said differently, two mutation profiles typically have only a handful of mutations in common, and sometimes even none. This property of mutation data, on top of its high-dimensional nature, seriously questions our ability to answer some of the fundamental questions mentioned above.

In order to circumvent these difficulties, a series of works has proposed to leverage known pathways and gene networks. The rationale behind these propositions is to summarise genetic aberrations at a higher level of organisation in the cell so as to reveal similarities between tumour mutation profiles that are not visible at the level of genes. Among these works, a method known as Network-Based Stratification [Hofree et al., 2013] showed particularly promising results. The authors provide a method to stratify cohorts of cancer patients based on their mutation profiles

and show that the obtained subgroups have significantly different chances of survival. This work gave the initial impulse to the NetNorM project, as we set out to thoroughly understand the nature of the underlying factors possibly captured by NBS. In order to illustrate the thought process from which NetNorM arose, we start by briefly describing the representation of mutation data used in NBS. This representation is obtained in two steps. First, the binary patients by genes mutation matrix is transformed via a diffusion process of mutations on the gene network (Fig. 1.5). Then, the resulting smoothed mutation matrix is quantile normalised. Given a mutation matrix $\boldsymbol{X}_0 \in \{0,1\}^{n\times p}$, the adjacency matrix of the gene network $\boldsymbol{A} \in \{0,1\}^{p\times p}$, the corresponding diagonal matrix containing node degrees $\boldsymbol{D} \in \mathbb{R}^{p\times p}$ and a parameter $\alpha \in \mathbb{R}$, the smoothed mutation matrix is obtained by running until convergence the following iteration process:

$$\boldsymbol{X} = \alpha \boldsymbol{X} \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}} + (1-\alpha)\, \boldsymbol{X}_0$$

Each iteration of this process spreads information from one gene to its neighbours while retaining part of the original information thanks to the introduction of the parameter α. The resulting smoothed mutation matrix is dense and its entries describe the proximity of each gene to mutated genes in the network. We will use the term of *mutational load* to refer to this notion of proximity. The smoothed mutation matrix is then *quantile normalised*, i.e, the rows, corresponding to mutation profiles, are normalised so that they all follow the same distribution of values while the ordering of gene mutation values in a mutation profile is preserved. More precisely, the lowest entries in each mutation profile are set equal to the median of all smallest entries across patients, the second smallest entries are set equal to the median of all second smallest entries, and so on until the largest entries are taken into account. While the first step is biologically motivated, it is less clear what the subsequent quantile normalisation step brings. We found however that this second step fundamentally transforms the mutation matrix and that it is crucial for the good performance of the method.

Our contribution makes a step towards a better understanding of the underlying explanatory factors that make this representation interesting for patient stratification purposes. We investigate the diffusion process, question its effective radius of influence. We also thoroughly examine the effects of quantile normalisation and the transformations it induces at the level of feature vectors. Based on this better understanding, we propose a new representation of mutations from which we show that improved cancer survival predictions can be obtained, although the performances remain globally modest. In a way, our new representation of mutations can be seen as a streamlined, stripped version of the representation used for NBS which, on top of providing improved cancer prognosis, provides a ground for an easier identification of the important underlying factors in mutation data.

1.7.2 Suquan

In NBS, a quantile normalisation procedure is applied to a smoothed version of mutation data. As described above, quantile normalisation normalises the samples so that after normalisation they all follow the same distribution. This distribution, which we will also refer to as the *quantile function* or *target quantile* in the sequel, is chosen in NBS as the vector whose first entry (resp. i^{th} entry) is equal to the median of all lowest (resp. i^{th} lowest) values across samples. For simplicity, we will later refer to this choice of target quantile as the *median quantile function*. We launched the project Suquan based on the results obtained with NetNorM which underlined

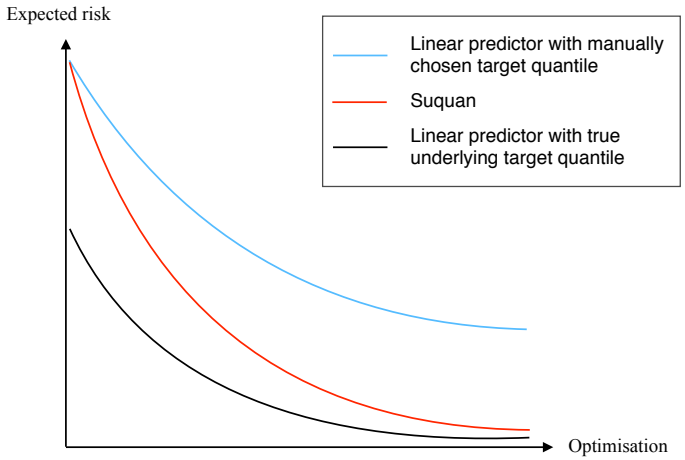


Figure 1.6 – Ideal behaviour of Suquan. The intended goal of Suquan is to identify the target quantile, or equivalently the feature space, for which the expected risk of the subsequent linear predictor is minimal.

the importance of the choice of target quantile for the subsequent prediction performances. Indeed, NetNorM can be seen as method which also implements a quantile normalisation step, but with a 'step' target quantile instead of a median target quantile.

We are not aware of applications of the quantile normalisation procedure to mutation data before the work of Hofree et al. [2013]. However this procedure has been used extensively in bioinformatics to preprocess many other data types including DNA microarrays for gene expression, genotyping or methylation measurements, RNA-sequencing data or ChIp-seq sequencing data. The measurements of these quantities are subject to much unwanted variations, such as changes in temperature or protocol implementation from one sample to another, and cannot be fed to a machine learning algorithm without proper normalisation. Quantile normalisation was precisely proposed to remove such unwanted variations in these types of data. Its extensive use in bioinformatics has been launched by Bolstad et al. [2003]. This work compares various normalisation procedures for gene expression microarrays and concludes by recommending the use quantile normalisation, where the target quantile is chosen as the median. Interestingly, the authors mention that while the quantile normalisation procedure they advocate for is based on the median quantile, 'there might be some advantages to using a common, non-data driven, distribution with the quantile method'. However in front of the arbitrary nature of such a choice they propose to establish the median quantile as a standard.

Suquan, which stands for 'Supervised Quantile Normalisation', is precisely aimed at providing a principled choice of the target quantile function. It jointly optimises the quantile function and the weights of a linear model for a given regression or classification task. Suquan can be seen as a representation learning technique in that it aims at finding a transformed input space, parametrised by the target quantile function, in which a linear predictor will perform best.

If we believe that the true data is such that all samples follow the same distribution, which has subsequently been corrupted by noisy and biased measurements, Suquan is designed to recover this distribution and consequently make better predictions from the data (Fig. 1.6). In other words, Suquan is expected to reduce the approximation error of the subsequent linear

predictor. Since, as usual in genomics, the number of available samples is quite limited, we take due care of controlling the possible rise in the estimation error by applying constraints on, or equivalently regularising, the learned target quantile. We applied Suquan to gene expression microarrays and showed that such a model, in spite of the limited number of samples available in our experiments, is able to learn a representation of the data on which the subsequent linear predictor yield overall better performances than with a manually chosen target quantile function.

1.7.3 WHInter

The LASSO is an interesting model to compute Polygenic risk scores (PRS) since it enables estimation and feature selection simultaneously. Feature selection, on top of playing a crucial statistical role for the generalisation ability of the model, allows to obtain more interpretable PRS than other models such as ridge regression. This notion of interpretability is critical if PRS are to be accepted one day in clinical setting, although one should be cautious regarding the interpretation of the features selected with the LASSO and take into account linkage disequilibrium and imputation issues. Compared to traditional PRS which a priori select features based on a predefined significance threshold, the LASSO allows to select features in a principled fashion. WHInter was designed with PRS applications in mind, and aims at solving the LASSO when dealing with two-way interaction features, as is the case when one seeks epistatic interactions between genes. This echoes to the search for missing heritability, which may be due to epistasis. If such is the case, a polygenic risk score that takes into account epistasis is expected to reach better performances than those which don't. The main contribution in WHInter tackles the computational challenges that arise when dealing with two-way interaction features for problem sizes that include today's SNP datasets, i.e, several hundred thousands of original features, meaning billions to trillions of interaction features. While much progress has been made regarding LASSO solvers, they still do not apply to problems of such size, where the design matrix does not even fit in memory. WHInter is able to provide an exact solution to l1 regularised linear models, when interactions are taken into account and without any heredity assumption on the interactions, in a reasonable amount of time for problem sizes that include today's SNP datasets, i.e, several hundred thousands of original features. It relies on an active set strategy, with at its core contributions which allow to delineate the active set effectively when considering interaction features.

1.7.4 Published work appearing in this thesis

The contributions in this thesis are available as published articles or preprints.

- M. Le Morvan, A. Zinovyev and J. P. Vert. NetNorM: Capturing cancer-relevant information in somatic exome mutation data with gene networks for cancer stratification and prognosis. *PLoS Comput. Biol.*, 13(6):e1005573, 2017
- M. Le Morvan and J. P. Vert. Supervised Quantile Normalisation. *ArXiv e-prints*, 2017
- M. Le Morvan and J. P. Vert. WHInter: A Working set algorithm for High-dimensional sparse second order Interaction models. *ArXiv e-prints*, 2018 This preprint is submitted to ICML 2018.

The following chapters closely follow these references.

Chapter 2

NetNorM: Capturing Cancer-relevant Information in Somatic Exome Mutation Data with Gene Networks for Cancer Stratification and Prognosis

Contents

2.1 Introduction . . . 33
2.2 Overview of NetNorM . . . 34
2.3 Survival prediction . . . 36
2.3.1 NetNorM provides state-of-the-art prognosis for patient survival based on mutation profiles . . . 36
2.3.2 The biological information encoded in the gene network contributes to the prognosis . . . 38
2.3.3 Analysis of predictive genes . . . 39
2.3.4 NetNorM enhances clinical data based prognosis . . . 44
2.4 Patient stratification . . . 46
2.4.1 NetNorM allows stable unsupervised stratification of patients with significantly different survival curves . . . 46
2.4.2 Patient stratification with randomised networks . . . 48
2.4.3 Patient subtypes obtained with NetNorM are characterised by distinct pathways . . . 48
2.5 Discussion . . . 51
2.6 Materials and Methods . . . 53

Abstract

Genome-wide somatic mutation profiles of tumours can now be assessed efficiently and promise to move precision medicine forward. Statistical analysis of mutation profiles is however challenging due to the low frequency of most mutations, the varying mutation rates across tumours, and the presence of a majority of passenger events that hide the contribution of driver events. Here we propose a method, NetNorM, to represent whole-exome somatic mutation data in a form that enhances cancer-relevant information using a gene network as background knowledge. We evaluate its relevance for two tasks: survival prediction and unsupervised patient stratification. Using data from 8 cancer types from The Cancer Genome Atlas (TCGA), we show that it improves over the raw binary mutation data and network diffusion for these two tasks. In doing so, we also provide a thorough assessment of somatic mutations prognostic power which has been overlooked by previous studies because of the sparse and binary nature of mutations.

Résumé

Les profils de mutations somatiques pangénomiques des tumeurs cancéreuses peuvent aujourd'hui s'obtenir de façon efficace, et promettent de faire progresser la médecine de précision. L'analyse statistique de ces profils de mutations est cependant complexe en raison de la faible fréquence de la plupart des mutations, de la disparité des taux de mutations selon les tumeurs, et de la présence d'une majorité de mutations passagères qui cachent la contribution de celles qui sont causales. Nous proposons ici une méthode, NetNorM, pour représenter les données de mutations somatiques pangénomiques sous une forme qui mette en avant les informations pertinentes liées au cancer, en utilisant un réseau de gène comme connaissance de fond. Nous évaluons la pertinence de NetNorM pour deux tâches : la prédiction de survie et la stratification de patients non supervisée. En utilisant les données de 8 types de cancers provenant du TCGA (The Cancer Genome Atlas), nous montrons que la méthode proposée améliore les performances obtenues pour ces deux tâches en comparaison des données de mutation brutes binarisées mais également des processus de diffusion sur les réseaux de gènes. Ce faisant, nous fournissons également une évaluation approfondie du potentiel pronostique des mutations somatiques, évaluation qui a été négligée par les études précédentes du fait de la nature parcimonieuse et binaire des mutations.

2.1 Introduction

Tumourigenesis and cancer growth involve somatic mutations which appear and accumulate during cancer progression. These mutations impair the normal behaviour of various cancer genes, and give cancer cells an often devastating advantage to proliferate over normal cells [Hanahan and Weinberg, 2011; Stratton et al., 2009; Vogelstein et al., 2013]. Systematically assessing and monitoring somatic mutations in cancer therefore offers the opportunity not only to better understand the biological processes involved in the disease, but also to help rationalise patient treatment in a clinical setting. Rationalising treatment involves finely characterising the genomic abnormalities of each given patient to discover which may be treatable by a targeted therapeutic agent, as well as improving prognosis using molecular information [Chin and Gray, 2008; Mardis, 2012; Olivier and Taniere, 2011]. The development of fast and cost-effective technologies for high-throughput sequencing in the last decade has triggered the launch of numerous data collection projects such as The Cancer Genome Atlas (TCGA) [The Cancer Genome Atlas Research Network et al., 2013] or the International Cancer Genome Consortium (ICGC) [Hudson et al., 2010], aiming at characterising at the molecular level, including genome-wide or exome-wide somatic mutations, thousands of cancer samples of multiple origins. By systematically comparing the molecular portraits of the resulting cohorts, one might expect to be able to detect frequently mutated genes or groups of genes, and find associations between particular mutations and cancer phenotypes, response to treatment, or survival [Kandoth et al., 2013; The Cancer Genome Atlas Network, 2012; The Cancer Genome Atlas Research Network, 2008, 2011].

The analysis of somatic mutation profiles is however challenging for multiple reasons. First, most somatic mutations detected by systematic sequencing are likely to be irrelevant for biological or clinical applications. This is due to the fact that only a few driver mutations are required to confer a growth advantage to the cancer cell, and therefore most somatic mutations are likely to be passenger mutations which do not contribute to the cancer phenotype [Greenman et al., 2007] [see Vogelstein et al., 2013, for a review]. Second, sequencing efforts have shown that while a few genes are frequently mutated, the vast majority of genes are mutated in only a handful of patients [Lawrence et al., 2014; Wood et al., 2007]. As a result, the mutation profiles of two tumours often only share a few if any genes in common. Third, even if originating from the same tissue, tumours may exhibit widely varying mutation rates. The overall mutational burden of a tumour constitute a strong and informative signal [Birkbak et al., 2013; Lawrence et al., 2013; Rizvi et al., 2015] but can however complicate the retrieval of more subtle signals. Combined with the inherent high dimensionality of somatic mutation datasets, this makes any statistical analysis of cohorts of whole-exome somatic mutation profiles extremely challenging.

In order to make somatic mutation profiles more amenable to statistical analysis, several studies have used gene networks as prior knowledge [Barillot et al., 2012; Creixell et al., 2015]. Considering genes in the context of networks instead of analysing them independently allows sharing mutation information among neighbouring genes and identifying disruptions at the level of pathways or protein complexes instead of single genes. A popular method to leverage this prior knowledge consists in using a diffusion process on the gene network. This technique first appeared for the analysis of gene expression and GWAS data [Köhler et al., 2008; Kuperstein et al., 2015; Qian et al., 2014; Rapaport et al., 2007; Vanunu et al., 2010], and has more recently been used for mutation profiles [Babaei et al., 2013; Hofree et al., 2013; Hou and Ma, 2014; Jia and Zhao, 2014; Leiserson et al., 2014; Vandin et al., 2011]. Network diffusion processes allow

smoothing binary vectors of somatic gene mutations into non-negative real-valued vectors of mutational statuses, where the mutational status of a gene increases when it is close to mutated genes in the network. This approach led to state-of-the-art methods for the discovery of driver pathways or complexes [Leiserson et al., 2014] and for the stratification of patients into clinically relevant subtypes [Hofree et al., 2013] using whole-exome mutation profiles.

In this work we propose NetNorM, a new method to enhance mutation data with gene networks. NetNorM transforms a patient's binary mutation profile by either removing mutations or creating "proxy" mutations based on the gene network topology, until all patients reach a consensus number of mutations. The resulting mutation matrix is binary like the initial one, nonetheless we establish that it encodes new information reflecting both local network neighbourhood mutational burdens and the overall tumour mutational burden.

We evaluate the relevance of NetNorM on two tasks: survival prediction and patient stratification from exome somatic mutation profiles. In doing so, we also provide a thorough assessment of somatic mutations prognostic power which has been overlooked by previous studies because of the sparse and binary nature of mutations [Yuan et al., 2014]. We show that NetNorM produces state-of-the-art results for these two tasks compared to the raw binary mutation data and to network diffusion-based methods. By comparing results obtained with real versus randomised networks, we further show that the increase in relevance is actually partly driven by the gene's network prior knowledge. However, we observe that considering interactions between mutated genes and their network neighbours only is enough do achieve state-of-the-art results, thereby shedding light on which are the network features that are the most informative.

2.2 Overview of NetNorM

NetNorM takes as input an undirected gene network and raw exome somatic mutation profiles and outputs a new representation of mutation profiles which allows better survival prediction and patient stratification from mutations (Fig. 2.1). Here and in what follows, the "raw" mutation profiles refer to the binary patients times genes matrix where 1s indicate non-silent somatic point mutations or indels in a patient-gene pair and 0s indicate the absence of such mutations. The new representation of mutation profiles computed with NetNorM also takes the form of a binary patients times genes mutation matrix, yet with new properties. While different tumours usually harbour different number of mutations, with NetNorM all patient mutation profiles are normalised to the same number k of genes marked as mutated. The final number of mutations k is the only parameter of NetNorM, which can be adjusted by various heuristics, such as the median number of mutations in the original profiles, or optimised by cross-validation for a given task such as survival prediction. In order to represent each tumour by k mutations, NetNorM adds "missing" mutations to samples with less than k mutations, and removes "non-essential" mutations from samples with more than k mutations. The "missing" mutations added to a sample with few mutations are the non-mutated genes with the largest number of mutated neighbours in the gene network, while the "non-essential" mutations removed from samples with many mutations are the ones with the smallest degree in the gene network. These choices rely on the simple ideas that, on the one hand, genes with a lot of interacting neighbours mutated might be unable to fulfil their functions and, on the other hand, mutations in genes with a small number of interacting neighbours might have a minor impact compared to mutations in more connected genes.

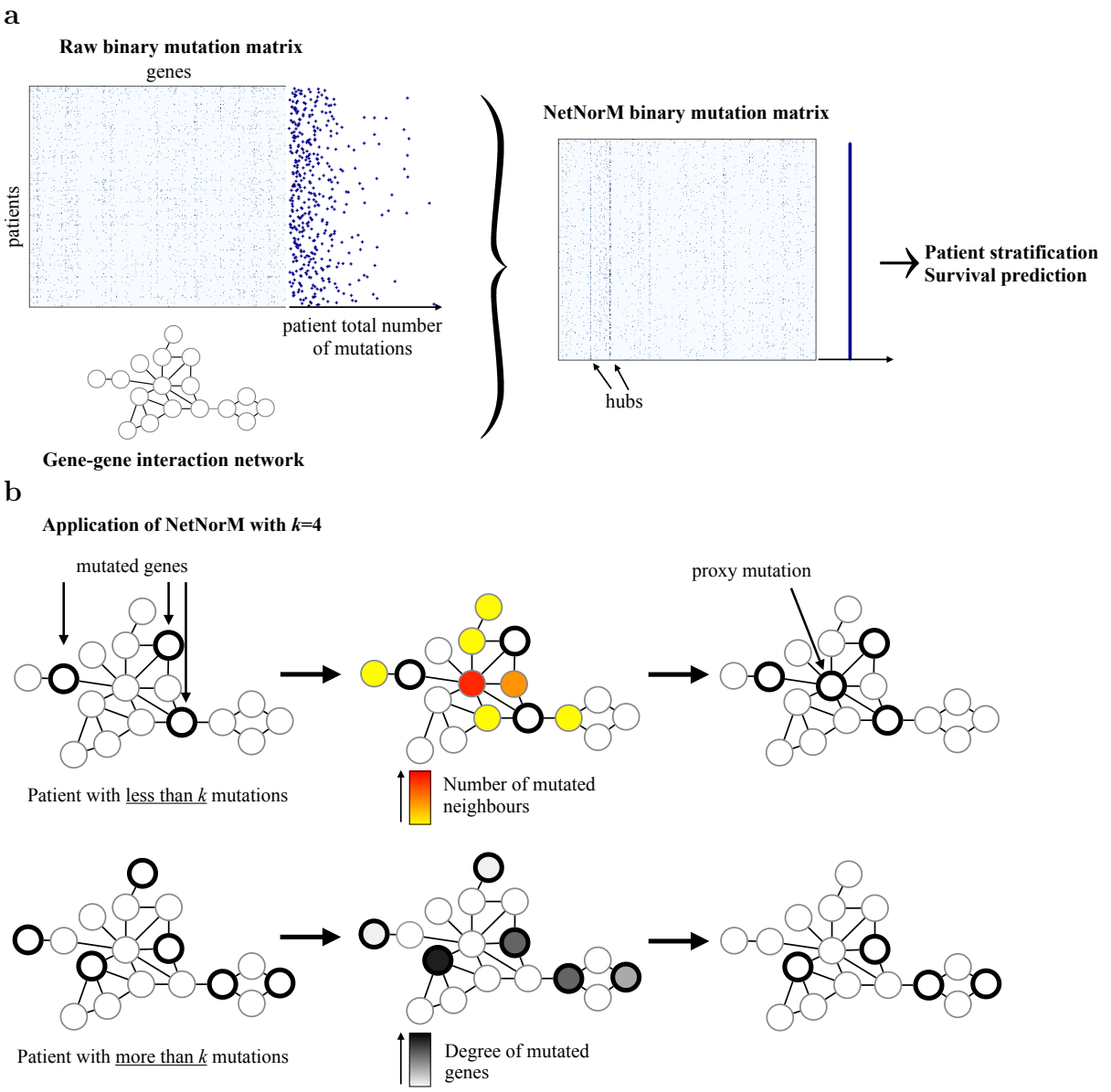


Figure 2.1 – Overview of NetNorM. (a) Using a gene network as background knowledge (lower left), NetNorM normalises each mutation profile in a collection of somatic mutation profiles (upper left) into a new, binary representation (right) which encodes additional information relative to tumours' overall mutational burden and hubs' neighbourhood mutational burden. This new representation allows performing patient stratification with unsupervised clustering techniques, or survival analysis. (b) NetNorM normalises every patient mutation profile to k mutations. Patients with less than k mutations get 'proxy' mutations in their genes with the highest number of mutated neighbours until they reach k mutations. Patients with more than k mutations have mutations 'removed' in their genes with lowest degree until they reach k mutations.

In this study, we compare NetNorM-processed profiles with the raw mutation data and with profiles processed with network smoothing (NS) [Zhou et al., 2004] (also called network diffusion, or network propagation) followed by quantile normalisation (QN) as implemented in [Hofree et al., 2013]. We refer to this method as NSQN below. Mutation profiles, either raw or processed with NetNorM or NSQN, are restricted to the genes present in the network used. While both NetNorM and NSQN leverage gene network prior knowledge to enhance mutation data, the two methods have fundamental differences. First, NetNorM leverages information about first neighbours in the network only while NSQN spreads mutation information at a more global scale on the gene network. Second, with NetNorM the normalised profiles all have the same value distribution by construction, since they are all binary vectors with k ones, removing the need for further quantile normalisation which, as we discuss below, is critical for NSQN.

2.3 Survival prediction

2.3.1 NetNorM provides state-of-the-art prognosis for patient survival based on mutation profiles

To assess the relevance of NetNorM, we first explore the capacity of somatic mutations to predict patient survival. We collected a total of 3,278 full-exome mutation profiles of 8 cancer types from the TCGA portal (Table 2.1), censored survival information and clinical data. In parallel we retrieved a gene network to be used as background information for NSQN and NetNorM : Pathway Commons, which integrates a number of pathway and molecular interaction databases [Cerami et al., 2010]. For each cancer type, we use these data to assess how well survival can be predicted from somatic mutations. For that purpose, we perform survival prediction with a sparse survival SVM (see Methods) using either the raw mutation profiles or the profiles processed with NSQN or NetNorM, respectively, and assess their performance by cross-validation using the concordance index (CI) on the test sets as performance metric.

Figure 2.2 summarises the survival prediction performances for the 8 cancer types, when the sparse survival SVM is fed with the raw mutation profile, or with the mutation profiles modified by NSQN or NetNorM using Pathway Common as gene network. For two cancers (LUSC, HNSC), none of the methods manages to outperform a random prediction, questioning the relevance of the mutation information in this context. For OV, BRCA, KIRC and GBM, all three methods are significantly better than random, although the estimated CI remains below 0.56, and we again observe no significant difference between the raw data and the data transformed by NSQN or NetNorM. Finally, the last two cases, SKCM and LUAD, are the only ones for which we reach a median CI above 0.6. In both cases, processing the mutation data with NetNorM significantly improves performances compared to using the raw data or profiles processed with NSQN. More precisely, for LUAD the median CI increases from 0.56 for the raw data and 0.53 for NSQN to 0.62 for NetNorM. In the case of SKCM, the median CI increases from 0.48 for the raw data to 0.52 for NSQN, and to 0.61 for NetNorM. For SKCM, both NetNorM and NSQN are significantly better than the raw data ($P < 0.01$).

In our experiments, silent mutations are systematically filtered out. To evaluate whether this preprocessing step is actually detrimental or beneficial for the survival prediction task, we performed further experiments where silent mutations are not filtered out (Fig. A.1). We find that considering silent mutations does not improve survival prediction performances compared

Cancer type	Patients	Genes	Deaths	Download date
LUAD (Lung adenocarcinoma)	430	20 596	110	6/22/2015
SKCM (Skin cutaneous melanoma)	307	17 461	129	11/18/2015
GBM (Glioblastoma multiform)	265	14 748	195	11/18/2015
BRCA (Breast invasive carcinoma)	945	16 806	97	11/25/2015
KIRC (Kidney renal clear cell carcinoma)	411	10 608	136	11/25/2015
HNSC (Head & Neck squam. cell carcinoma)	388	17 022	140	11/25/2015
LUSC (Lung squamous cell carcinoma)	169	13 589	70	11/25/2015
OV (Ovarian serous cystadenocarcinoma)	363	10 192	172	11/24/2014

Table 2.1 – Summary of the full exome mutation profiles used in this study. We analysed a total of 3,278 samples from 8 cancer types, downloaded from the TCGA portal.

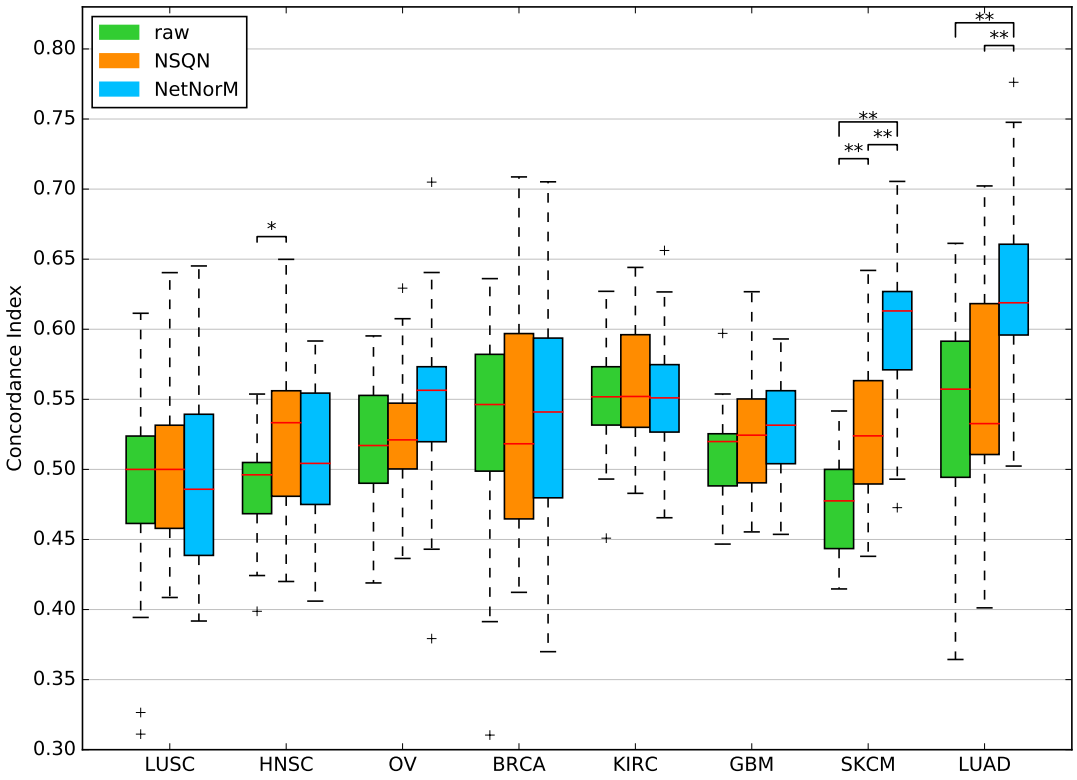


Figure 2.2 – Comparison of the survival predictive power of the raw mutation data, NSQN and NetNorM (with Pathway Commons as gene network) for 8 cancer types. For each cancer type, samples were split 20 times in training and test sets (4 times 5-fold cross-validation). Each time a sparse survival SVM was trained on the training set and the test set was used for performance evaluation. The presence of asterisks indicate when the test CI is significantly different between 2 conditions (Wilcoxon signed-rank test, $P < 5 \times 10^{-2}$ (*) or $P < 1 \times 10^{-2}$ (**)).

to the case where they are filtered out. In fact, the performance of NetNorM on LUAD is significantly decreased when silent mutations are taken into account.

To assess the influence of the gene network used on the survival prediction performances, we also repeated our experiments with four gene networks instead of Pathway Commons: BioGRID [Chatr-aryamontri et al., 2016] , HPRD [Prasad et al., 2009], HumanNet [Lee et al., 2011] and STRING [Szklarczyk et al., 2015] (Fig. A.2). For HumanNet and STRING, only the 10% most confident interactions were retained. We observe that no gene network clearly stands out as the best network for all cancers. For two cancers, LUSC and HNSC, performances remain very low, close to a concordance index of 0.5, whatever the method or network used. For three cancers, OV, BRCA and KIRC, NetNorM is the only method to significantly outperform the raw data with at least one network (HumanNet and STRING for OV, HPRD for BRCA, and STRING for KIRC) with a median concordance index above 0.55. For GBM, NSQN is the only method to outperform the raw data (with HumanNet and STRING) with a median concordance index above 0.55. For the two remaining cancers, LUAD and SKCM, the best performances are those obtained with NetNorM using Pathway Commons, with median CI of 0.62 and 0.61 respectively. Across all cancers, methods, and networks combinations, these two cases are the only ones where the median CI obtained exceeds 0.60.

Finally, as mutations in some genes are known to be associated with survival, such as *TP53* in BRCA and HNSC which is associated with worsened survival [Robles and Harris, 2010], we evaluate the prediction ability of individual genes' mutation status. For each cross-validation fold, the gene giving the best concordance index on the training set is selected and its performance evaluated on the test set. We find that for 5 cancers, the performances of individual genes are similar to those of the survival SMV applied to the whole raw mutations datasets (Fig. A.3). However for BRCA and HNSC, better survival predictions are obtained using a single gene than the whole raw mutational profiles. Yet these predictions are not better than those obtained with NetNorM. For these two cases, *TP53* is the gene selected in the majority of folds (17/20 for HNSC and 19/20 for BRCA), which is in accordance with existing literature (Table A.1). Lastly, the survival SVM applied to the whole dataset yields significantly better performances than the single gene approach for LUAD. This means that contrary to the BRCA and HNSC cases, the linear combinations of genes which are found for LUAD have a predictive power that generalises well to unseen data.

In summary, these results show that for at least 6 out of 8 cancers investigated, somatic mutation profiles have a prognostic value, and that for two of them (SKCM and LUAD) it is possible to improve the prognostic power of mutations by using gene networks and to reach a CI above 0.6. In both cases, NetNorM is significantly better than NSQN.

2.3.2 The biological information encoded in the gene network contributes to the prognosis

To test whether the biological information contained in the gene network plays a role in the improvement of survival predictions for LUAD and SKCM, we evaluate again NetNorM and NSQN using 10 different randomised versions of Pathway Commons for these two cancers. Random networks were obtained by shuffling the nodes' labels of the real network while keeping the structure unchanged. The results, shown on Fig. 2.3, demonstrate that NetNorM performs significantly better with a real network. More precisely, the real network significantly outper-

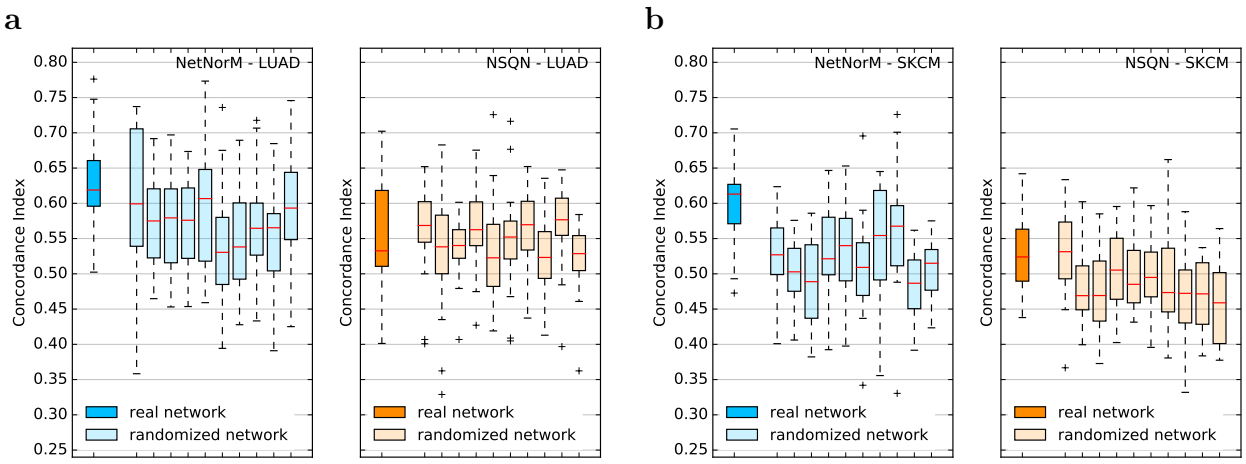


Figure 2.3 – Effect of network randomisation on survival prediction performances. (a-b) Performances obtained for 20 cross-validation folds with Pathway Commons (real network) and 10 randomised versions of Pathway Commons (randomised network) with NetNorM (left) and NSQN (right) for LUAD (a) and SKCM (b).

forms all random networks for SKCM and 8 out of 10 random networks for LUAD (Wilcoxon signed-rank test with correction for multiple hypothesis testing , FDR $\leq 5\%$). NSQN also performs significantly better with a real network for SKCM (7 out of 10 cases) but not for LUAD (0 out of 10 cases). This last observation is not surprising since NSQN does not improve over the raw data for LUAD, which suggests that the method may have failed to leverage network information in this case. In summary, these results indicate that the improvements obtained with NetNorM and NSQN compared to the raw data do rely on biological information encoded in the network.

2.3.3 Analysis of predictive genes

In order to shed light on the reasons why NetNorM outperforms the raw data and NSQN on survival prediction for SKCM and LUAD, we now analyse more finely the normalisation carried out by NetNorM on the mutation profiles, and why they lead to better prognostic models. For that purpose, we focus on the genes that are selected at least 50% of the times by the sparse survival SVM during the 20 different train/test splits of cross-validation, after NetNorM normalisation. This leads to 21 frequently selected genes for LUAD and 10 for SKCM (Fig. 2.4). Remembering that NetNorM either removes mutated genes for patients with many mutations, or adds proxy mutations for patients with few mutations, we can assess for each frequently selected gene whether it tends to exhibit proxy mutations or whether it tends to be actually mutated in the tumour. This is done by comparing how frequently it is marked as mutated on the raw data and after NetNorM normalisation (Fig. 2.4, top plot). For both cancers, we observe two clearly distinct groups of frequently selected genes: those that concentrate proxy mutations (which we will call *proxy genes*, in red in Fig. 2.4), and those to which NetNorM brings only few modifications compared to the raw data, meaning they are usually actually mutated in the tumours (in black in Fig. 2.4).

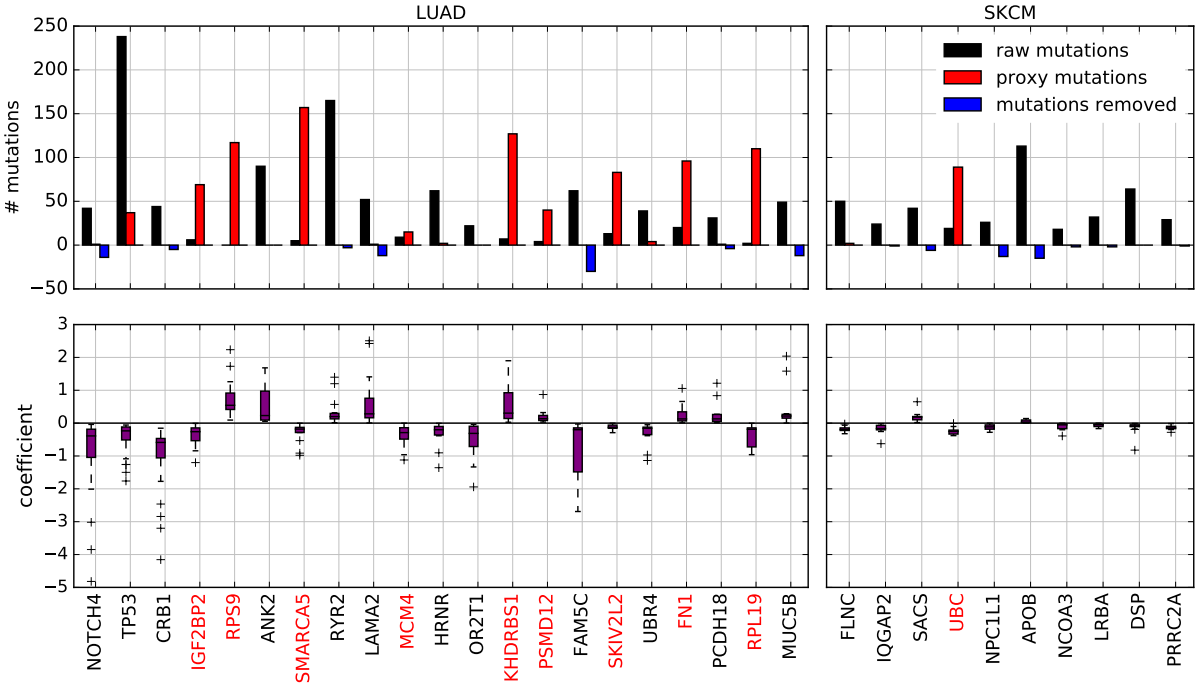


Figure 2.4 – Genes frequently selected in the survival prediction model for LUAD (left) and SKCM (right) learned using the NetNorM representation of mutations with Pathway Commons as gene network. The genes reported are those that were selected at least 10 times in 20 cross-validation folds. For each cancer, genes are ordered from the most frequently selected (left) to the least frequently selected (right). The top panel reports the number of raw mutations in the selected genes (black), as well as the number of "proxy" mutations (red) and the number of mutations removed (blue) after application of NetNorM. The bottom panel reports the coefficients of a gene in the survival SVM model across the cross-validation folds where this gene was selected. Gene names marked in red indicate proxy genes.

Genes with few modifications imputed by NetNorM

In the case of LUAD, 12 out of the 21 selected genes are non-proxy genes, meaning they tend to be really mutated when they are marked as mutated after NetNorM normalisation. Interestingly, mutations in 5 of these genes are predictive of an increased survival time (corresponding to a positive coefficient in the sparse survival SVM) while mutations in the remaining 7 genes are predictive of a decreased survival time (corresponding to a negative coefficient) (Fig. 2.4, bottom plot). The three most important predictors according to their frequency of selection include *NOTCH4*, *TP53* and *CRB1* (selected in all of the 20 folds) and are all predictive of a decreased survival time. *TP53* is a well-known cancer gene and has been reported as significantly mutated in LUAD [Collisson et al., 2014; Ding et al., 2008]. *NOTCH4* is part of the *NOTCH* signalling pathway which has been widely implicated in cancer and shown to act as both oncogene or tumour suppressor depending on the context [Ranganathan et al., 2011]. Finally, *CRB1* is known to localise at tight junctions but little is known about its role in carcinogenesis [Roh et al., 2002]. Among the remaining genes, *LAMA2* (selected in 16 out of 20 folds) has been detected as a driver gene in head and neck squamous cell carcinoma and *PCDH18* (selected in 11 out of 20 folds) has been detected as a driver in bladder carcinoma, cutaneous melanoma and in a pan-cancer analysis setting [Gonzalez-Perez et al., 2013]. In the case of SKCM, 9 out of the 10 selected genes are genes with few modifications. This includes 7 genes whose mutations are predictive of a decreased survival time (*FLNC*, *IQGAP2*, *NPC1L1*, *NCOA3*, *LRBA*, *DSP*, *PRRC2A*), and 2 whose mutations are predictive of an increased survival time (*SACS* and *APOB*). Among these genes, *NCOA3* (also known as *AIB1* or *SRC3*) is an important oncogene in breast cancer [Anzick, 1997; Lahusen et al., 2009]. Its role in other cancers is unclear however it has been shown that overexpression of *NCOA3* is a marker of melanoma outcome [Rangel et al., 2006]. *LRBA* interacts with multiple important signal transduction pathways including *EGFR* and its deregulation in several cancer types has been shown to facilitate cancer cell growth [Wang et al., 2004]. Moreover *LRBA* expression has been indicated as a clinical outcome predictor in breast cancer [Andres et al., 2013]. *Filamin C* (*FLNC*, selected in all of the 20 folds) is a large actin-cross-linking protein which has been shown to inhibit proliferation and metastasis in gastric and prostate cancer cell lines [Qiao et al., 2014]. *Desmoplakin* (*DSP*) is required for functional desmosomal adhesion which has been linked to cancer cells development and progression in several cancers [Chidgey and Dawson, 2007; Dusek and Attardi, 2011]. Moreover *IQGAP2* has been identified as a tumour suppressor gene in hepatocellular carcinoma, gastric and prostate cancers [Xie et al., 2015].

Proxy genes

In addition to somatically mutated genes, several proxy genes, mutated by the NetNorM procedure, are often selected by the survival model. The proxy genes for LUAD are *IGF2BP2*, *RPS9*, *SMARCA5*, *MCM4*, *KHDRBS1*, *PSMD12*, *SKIV2L2*, *FN1*, *RPL19* and for SKCM *UBC* is the only one. These genes are among the biggest hubs in the network. This is expected as proxy mutations are imputed in genes with a lot of mutated neighbours, which is more likely to occur for genes that simply have a lot of neighbours. The fact that these proxy genes were selected in the survival models means that they have some prognostic power. In particular for LUAD, the better prediction performances achieved by NetNorM compared to the raw data is largely explained by better predictions made for the half of patients with fewer mutations, and therefore

by the proxy mutations that were created in these patients (Fig. 2.5a).

The prognostic power of proxy genes in NetNorM comes from at least two types of information they capture. The first type of information captured by proxy mutations is the total number of mutations in a patient. Patients harbouring proxy mutations are significantly less mutated than those without proxy mutations (Welsh t-test, $P \leq 1 \times 10^{-2}$) in a given proxy gene. This results from the fact that patients with few mutations receive as many proxy mutations as needed to reach the target number of mutations k, and therefore proxy mutations have a higher probability to be set in patients with few mutations. The fact that NetNorM creates proxies for the total number of mutations raises the question of whether or not the total number of mutations can improve survival predictions made using the raw binary mutation profiles. To answer this question, we trained a model to predict survival from the raw binary mutation profiles concatenated with a feature, scaled to unit variance, which records the total number of mutations in each patient (Fig. A.4). According to our results, taking into account such a feature does not improve survival prediction performances compared to using the raw data alone. We therefore tested another feature which better mimics the proxies created by NetNorM, which we call 'proxies'. This feature is equal to the total number of mutations in a patient for patients with less than k mutations, and is equal to 0 otherwise. We trained a survival prediction model on the raw data concatenated with the feature 'proxies', scaled to unit variance, where k is chosen by cross-validation. Interestingly, we find that using such a feature allows to significantly improve the results obtained for OV, KIRC and LUAD compared to the raw data alone. In particular, the performances obtained for LUAD are on par with those obtained with NetNorM, suggesting that the feature 'proxies' summarises well the information leveraged by NetNorM. However this is not the case for SKCM since considering the feature 'proxies' does not improve over using the raw data alone. We draw two conclusions from these observations: first, NetNorM creates relevant proxies for the total number of mutations which, in combination with the binary mutation profiles, have predictive power; second, such proxies do not entirely explain the performances of NetNorM, at least for SKCM.

The second type of information captured by proxy mutations is genes' neighbourhood mutational burden (NMB). When we look at which patients get mutated in a given gene after NetNorM normalisation (red dots in Fig. 2.5b), we observe that they tend to have more mutations in the neighbours of this gene than what the sole mutational burden would predict (represented by the regression line in Fig. 2.5b). In other words, among the hubs that could get mutated by NetNorM for patients with few mutations, the ones that get mutated tend to be the ones surrounded by more mutations than expected given the mutational burden of the patient. NetNorM thus creates proxy mutations when a gene's NMB is higher than expected.

Among the proxy genes selected in LUAD (resp. SKCM), *IGF2BP2*, *SMARCA5*, *MCM4*, *PSMD12* and *SKIV2L2* (resp. *UBC*) define groups of patients with significantly different survival outcomes (log-rank test, $P \leq 5 \times 10^{-2}$) . Given the discussion in the previous paragraph, this may be due to differences in the overall mutational burden between tumours, to differences in NMB for some genes, or to both effects. To clarify the contributions of each effect, we investigate whether such distinct survival outcomes can be obtained with proxies for the total number of mutations only, regardless of NMBs. To this end, we simulate proxy mutations according to a probability depending on patients' total number of mutations only. By contrast, NetNorM mutates genes according to patients' total number of mutations and according to genes' NMB. Then for each gene we compare the survival outcomes of the obtained subgroups (patients which

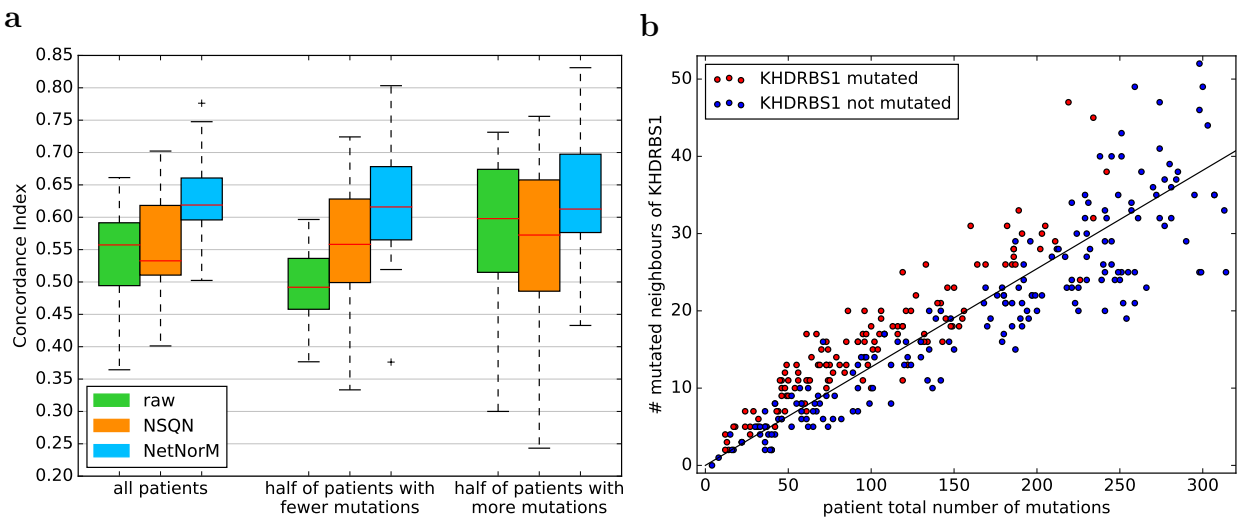


Figure 2.5 – Analysis of predictive genes. (a) Comparison of survival prediction performances according to patients' mutational burden for LUAD. Three different representations of the mutations are used to perform survival prediction using a ranking SVM: *raw* (the raw binary mutation data), *NSQN* (network smoothing with quantile normalisation) and *NetNorM*. Performances for half of the patients with fewer (resp. more) mutations are derived from the predictions made using the whole dataset. (b) Scatter plot of the total number of mutations in a patient of the LUAD cohort (x-axis) against the number of mutated neighbours of *KHDRBS1* in a patient (y-axis). Only patients with less than $k_{med} = 295$ mutations are shown, where k_{med} is the median value of k learned across cross-validation folds. Red (resp. blue) indicate patients mutated (resp. non mutated) in *KHDRBS1* after processing with NetNorM using $k = k_{med}$. The black line was fit by linear regression and by definition indicates the expected number of mutated neighbours of *KHDRBS1* given the mutational burden of a patient.

were imputed a proxy mutation versus those that were not) using a log-rank test and examine whether the log-rank statistic is higher with NetNorM than with the simulations (see Methods for more details). We find that all of *IGF2BP2*, *SMARCA5*, *MCM4*, *PSMD12*, *SKIV2L2* and *UBC* produce groups with a significantly higher log-rank statistic with NetNorM than with their simulated counterpart (log-rank test, $P \leq 5 \times 10^{-2}$). This clarifies that the prognostic information captured by proxy mutations with NetNorM combines the overall mutational burden of the patient with local mutational burden on the gene network.

2.3.4 NetNorM enhances clinical data based prognosis

We assess whether the combination of both mutations and clinical features can improve performances for LUAD and SKCM compared to using clinical data alone. For this purpose, two sparse survival SVM models are trained independently: one on the raw mutation data or mutations preprocessed with NSQN or NetNorM and one on the clinical data. Then the survival predictions from both models are simply averaged (after being standardised to unit variance). The resulting predictions are again evaluated in a 4 times 5 folds cross-validation setting. First, the results show that mutations preprocessed with NetNorM and the clinical data yield similar performances ($P = 0.52$, Wilcoxon signed rank test) for LUAD while the clinical data performs significantly better than NetNorM in the case of SKCM ($P \leq 1 \times 10^{-2}$) (Fig. 2.6). Moreover, we observe that combining mutations preprocessed with NetNorM with clinical features allows improving survival predictions compared to the clinical data alone for both LUAD ($P = 4.8 \times 10^{-2}$) and SKCM ($P = 5.7 \times 10^{-2}$). More precisely, the median CI increases from 0.64 with the clinical data to 0.66 with the combination of NetNorM and the clinical data for LUAD and from 0.66 to 0.70 in the case of SKCM. We also tried to concatenate the mutation profiles with the clinical data before training a unique model and observed that it did not improve the results compared to the previous strategy (Fig. A.5). Overall, these results suggest that mutations could provide useful prognostic information that is complementary to the clinical information available.

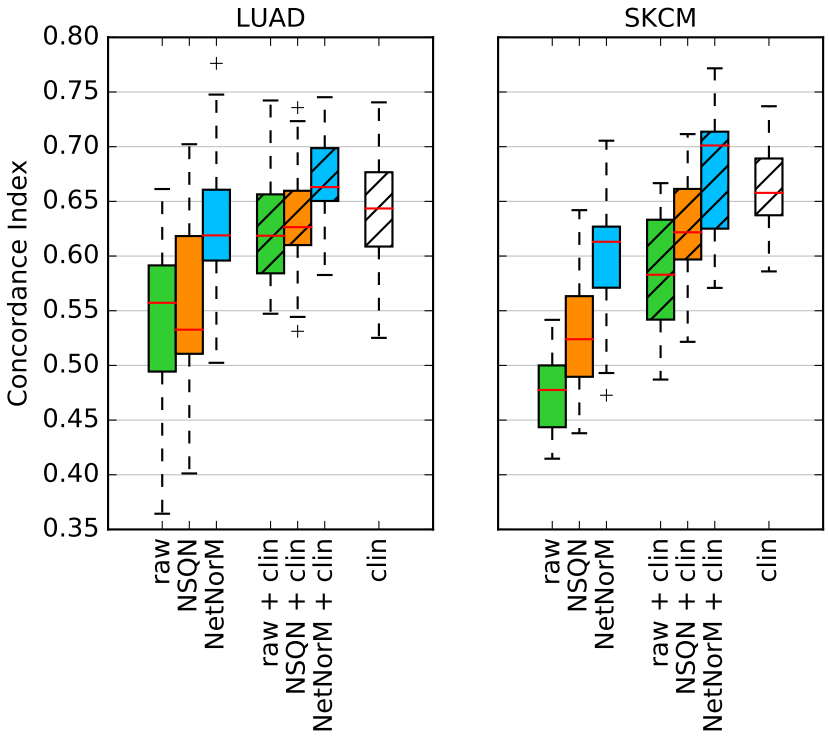


Figure 2.6 – **Survival predictive power of mutation data (raw binary mutations, mutations preprocessed with NSQN or NetNorM with Pathway Commons), clinical data, and the combination of both for LUAD and SKCM.** The combination of both data types was made by averaging the predictions obtained with each data type separately. For both cancers, samples were split 20 times in training and test sets (4 times 5-fold cross-validation). Each time a sparse survival SVM was trained on the training set and the test set was used for performance evaluation.

2.4 Patient stratification

2.4.1 NetNorM allows stable unsupervised stratification of patients with significantly different survival curves

We now assess the possibility to stratify patients into a small number of groups in an unsupervised way, meaning without using survival information, in order to identify distinct subgroups of patients in terms of mutational profiles. For that purpose, we use a standard unsupervised clustering pipeline based on nonnegative matrix factorisation (NMF), and apply it to the different cohorts of patients represented by the raw mutation profiles, or the profiles normalised by NSQN or NetNorM. The hyperparameters k (NetNorM) and α (NSQN) were set to default values chosen as the median number of mutations in a cohort for k and $\alpha = 0.5$ as recommended in [Hofree et al., 2013]. As we have no ground truth regarding "true" groups of patients, we assess the quality of clustering by two factors: (i) the stability of the clusters, assessed by the proportion of ambiguous clustering (PAC) which is the rate of discordant cluster assignments across 1,000 random subsamples of the full cohort; and (ii) the significance of association between clusters and survival.

With the raw data, NMF tends to stratify patients into very unbalanced subtypes with typically one subtype gathering the majority of patients (Fig. 2.7b). LUSC, HNSC and SKCM are extreme cases where one cluster contains 95% of the patients, whatever the number of clusters. In addition, in cases where the obtained clusters are reasonably balanced as for KIRC, the clustering stability is low. These results are coherent with [Hofree et al., 2013] who highlighted the difficulty to cluster raw mutation profiles. These undesirable behaviours disappear with both NSQN and NetNorM (Fig. 2.7). With NetNorM the obtained clusters are reasonably balanced across all cancers and the clusters are stable ($PAC \leq 30\%$). NSQN also provides stable clusters ($PAC \leq 30\%$) when the number of clusters is set between 4 and 6 however for 2 or 3 clusters the stability is not as good ($PAC \leq 50\%$). To assess the clinical relevance of the obtained subtypes, we test whether they are associated with significantly distinct survival outcomes (Fig. 2.7a). With the raw data, patient stratification is never significantly associated with clinical data. With NetNorM, significant associations of patient subtypes with survival times are achieved for HNSC, OV, KIRC and SKCM (Fig. 2.7c), while with NSQN, a significant association is only achieved for OV. The stratification based on NetNorM remains prognostic beyond clinical data for SKCM (Likelihood ratio test, $P = 2.4 \times 10^{-2}$ (SKCM, $N = 5$)). It can be surprising at first sight that no signal is recovered for LUAD with NetNorM and for SKCM with NSQN since some signal was observed in the survival prediction setting in these cases. We hypothesized that this could be due to a bad choice of the hyperparameters k and α for these cancer types. Therefore additional experiments were run for LUAD and SKCM with k and α set to their values learned by cross-validation for the survival prediction task (Table A.3). This corresponds to $k = 315$ and $\alpha = 0.6$ for LUAD (instead of $k = 189$ and $\alpha = 0.5$ as defaults) and $k = 140$ and $\alpha = 0.25$ for SKCM (instead of $k = 243$ and $\alpha = 0.5$ as defaults). With these new values for the hyperparameters, significant associations with survival are detected for LUAD with NetNorM (for 4, 5 and 6 clusters) and for SKCM with both NetNorM (for any number of clusters) and NSQN (for 4 clusters) (Fig. A.6). The recovery of a signal in these cases is in accordance with the results in the supervised setting. Overall, these results confirm the findings of [Hofree et al., 2013] that network-based normalisation with NSQN allows stratifying patients

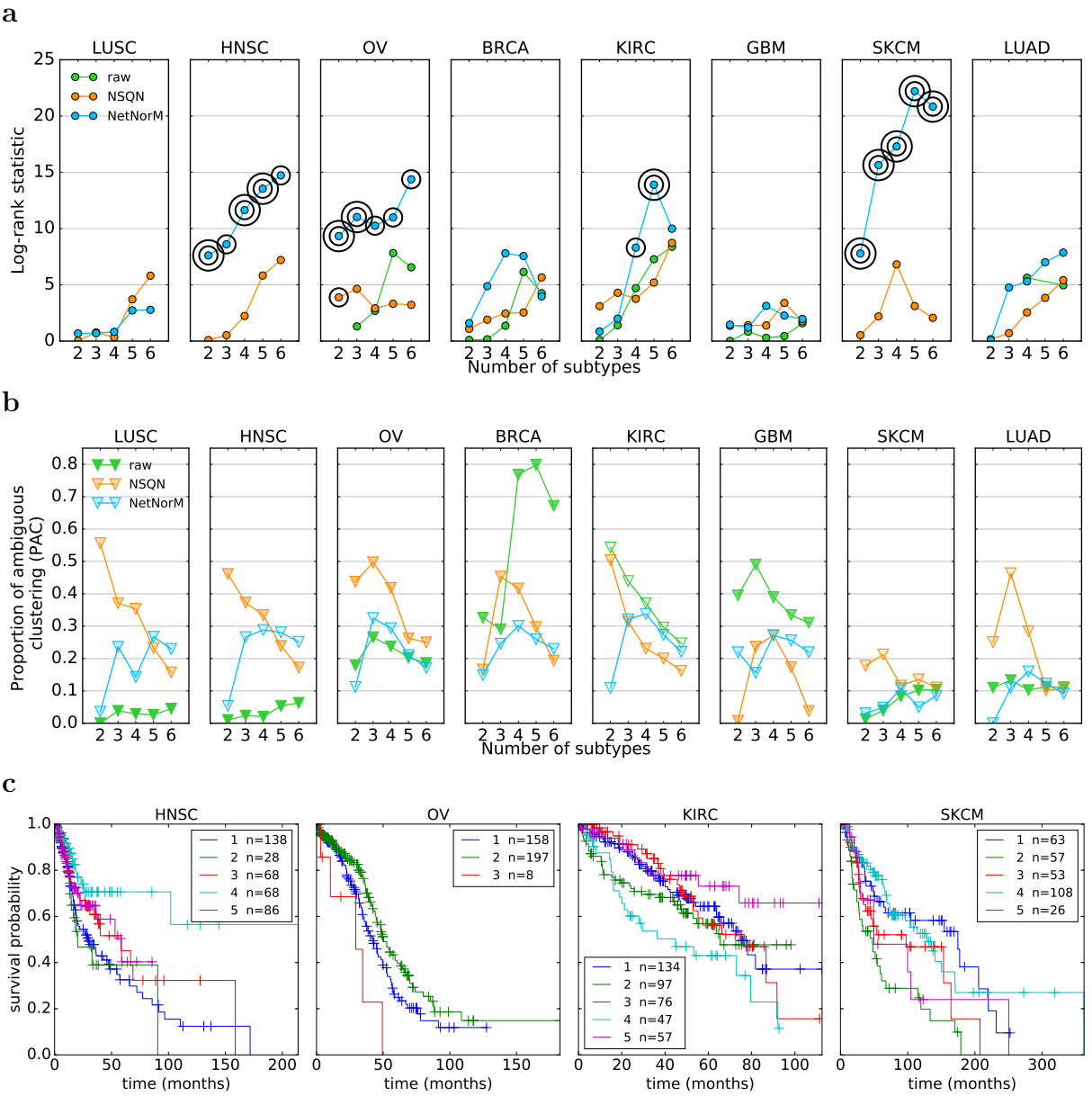


Figure 2.7 – Comparison of patient stratifications obtained with the raw mutation data, NSQN (Pathway Commons) and NetNorM (Pathway Commons) for 8 cancer types. (a) Association of patient subtypes with survival time. One circle indicates $P \leq 0.05$ and two concentric circles indicate $P \leq 0.01$ (log-rank test). Cases where clusters were too unbalanced (95% of the patients in one single cluster) are not shown. (b) Evaluation of the clustering stability as measured by the proportion of ambiguous clustering (PAC). The transparency of the triangles indicate the percentage of patients in the largest cluster. The scale ranges from 100% (totally opaque) to $\frac{1}{N}$ % (totally transparent) where N is the number of subtypes. Therefore opacity (resp. transparency) indicate unbalanced (resp. balanced) clusters. (c) Kaplan Meir survival curves for NetNorM subtypes with significantly distinct survival outcomes. In the legend are indicated the subtype number followed by the number of patients in the subtype.

better than the raw mutation profiles, and also show that the stratification obtained from NetNorM normalisation is both more stable and more clinically relevant than the one obtained with NSQN.

2.4.2 Patient stratification with randomised networks

We now assess whether the biological information contained in Pathway Commons is crucial to obtain subtypes with significantly distinct survival outcomes. For that purpose, we carry out patient stratification with NSQN and NetNorM using 10 randomised versions of Pathway Commons for HNSC, OV, KIRC and SKCM. As for the survival prediction experiment, the randomisation involves shuffling the vertices' labels so as to keep the structure of the network unchanged. Surprisingly, network randomisation does not affect the log-rank statistic obtained for HNSC and SKCM. This suggests that although NetNorM generates subtypes with more distinct survival times than NSQN for HNSC and SKCM, it does not benefit from Pathway Commons gene-gene interaction knowledge. Rather it exploits the prognostic information contained in the raw mutation profiles as well as the overall mutational burdens as captured by proxy mutations. Regarding KIRC and OV, NetNorM produces subtypes with significantly different survival times with 4 and 5 clusters for KIRC and for any number of clusters for OV. In the case of KIRC, the real network yields the subtypes with the most distinct survival times (N=5) (Fig. 2.8) while in the case of OV, most randomised networks (at least 15 out of 20 for each number of clusters) produce subtypes with worse association to survival time. This indicates that for KIRC and presumably for OV, NetNorM takes advantage of gene-gene interaction knowledge to stratify patients into clinically relevant subtypes. This is also clearly the case for LUAD with NetNorM when the hyperparameter k is set to its value learned by cross-validation in the survival prediction setting (Fig. A.6).

2.4.3 Patient subtypes obtained with NetNorM are characterised by distinct pathways

To interpret biologically the subgroups of patients identified by automatic stratification after NetNorM normalisation, we look at differentially mutated genes and pathways across subtypes. We focus on LUAD with N=5 groups as a proof of principle with k set to its value learned by cross-validation in the supervised setting. This choice is motivated by the fact that LUAD is the most promising cancer type for supervised survival prediction and produces interesting results in the unsupervised setting. As the basis vectors or "metapatients" yielded by the NMF summarise the mutational patterns found in the different subtypes, we analyse genes in terms of their weight in the different metapatients, and restrict our attention to the approximately 900 genes displaying highest variance (variance greater than 0.01) across basis vectors since these genes are expected to be the most differentially mutated across subtypes. Interestingly, this gene list comprises most significantly mutated genes in LUAD including *TP53*, *KRAS*, *KEAP1*, *EGFR*, *NF1*, *RB1* [Collisson et al., 2014; Ding et al., 2008]. To analyse these genes we cluster them into groups with similar weights across basis vectors using hierarchical clustering (Fig. 2.9b), and we test for enrichment in known biological pathways the 20 gene clusters (GCs) obtained.

One first observation is that the 5 patient subtypes have distinct overall mutational burdens with groups 4 and 5 (resp. 2 and 3) gathering patients with many (resp. few) mutations (Fig. 2.9e). This confirms the fact that NetNorM-normalised profiles contain information about

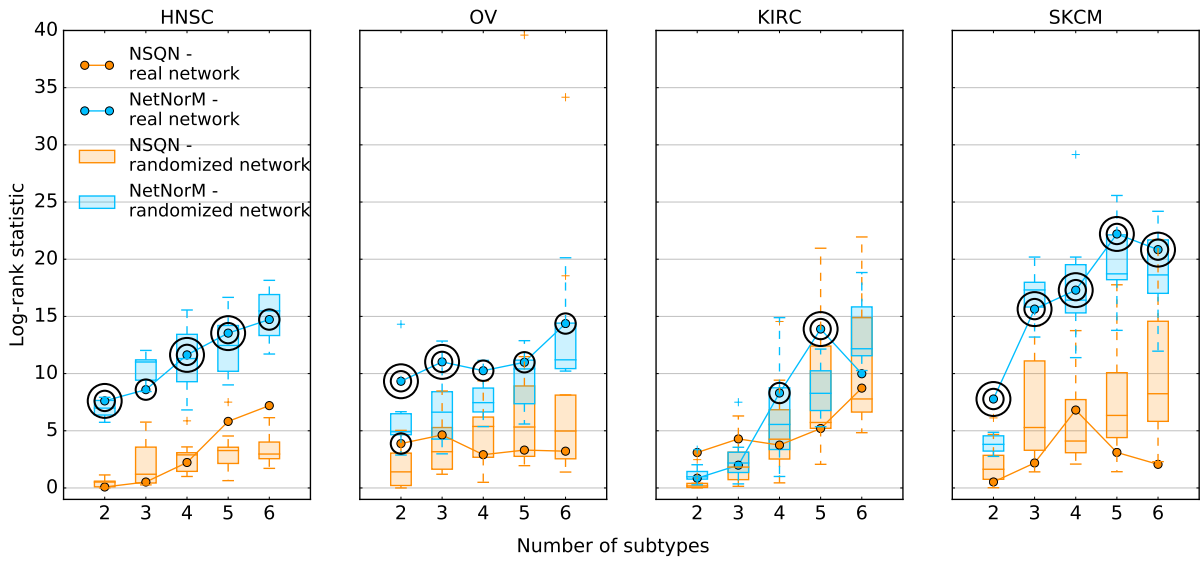


Figure 2.8 – Effect of network randomisation on patient stratification. Log-rank statistic obtained with Pathway Commons (curve) and 10 randomised versions of Pathway Commons (boxplots) with NetNorM (blue) and NSQN (orange) for HNSC, OV, KIRC and SKCM. One circle indicate a P-value $P \leq 5 \times 10^{-2}$ and two concentric circles indicate $P \leq 1 \times 10^{-2}$.

the initial number of mutations, although they are normalised to a fixed number of mutations. More importantly, most GCs exhibit high weights in one metapatient and low weights in others, suggesting that they are mainly enriched in mutations in one single patient subtype (Fig. 2.9b). χ^2 contingency tests (see Methods) for each GC confirms that for most of them (17/20), the distribution of the mutations across patient subtypes is not that expected according to subtypes' overall mutational burdens ($P < 5 \times 10^{-2}$) (Table A.4). The contribution of each subtype to the test statistic for each GC also confirms that GCs are often enriched in mutations in mainly one patient subtype (Fig. 2.9d). Subtypes could thus easily be associated with one or several GCs, and therefore pathways through pathway enrichment analysis using the KEGG database [Kanehisa et al., 2016] (see Methods).

Consequently, subtype 3 is characterised by an enrichment in mutations in genes associated with ribosomes and spliceosomes (GCs 2, 3, 4, 5, 6, 7, 8, 17, 18, 19) (Table A.4). Subtype 1 is enriched in mutations in two very small gene clusters (GCs 11 and 16): the first one consists of four genes including *KRAS* and the second one only includes *MUC16*. These two subtypes are those with poorest survival probability. Subtype 4 is mainly enriched in late replicating genes (GC 10) (Fig. 2.9c). This reflects the fact that subtype 4 is enriched in highly mutated patients as there exists a positive correlation between somatic mutation frequency and genes replication time [Lawrence et al., 2013]. Subtype 2 is enriched in mutations in genes related to endocytosis and phagosomes (GCs 16, 1, 11). Finally, subtype 5 is very strongly associated with gene clusters 9 and 13. Gene cluster 9 is enriched in genes from the cAMP and PI3K-Akt signaling pathways. Gene cluster 13 could not be significantly associated to a known biological pathway. However it contains *FANCD2* (Fanconi Anemia Complementation Group D2) which is involved in double-strand breaks DNA repair and the maintenance of chromosomal stability [Moldovan and D'Andrea, 2009]. We note that 12 of the 15 patients in subtype 4 present the

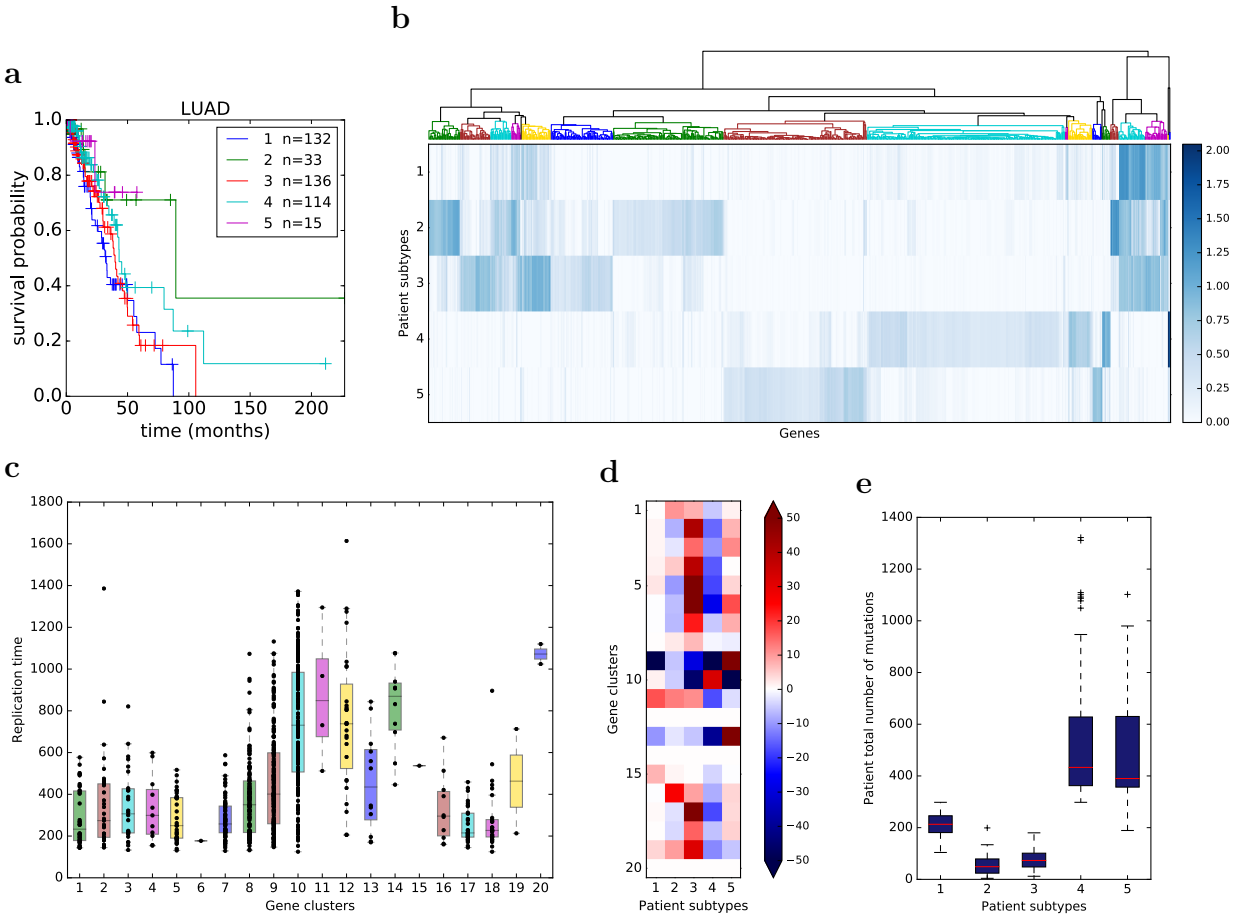


Figure 2.9 – Characterisation of LUAD patient subtypes obtained with NetNorM ($N = 5$ groups, k=315, Pathway Commons). (a) Kaplan Meir survival curves for NetNorM subtypes with significantly distinct survival outcomes. In the legend are indicated the subtype number followed by the number of patients in the subtype. (b) Metapatients matrix obtained by applying NMF to mutation profiles processed with NetNorM. The matrix shown is restricted to the genes with highest variance across metapatients. The genes (columns) are clustered via hierarchical clustering. Clusters are numbered from 1 to 20 from left to right. (c) Distribution of gene replication times across gene clusters. (d) A χ^2 contingency test was performed for each gene cluster to test its enrichment (or depletion) in mutations across patient subtypes given the subtypes' marginal number of mutations. The value represents the contribution of a subtype to the test statistic, and the colour indicates an enrichment (red) or a depletion (blue) in mutations. (e) Distribution of patients' total number of (raw) mutations across patient subtypes.

same 4-nucleotides splice site deletion in *FANCD2*, whereas across the rest of the 430 patients *FANCD2* is mutated in 6 patients only, and only one of these 6 mutations is the same as that observed in subtype 4 patients.

2.5 Discussion

Exploiting the wealth of cancer genomic data collected by large-scale sequencing efforts is a pressing need for clinical applications. Somatic mutations are particularly important since they may reveal the unique history of each tumour at the molecular level, and shed light on the biological processes and potential drug targets dysregulated in each patient. Standard statistical techniques for unsupervised classification or supervised predictive modelling perform poorly when each patient is represented by a raw binary vector indicating which genes have a somatic mutation. This is both because the relevant driver mutations are hidden in the middle of many irrelevant passenger mutations, and because there is usually very little overlap between the somatic mutation profiles of two individuals. NetNorM aims to increase the relevance of mutation data for various tasks such as prognostic modelling and patient stratification by leveraging gene networks as prior knowledge.

One important aspect of NetNorM is the property that, after normalisation, all patients have the same number of 1's in their normalised mutation profile. Although there is no biological rational for this constraint, we believe that the fact that all normalised samples have the same distribution of values is an important property for many high-dimensional statistical methods such as survival models or clustering techniques to work properly. To support this claim, we notice that the Network-based stratification (NBS) method proposed in [Hofree et al., 2013] performs a quantile normalisation step after network smoothing. To investigate whether the quantile normalisation step in NSQN plays an important role, we applied network smoothing without quantile normalisation (called NS) and performed survival prediction and patients stratification with this representation of the mutations. Surprisingly, NS does not improve over the raw mutation profiles for both LUAD and SKCM (Fig. 2.10c). Moreover just as the raw data, NS is unable to stratify patients into approximately balanced clusters (Fig. 2.10b). This suggests that quantile normalisation plays a crucial role in the performances obtained with NSQN, in spite of non obvious biological justification for this step.

Another important difference between NSQN and NetNorM is the fact that NetNorM only exploits mutation information about direct neighbours in the network, while NSQN can potentially diffuse a mutation further than the direct neighbours. However, we found that NSQN does not benefit from this possibility. Indeed, we tested a simplified version of NSQN where the network propagation is stopped after one iteration, and assessed the performance of the corresponding method which we call SimpNSQN. For survival prediction, we observe no significant difference between NSQN and SimpNSQN (Fig. 2.10c). For patient stratification, SimpNSQN produces subtypes that are vey similar to those produced by NSQN (Fig. 2.10d). Therefore the subtypes generated by both methods associate equally well to clinical data, and even slightly better for SimpNSQN in the case of LUAD (Fig. 2.10a). Overall, these pieces of information indicate that the useful information created by NSQN is mostly concentrated on shared mutated order 1 neighbourhoods, and explain why we observe no loss in performance with NetNorM which explicitly restricts the diffusion of mutations to direct neighbours only. More generally, these elements also indicate that diffusion to indirect neighbours is still difficult with current

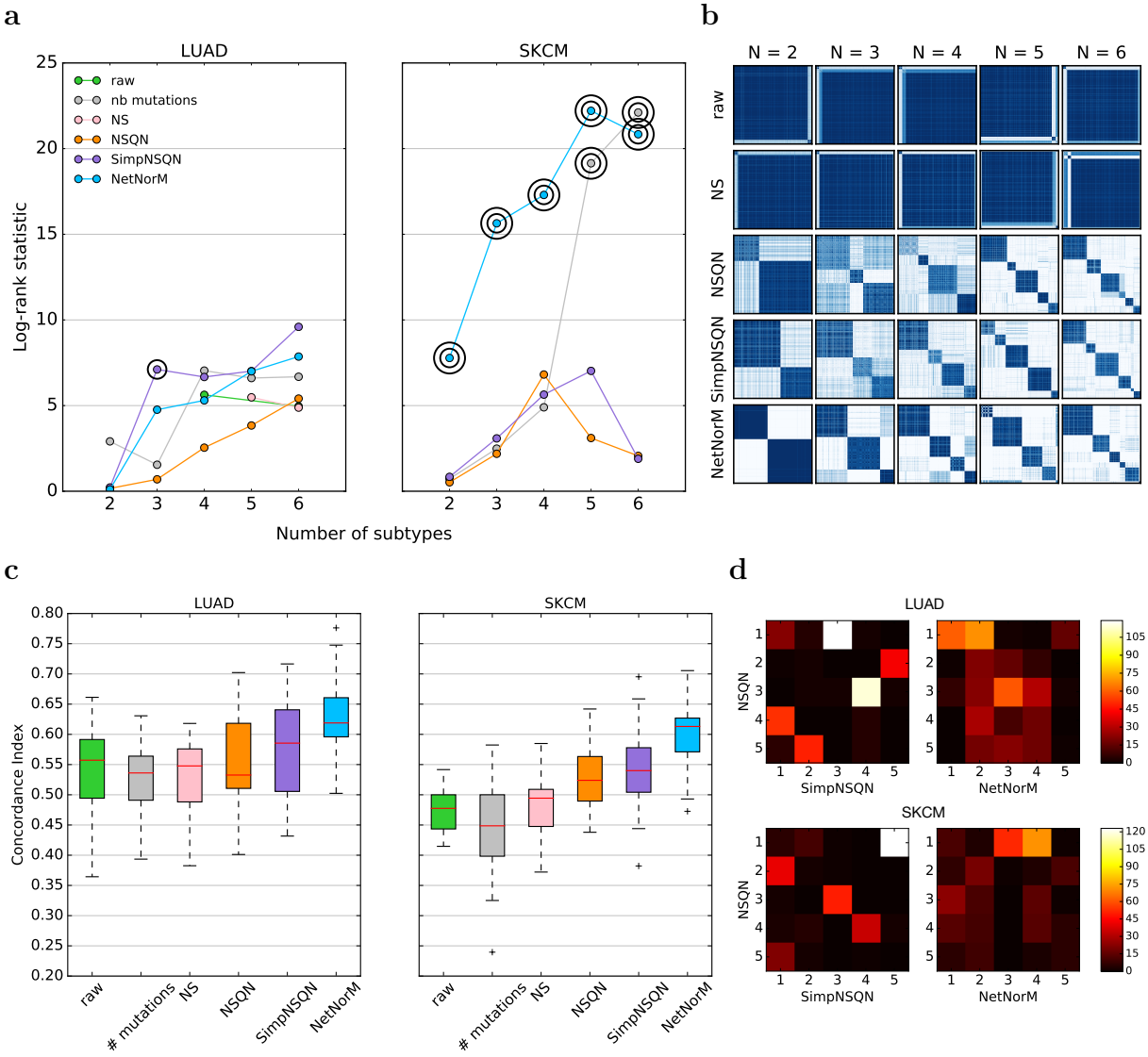


Figure 2.10 – **Exploring NSQN and NetNorM performances levers.** (a) Subtypes log-rank statistic obtained for LUAD (left) and SKCM (right). One circle indicate a P-value $P \leq 5 \times 10^{-2}$ and two concentric circles indicate $P \leq 1 \times 10^{-2}$ (log-rank test). (b) Consensus clustering matrices for LUAD. (c) Survival prediction performances for LUAD (left) and SKCM (right). (d) Confusion matrices for LUAD (top) and SKCM (bottom) comparing the subtypes obtained with NSQN and SimpNSQN on the one hand, and NSQN and NetNorM on the other hand. (a, b, c, d) were obtained with Pathway Commons.

methods. This is a likely consequence of the small world property of biological graphs [Watts and Strogatz, 1998]. Because the path between any two genes is usually short, diffusion even to order-2 neighbours reaches a substantial number of genes, and therefore the resulting signal observed for one gene is the superposition of a large number of signals originating from close mutations.

NetNorM encodes information about patients' total number of mutations in the raw data, and potentially can exploit it if this information is relevant for the problem at hand. However we found that the total number of mutations is a poor predictor or survival (Fig. 2.10c), and a poor feature for LUAD patient stratification (Fig. 2.10a). This confirms that NetNorM conserves useful information regarding both the total mutational burden of a patient and the distribution of the mutations on the gene network, and manages to leverage both types of information. In addition to mutational burdens, NetNorM also encodes information about genes' NMB which proved to carry some prognostic power. The fact that NMB might reveal new insights into mutation profiles is an emerging idea supported by this study. Further support has been formalised with two recently published methods [Cho et al., 2016; Horn et al., 2015] which rely on NMB to achieve state-of-the-art performances for cancer gene discovery.

We emphasise that randomised gene networks lead to significantly worse performances than the real network for survival prediction as well as for patient stratification for several cancers. While it is not always clear whether incorporating gene networks as prior knowledge does help for a given task, this provides a sound argument that such prior knowledge is effectively leveraged with NetNorM.

Increasing the relevance of mutation data to various tasks is a broad project and NetNorM could be extended in many ways. First, although NetNorM was successful for LUAD and SKCM, we note that the method brings few improvements compared to the raw data for the remaining cancer types. Therefore extensive efforts are needed to determine whether it is possible to design representations of mutations that would increase the statistical power of models learned on these datasets. Second, NetNorM does not integrate further information about mutations such as their predicted functional impact. A possible extension could therefore include this type of information. Finally, the distribution of values for the normalised profiles is defined as the mean distribution of the original profiles in the case of NSQN, and simply a binary vector with a fixed number of 1's in the case of NetNorM, however these choices are empirical. This suggests that an interesting future work may be to assess more precisely the effect of this distribution and, perhaps, optimise it for each specific task.

2.6 Materials and Methods

Patient mutation profiles preprocessing

Whole exome somatic mutation calls (MAF files) were downloaded from TCGA data portal (`https://tcga-data.nci.nih.gov/tcga`) for 8 cancer types (LUAD, SKCM, GBM, BRCA, KIRC, HNSC, LUSC, OV) (Table 2.1). The data include point mutations (single nucleotide polymorphism as well as di/tri/oligo-nucleotide polymorphism) and indels. Silent mutations were filtered out and mutations profiles were defined as binary vectors with ones whenever a patient is mutated in a given gene and zeros otherwise.

Gene-gene interaction network

Pathway Commons (http://www.pathwaycommons.org/pc2/downloads) was used throughout this work (Pathway Commons v6, SIF format). It integrates gene networks from several public databases and aggregates both genetic and protein-protein interactions (PPIs). PPIs refer to physical contacts established between proteins while genetic interactions refer to interactions through regulatory and signalling pathways. To remove interactions involving small molecules in Pathway Commons, the following interaction types were filtered out: "consumption-controlled-by", "controls-production-of", "controls-transport-of-chemical", "chemical-affects", "reacts-with", "used-to-produce", "SmallMoleculeReference", "ProteinReference;SmallMoleculeReference", "ProteinReference". We obtained a network with 16,674 nodes (genes) and 2,117,955 edges (interactions). For the survival prediction task, we also tested the following gene networks: BioGRID v3.4.131, HPRD release 9, HumanNet v1 and STRING v10. For HumanNet and STRING, only the top 10% most confident interactions were retained.

Network based Normalisation of Mutation profiles (NetNorM)

NetNorM is a method that integrates patients mutation profiles with a gene network to produce normalised mutation profiles where all patients have the same number k of mutations. The target number of mutations k is a tuning parameter. In the context of survival prediction (supervised setting), it is learned by cross-validation while for patient stratification (unsupervised setting), it is set as the median number of mutations in a cohort, or alternatively to the median best k learned across cross-validation folds for survival prediction. Concretely, NetNorM defines a ranking over genes separately for each patient and then use this ranking to normalise mutation profiles. The ranking defined in NetNorM is obtained with a simple two-step procedure. First, genes are ranked according to their mutation status with mutated genes ranked higher than non mutated genes. Then, mutated genes are ranked according to their degree (i.e. their number of neighbours) and non mutated genes are ranked according to their number of mutated neighbours. The normalisation is then obtained by considering the k highest ranked genes as mutated while the rest of the genes will be considered non mutated. By construction, mutated genes are always ranked higher than non-mutated genes. Therefore patients with a lot of mutations will have mutations removed while patients with few mutations will hold artificial proxy mutations. Note that when the obtained ranking contains ties, all genes are given distinct ranks according to the order in which they occur in the mutation matrix.

Network smoothing with quantile normalisation (NSQN)

Network smoothing propagates the influence of mutations over gene-gene interaction networks. It was implemented according to the following update function [Hofree et al., 2013]:

$$\boldsymbol{X_{t+1}} = \alpha \boldsymbol{X_t} \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}} + (1-\alpha)\boldsymbol{X_0}$$

where $\boldsymbol{X_t}$ is the patient $\times$ genes mutation matrix at iteration t, $\boldsymbol{X_0}$ is the initial binary mutation matrix, $\boldsymbol{A}$ is the adjacency matrix representing the network used and $\boldsymbol{D}$ is the diagonal degree matrix where $D_{ii} = \sum_j A_{ij}$. α is a tuning parameter controlling the length of diffusion paths

over the network. Similarly to the parameter k in the context of NetNorM, it is learned by cross-validation for survival prediction (supervised task) while for patient stratification (unsupervised task) it is set as $\alpha = 0.5$ as recommended in [Hofree et al., 2013] with Pathway Commons or alternatively to the median best α learned across survival prediction cross-validation folds. The update function is applied until convergence, and the resulting smoothed matrix is then quantile normalised so that all patients have the same mutation distribution.

Simplified version of NSQN (SimpNSQN)

The simplified version of NSQN does not propagate mutations further than to order 1 neighbours in the network. More precisely, the SimpNSQN score of a gene is equal to its number of mutated neighbours normalised by its degree and by the degrees of its neighbours, plus a constant if the gene is mutated. This is obtained by computing:

$$\boldsymbol{X} = \alpha \boldsymbol{X_0} \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}} + (1-\alpha)\boldsymbol{X_0}$$

where $\boldsymbol{X_0}$ is the initial binary mutation matrix, $\boldsymbol{A}$ is the adjacency matrix representing the network used, $\boldsymbol{D}$ is the diagonal degree matrix where $D_{ii} = \sum_j A_{ij}$ and $\alpha \in \mathbb{R}$ is a tuning parameter. Note that SimpNSQN uses the same update equation as NSQN but it is run only once.

Sparse survival SVM

To estimate a survival model from high-dimensional mutation profiles, we use a survival SVM model [Van Belle et al., 2007] combined with a sparsity-inducing regularisation to automatically perform gene selection. Let $\delta_i = 1$ (resp. $\delta_i = 0$) if patient i is deceased (resp. censored), and $y_i \in \mathbb{R}$ be the observed survival time of patient i. It corresponds to either a failure or a censoring time depending on whether the patient is deceased or censored. Define $Z \in \{0,1\}^{n \times n}$ which indicates whether a pair of patients is comparable, i.e,

$$Z_{ij} = \begin{cases} 1 \text{ if } (y_i < y_j \text{ and } \delta_i = 1) \text{ or } (y_j < y_i \text{ and } \delta_j = 1)\,, \\ 1 \text{ if } (y_i = y_j \text{ and } (\delta_i = 1 \text{ or } \delta_j = 1))\,, \\ 0 \text{ otherwise }. \end{cases}$$

Finally, let $\boldsymbol{x_i} \in \{0,1\}^p$ be the mutation profile of patient i. The survival time of patient i is modelled as $s_i = \boldsymbol{w^T x_i}$ where $\boldsymbol{w} \in \mathbb{R}^p$ is the model parameter learned using ranking Support Vector Machines (rSVM) as in [Van Belle et al., 2007]. However to get a sparse $\boldsymbol{w}$ we introduce an ℓ_1 regularisation instead of the ℓ_2 regularisation in [Van Belle et al., 2007] and thus solve the following optimisation problem:

$$\underset{w}{\text{minimise}}\ \frac{1}{2}||w||_1 + C\sum_{i,j} Z_{ij}\ \ell_{hinge}(w^T(\boldsymbol{x_j} - \boldsymbol{x_i}))\,,$$

where $\ell_{hinge}(u) = \max(1-u, 0)$ is the hinge loss and $C \in \mathbb{R}$ is the regularisation parameter. To solve this problem we used the support vector classification algorithm `svm.LinearSVC` from the Python package `scikit learn` [Pedregosa et al., 2012]. This optimisation problem maximises a

convex relaxation of the Concordance Index (CI) which measures how well the predicted survival times $\boldsymbol{s}$ are in accordance with the observed survival times $\boldsymbol{y}$ for the comparable pairs of patients. Formally, CI $= \frac{1}{|Z|} \sum_{y_i \leq y_j} Z_{ij} I(s_j - s_i)$ where

$$I(x) = \begin{cases} 1 \text{ if } x > 0\,, \\ \frac{1}{2} \text{ if x} = 0\,, \\ 0 \text{ otherwise}\,, \end{cases}$$

and $|Z| = \sum_{y_i \leq y_j} Z_{ij}$. To evaluate the CI obtained on a given dataset, samples were split in 80% train and 20% test sets 20 times using 4 five-fold cross-validation. Each time, a model was learned on the training set and tested on the test set. The CI was computed according to a python implementation of the function `estC` from the R package `compareC`. Hyperparameters were learned thanks to an inner 5-fold cross-validation on the training set. The values tested for C ranged from 1×10^{-4} to 1×10^{2} included in log scale. The values tested for α ranged from 0.1 to 0.9 included with steps of 0.1. Finally the values tested for k were chosen to span a grid from k_{min} and k_{max} with steps of 2, where k_{min} and k_{max} are the first and third quartiles of the distribution of patients' total number of mutations. k_{min} and k_{max} differ for each cohort (Table A.2).

Patient stratification

Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ be the matrix with patient mutations profiles as rows. To cluster the patients we perform a non-negative matrix factorisation (NMF) on $\boldsymbol{X}$, i.e., solve the following optimisation problem:

$$\underset{\boldsymbol{W},\boldsymbol{H}>0}{\text{minimise}} \; ||\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}||_2^2\,,$$

where $\boldsymbol{H} \in \mathbb{R}^{N \times p}$ defines N basis vectors or "metapatients" and $\boldsymbol{W} \in \mathbb{R}^{n \times N}$ defines basis vectors loadings. Patient i was then assigned to the group $j \in \{1..N\}$ that represents him best i.e. $\underset{j}{argmax}\ W_{ij}$. To promote robust cluster assignments, NMF was applied 1000 times to subsamples of the dataset composed of 80% of the samples and 80% of the features chosen at random without replacement. A consensus matrix $\boldsymbol{C} \in \mathbb{R}^{n \times n}$ was then derived from the 1000 cluster assignments obtained where each entry C_{ij} corresponds to the frequency at which two patients where clustered in the same group over all samplings where both patients were retained. The final cluster assignment was obtained by applying hierarchical clustering to the consensus matrix with euclidean distance and average linkage.

To assess the stability of the obtained clusters, we computed the proportion of ambiguous clustering (PAC) which is the proportion of discordant cluster assignments obtained through consensus clustering. Cluster assignments for a pair of patients (i, j) were considered discordant when $0.25 \leq C_{ij} \leq 0.75$.

In the case where only the total number of mutations was used for stratification, NMF is not applicable and kMeans was used instead with 1000 restarts and initialisation by kMeans++ [Arthur and Vassilvitskii, 2007].

Analysis of the proxy genes selected by the sparse survival SVM with NetNorM

Several proxy genes have a prognostic power according to log-rank tests performed for each gene separately and which compare patients with mutations (proxy or not) versus those without ($P \leq 1 \times 10^{-2}$). The difference in survival outcomes observed may be due to at least two types of information encoded in proxy genes: patients' overall mutational burden and genes' neighbourhood mutational burden (NMB). To clarify the contributions of each effect, we investigate whether such distinct survival outcomes can be obtained with proxies for the total number of mutations only, regardless of NMBs. To this end, we simulate proxy mutations for each gene separately according to a model that only depends on patients' total number of mutations. Let $T_i \in \mathbb{N}$ be the total number of mutations of patient i, $i \in \{1, ..., n\}$. Let $\mathcal{M}_o \subset \{1, ..., n\}$ and $\mathcal{M}_p \subset \{1, ..., n\}$ indicate which patients have original and proxy mutations respectively. For a given proxy gene whose mutations are described by the sets M_o and M_p, we leave the original mutations untouched and reallocate the proxy mutations according to

$$P(i \in \mathcal{M}_p | T_i) = \begin{cases} 0 \text{ if } (T_i \geq k) \text{ or } (i \in \mathcal{M}_o) \\ \frac{k - T_i}{\alpha} \text{ otherwise} \end{cases}$$

where α is chosen so that the probabilities sum to 1. Proxy mutations are drawn from this model 1000 times. Each time we compute the log-rank statistic between the mutated and non mutated patients which yields a distribution of the log-rank statistic under the null hypothesis. The actual log-rank statistic obtained using NetNorM is then compared to this distribution to accept or reject the null hypothesis. Rejecting the null hypothesis means that the difference in survival outcomes observed between the patients with and without artificial mutations is not only driven by patients' total number of mutations.

Survival analysis using patient subtypes and clinical data

To determine whether the obtained patient subtypes are predictive of survival beyond clinical data, we fitted a Cox proportional hazards regression model to the clinical data and to the clinical data augmented with a variable describing patients' subtypes. We then performed a likelihood ratio test to compare the two models. The clinical variables used were downloaded from TCGA. It includes age, gender, stage, extent of spread to the lymph nodes, presence of metastasis, histology for both LUAD and SKCM and further variables such as smoking history, history of prior malignancy, residual tumour after surgery, tumour dimensions for LUAD and clark level at diagnosis, primary melanoma mitotic rate, new tumour event after initial treatment (yes/no), primary melanoma tumour ulceration (yes/no), primary melanoma known (yes/no) for SKCM.

Identifying differentially mutated genes and pathways across subtypes

We obtain gene clusters by applying hierarchical clustering with centroid linkage and Euclidean distance to the columns of the metapatients matrix (restricted to high variance genes). To obtain a reasonable number of gene clusters to analyse, we cut the hierarchical cluster tree at a distance threshold of 5.5, yielding 20 clusters. Gene clusters can be categorised into two types: those that contain a lot of proxy mutations ($\geq 80\%$ of the total mutational load of the cluster) and whose

genes form a dense subgraph, and those that have neither of these two features. The presence of dense subgraphs with many proxy mutations results from the fact that NetNorM tends to add proxy mutations to all genes in a dense subgraph or none since they all have roughly the same number of mutated neighbours. The association of a gene cluster with one subtype can therefore indicate two things: either the subtype is expected to be enriched in proxy mutations in the corresponding gene cluster, which in turn indicates that the subgraph in which the cluster lies is expected to be enriched in mutations, or the gene cluster itself is expected to be enriched in mutations in the corresponding subtype. The enrichment or depletion in mutations of one gene cluster across patient subtypes was therefore tested slightly differently according to the gene cluster type. In the first case, we first define the neighbourhood of the gene clusters as all genes lying in the same dense subgraph. Specifically, we include in the subgraph all genes sharing an edge with at least 90% of the genes in the cluster, thus keeping subgraphs very dense. The obtained set of genes is the one tested for enrichment in mutations across subtype. In the second case, the gene cluster is directly tested for enrichment. Enrichment is assessed with a χ^2 contingency test, where the contingency table is defined by the following marginals: the total number of raw mutations in each subtype, and the total number of raw mutations in and outside the gene cluster (generalised to the embedding of a dense subgraph if it is relevant).

Gene clusters are searched for pathway enrichment using DAVID online tool [Huang et al., 2009] (https://david.ncifcrf.gov/summary.jsp) with the KEGG database [Kanehisa et al., 2016]. They are also tested for enrichment in late replicating genes thanks to a permutation test using data downloaded from http://www.broadinstitute.org/cancer/cga/mutsig_run. For each gene cluster c of length l_c, l_c genes are chosen uniformly at random without replacement from the list of genes with replication time information. This sampling is performed 1000 times and the null distribution was obtained by computing the median replication time of these 1000 gene sets. The median replication time of cluster c is then compared to the null distribution to yield a p-value, i.e. the probability to observe a set of genes of length l_c with median replication time at least as extreme.

Chapter 3

Supervised Quantile Normalisation

Contents

3.1 Introduction . . . 61
3.2 Quantile normalisation (QN) . . . 62
3.3 Supervised quantile normalisation (SUQUAN) . . . 63
3.4 SUQUAN as a matrix regression problem . . . 64
3.5 Algorithms . . . 65
3.5.1 SUQUAN-SVD . . . 66
3.5.2 SUQUAN-BND and SUQUAN-SPAV . . . 66
3.6 Experiments . . . 68
3.6.1 Simulated data . . . 68
3.6.2 CIFAR-10 dataset . . . 68
3.6.3 Gene expression data . . . 71
3.7 Discussion . . . 73

Abstract

Quantile normalisation is a popular normalisation method for data subject to unwanted variations such as images, speech, or genomic data. It applies a monotonic transformation to the feature values of each sample to ensure that after normalisation, they follow the same target distribution for each sample. Choosing a 'good' target distribution remains however largely empirical and heuristic, and is usually done independently of the subsequent analysis of normalised data. We propose instead to couple the quantile normalisation step with the subsequent analysis, and to optimise the target distribution jointly with the other parameters in the analysis. We illustrate this principle on the problem of estimating a linear model over normalised data, and show that it leads to a particular low-rank matrix regression problem that can be solved efficiently. We illustrate the potential of our method, which we term SUQUAN, on simulated data, images and genomic data, where it outperforms standard quantile normalisation.

Résumé

La normalisation par les quantiles est une méthode répandue pour la normalisation de données sujettes à des variations indésirables commes les images, les données audio ou les données génomiques. Cette méthode applique une transformation monotone aux valeurs qui décrivent chaque échantillon afin de garantir que, après normalisation, ces valeurs suivent la même distribution cible pour chaque échantillon. Le choix d'une 'bonne' distribution cible reste cependant largement empirique et heuristique, et est généralement fait indépendamment de l'analyse ultérieure des données normalisées. Au lieu de cela, nous proposons de coupler la normalisation par les quantiles avec l'analyse ultérieure, et d'optimiser la distribuiton cible conjointement avec les autres paramètres de l'analyse. Nous illustrons ce principe pour l'estimation d'un modèle linéaire sur des données normalisées, et montrons qu'il mène à un problème de régression matricielle de faible rang qui peut être résolu efficacement. Nous illustrons le potentiel de cette méthode, que nous appellons SUQUAN, sur des données simulées ainsi que sur des images et des données génomiques où elle surpasse la normalisation par les quantiles standard.

3.1 Introduction

In many application fields where data are collected for a particular task, data acquisition is often plagued with various sources of perturbations which induce unwanted variations in the captured data and make the task harder to solve. For example, two photos of the same object taken from the same position may still vary considerably in terms of colour distribution or other statistical properties depending on the ambient light, the device used to take the picture, or the person in charge of taking the picture [Gonzalez and Woods, 2008]. Similarly, pixel intensities of an MRI scan do not have a fixed meaning and can vary considerably between two scans on the same patient with the same protocol and same scanner [Shinohara et al., 2014]; speech recognition is challenging in part because the acoustic signal corresponding to a given word varies a lot with the speaker, the noise pollution around and the device used to capture the signal [Hilger and Ney, 2006]; and microarray - or sequencing - based measurements in genomics are famous for being extremely sensitive to a variety of unwanted perturbations such as temperature, sample preparation protocol, or amount of material [Bullard et al., 2010].

In order to reduce the burden of unwanted variations for subsequent data analysis applications, the standard way to proceed is often to *normalise* the data prior to any analysis, in order to remove unwanted variations as much as possible while keeping relevant signals. Normalisation procedures vary from simply centering and scaling each sample to impose a common scale across samples, to more sophisticated and data-specific procedure, e.g., [Bullard et al., 2010]. In this work we are interested in a particular normalisation procedure, pervasive across different fields and known under different names, which monotonically modifies the entries of a given sample so that after normalisation, all samples have the same distribution of entries. Following the terminology used in biostatistics [Hicks and Irizarry, 2015], we refer to this procedure as *quantile normalisation* (QN). QN is ubiquitous in high-dimensional biological data analysis, where samples are often corrupted by various technical or biological unwanted variations, and is widely used for many types of data including low-density [Amaratunga and Cabrera, 2001; Yang and Thorne, 2003] or high-density [Bolstad et al., 2003; Irizarry et al., 2003] microarray data for gene expression analysis, high-density microarray for genotyping [Carvalho et al., 2007; Scharpf et al., 2011], RNA-seq sequencing data for gene expression analysis [Bullard et al., 2010; Cloonan et al., 2008; Dillies et al., 2013], microarray data for DNA methylation analysis [Yousefi et al., 2013], or ChIp-seq sequencing data for protein-DNA interaction analysis [Bilodeau et al., 2009; Kasowski et al., 2010]. QN is also widely used in image processing under the name of *histogram matching*, or more specifically *histogram equalisation* when the pixel intensities of an image are monotonically transformed in such a way that the distribution of values becomes approximatively uniform [Gonzalez and Woods, 2008]. A popular application of histogram matching is in MRI brain imaging, where a popular approach to preprocess images is to apply a variant of QN proposed by Nyúl and Udupa [1999] and refined by Nyúl et al. [2000] and Shah et al. [2011]. Similarly, another variant of QN targeting a uniform distribution is popular in speech recognition under the name of *histogram normalisation* [Dharanipragada and Padmanabhan, 2000; Hilger and Ney, 2006; Molau et al., 2001]. In geostatistics, a popular trick to analyse non-gaussian spatial data is to perform a *Gaussian anamorphosis*, i.e., a QN where the data is modified to follow an approximately gaussian distribution [Chilès and Delfiner, 2012].

In spite of its popularity and success, QN suffers from a practical question: *how to choose the target distribution?* Various choices of target distribution have been popularised for different

reasons in different fields, such as the uniform distribution in histogram equalisation in order to increase the global contrast of images; the gaussian distribution in Gaussian anamorphosis in order to be able to apply statistical methods that work well for gaussian data; or the median of the empirical distribution of the samples in biology as an attempt to keep some information of the original values. Beyond such heuristics, we are not aware of any rigorous guiding principle that could justify these choices, and as mentioned by Bolstad et al. [2003], 'it seems unlikely that an agreed standard could be reached' for the choice of the target distribution, leaving this question largely open.

In this work we propose a general principle to answer this question, namely, *to optimise the target distribution for the task to be performed after normalisation*, and illustrate this principle when after normalisation a linear model is trained for a classification or regression task. Coupling prior normalisation with subsequent linear model estimation results in a new model, which we term *supervised quantile normalisation* (SUQUAN), where the optimal target distribution is the solution to an optimisation problem. We show that, equivalently, SUQUAN can be thought of as a particular linear model with rank constraint over the space of $p \times p$ matrices, where each sample $x \in \mathbb{R}^p$ is embedded as a permutation matrix defined by the order of its features. We propose three algorithms to approximate a solution under different prior assumptions on the target distribution. We illustrate the behaviour of SUQUAN on simulated data and on real images and biological data, where it outperforms the standard QN procedures.

3.2 Quantile normalisation (QN)

Let us first set up notations and present the standard QN procedure. We consider data $x_1, \ldots, x_n \in \mathbb{R}^p$ where each sample is a p-dimensional vector, such as an image represented by the intensities of p pixels or a biological sample represented by the expression of p genes. QN is a nonlinear transform $\Phi_f : \mathbb{R}^p \to \mathbb{R}^p$ indexed by a vector $f \in \mathbb{R}^p$ which we call the *target quantile*. In words, QN monotonically modifies the entries of any input vector x so that $\Phi_f(x)$ has the same distribution of entries as f, but ranked in the same order as the entries of x. When $f = (f_1, \ldots, f_p)^\top$ is a valid quantile its entries are sorted in increasing order ($f_1 \leq f_2 \leq \ldots \leq f_p$), so that the smallest entry of x becomes f_1 in $\Phi_f(x)$, the second smallest becomes f_2, and so on. Ties in the entries of x are arbitrarily broken, e.g., by considering x_i before x_j if $x_i = x_j$ and $i < j$.

QN can be formalised mathematically as follows. Given any $x \in \mathbb{R}^p$, we call Π_x the $p \times p$ binary permutation matrix defined by $(\Pi_x)_{ij} = 1$ if the i-th entry of x is ranked at the j-th position when all entries of x are sorted from the smallest to the largest. Then by construction, the QN normalisation can be simply written as:

$$\forall x \in \mathbb{R}^p\,, \quad \Phi_f(x) = \Pi_x f\,. \tag{3.1}$$

The following example illustrates these notations and the relation (3.1) for an arbitrary sample $x \in \mathbb{R}^4$ and an arbitrary target quantile $f \in \mathbb{R}^4$:

$$x = \begin{pmatrix} 4.5 \\ 1.2 \\ 10.1 \\ 8.9 \end{pmatrix}, \quad \Pi_x = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad f = \begin{pmatrix} 0 \\ 1 \\ 3 \\ 4 \end{pmatrix}, \quad \Phi_f(x) = \Pi_x f = \begin{pmatrix} 1 \\ 0 \\ 4 \\ 3 \end{pmatrix}.$$

3.3 Supervised quantile normalisation (SUQUAN)

The QN transform is defined for any arbitrary target quantile f by (3.1). After QN our n samples $x_1, \dots, x_n$ therefore become n vectors $\Pi_{x_1} f, \dots, \Pi_{x_n} f$, amenable for further analysis. We propose that instead of separating the tasks of choosing a "good" target quantile for QN on the one hand, and analysing the normalised data for some application on the other hand, we couple the two problems and optimise the target quantile in order to better solve the subsequent data analysis problem.

Let us now instantiate this general principle to the problem of estimating a linear model after QN normalisation; this is useful, for example, when one wants to build a prognostic model for cancer from gene expression data, or classify images based on their content. A linear model with weights $w \in \mathbb{R}^p$ and offset $b \in \mathbb{R}$ applied after quantile normalisation with target quantile $f \in \mathbb{R}^p$ takes the form

$$\forall x \in \mathbb{R}^p\,, \quad F_{w,b,f}(x) = w^\top \Phi_f(x) + b\,. \tag{3.2}$$

Given samples $x_1, \dots, x_n$, let us consider a standard procedure where the parameters (w, b) of the linear model are estimated by penalised empirical risk minimisation, i.e., solve an optimisation problem of the form

$$\min_{w,b} \frac{1}{n} \sum_{i=1}^{n} \ell_i \left(F_{w,b,f}(x_i)\right) + \lambda \Omega(w)\,, \tag{3.3}$$

where ℓ_i is a loss function for sample i, such as the squared loss $\ell_i(u) = (y_i - u)^2$ for a regression problem with response output $y_i \in \mathbb{R}$, or the logistic loss $\ell_i(u) = \log\left(1 + \exp(-y_i u)\right)$ for a binary classification problem with response output $y_i \in \{-1, 1\}$, Ω is a penalty function such as the ℓ_1 or ℓ_2 norm, and $\lambda \geq 0$ is a regularisation parameter. Note that we can rewrite the regularised problem (3.3) as a constrained optimisation problem:

$$\min_{(w,b) \in \mathcal{W} \times \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} \ell_i \left(F_{w,b,f}(x_i)\right) \quad \text{where} \quad \mathcal{W} = \{w \in \mathbb{R}^p \,:\, \Omega(w) \leq C\}\,. \tag{3.4}$$

Under mild assumptions, such as the convexity of the ℓ_i's and Ω being a norm, both formulations (3.3) and (3.4) are equivalent in the sense that for all $\lambda > 0$ there exist a choice of $C \geq 0$ such that (3.3) and (3.4) have the same solution.

Solving (3.3) or (3.4) is a standard problem in machine learning and statistical estimation, and can be done by a variety of algorithms depending on n, p, and the specific loss and penalty. Instead of just optimising in (w, b) for a fixed target quantile f, chosen independently and often arbitrarily, SUQUAN considers f as a parameter of the full process from the raw data to the final linear models, and optimises f jointly with (w, b). For example, the constrained formulation (3.4) becomes:

$$\min_{(w,b,f) \in \mathcal{W} \times \mathbb{R} \times \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell_i \left(F_{w,b,f}(x_i)\right)\,, \tag{3.5}$$

where $\mathcal{F} \subset \mathbb{R}^p$ is a set of candidate target quantiles. Note that the only difference between (3.4) and (3.5) is the fact that that f is optimised in (3.5) and not in (3.4); obviously this not only impacts the choice of f, but also the solution in (w, b) that is usually different between (3.4) and (3.5). Note also that since SUQUAN optimises the same objective function as (3.4) but over more parameters, the objective function is lower at the optimal solution for SUQUAN than at

the optimal solution of (3.4); this suggests that SUQUAN has more flexibility to fit the training data, but also more chance of overfitting, and therefore that it may require more regularisation to have good generalisation performance compared to (3.4).

Regarding the set of candidate target quantiles $\mathcal{F}$, one possibility is to simply constrain the Euclidean norm of f to ensure that the regularisation in w has an effect, and consider:

$$\mathcal{F}_0 = \left\{ f \in \mathbb{R}^p \,:\, \frac{1}{p}\sum_{i=1}^{p} f_i^2 \leq 1 \right\} .$$

A caveat with $\mathcal{F}_0$ is that the target quantile may not be non-decreasing. We therefore consider a second set of bounded non-decreasing candidate target quantiles:

$$\mathcal{F}_{\text{BND}} = \mathcal{F}_0 \cap \mathcal{I}_0 \,, \quad \text{where} \quad \mathcal{I}_0 = \{ f \in \mathbb{R}^p \,:\, f_1 \leq f_2 \leq \ldots \leq f_p \}$$

denotes the set of non-decreasing vectors. Further constraints regarding the structure of f may also be encoded in $\mathcal{F}$. For example, if we expect the target quantile to be smooth, we propose to consider the following set of non-decreasing and smooth functions [Sysoev and Burdakov, 2016]:

$$\mathcal{F}_{\text{SPAV}} = \left\{ f \in \mathcal{I}_0 \,:\, \sum_{j=1}^{p-1} (f_{j+1} - f_j)^2 \leq 1 \right\} .$$

Plugging any of $\mathcal{F}_0$, $\mathcal{F}_{\text{BND}}$ or $\mathcal{F}_{\text{SPAV}}$ into (3.5) leads to a SUQUAN formulation with different sets of candidates target quantiles. Note that the presence of the non-penalised intercept $b \in \mathbb{R}$ in (3.2) ensures that a solution f to (3.5) is defined up to a constant; we can therefore constrain without loss of generality f to be centered ($\sum_{i=1}^{p} f_i = 0$) in $\mathcal{F}_0$ and $\mathcal{F}_{\text{BND}}$, since it corresponds to the constant that minimises the Euclidean norm of f, as well as in $\mathcal{F}_{\text{SPAV}}$, since the smoothness constraint is invariant to the addition of a constant.

3.4 SUQUAN as a matrix regression problem

In order to derive practical algorithms and shed light on the underlying optimisation problems for the different SUQUAN formulations, it is useful to rewrite them as equivalent regression problems. For that purpose, let us now combine the definition of QN (3.1) and of SUQUAN (3.5) together. Plugging (3.1) into (3.2) and (3.2) into (3.5) , we easily get that the objective function of SUQUAN can be rewritten as:

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n} \ell_i \left(F_{w,b,f}(x_i) \right) &= \frac{1}{n}\sum_{i=1}^{n} \ell_i \left(w^\top \Pi_{x_i} f + b \right) \\ &= \frac{1}{n}\sum_{i=1}^{n} \ell_i \left(< w f^\top, \Pi_{x_i} >_F + b \right) , \end{aligned} \tag{3.6}$$

where $< A, B >_F = \text{Tr}(A^\top B) = \sum_{i,j=1}^{p} A_{ij} B_{ij}$ is the standard Frobenius inner product between matrices. This reformulation clarifies that SUQUAN can be interpreted as a particular linear regression model after embedding the inputs space onto the space of $p \times p$ matrices. Indeed, let $\Psi : \mathbb{R}^p \to \mathbb{R}^{p \times p}$ be the mapping defined by

$$\forall x \in \mathbb{R}^p \quad \Psi(x) = \Pi_x \,, \tag{3.7}$$

then plugging (3.6) and (3.7) into (3.5) we obtain the following expression for SUQUAN:

$$\min_{(M,b)\in\mathcal{M}\times\mathbb{R}} \frac{1}{n}\sum_{i=1}^{n} \ell_i\left(< M, \Psi(x_i) >_F + b\right) , \tag{3.8}$$

where

$$\mathcal{M} = \mathcal{W} \otimes \mathcal{F} = \left\{ w f^\top \,:\, w \in \mathcal{W}\,,\, f \in \mathcal{F} \right\} .$$

In other words, SUQUAN can be interpreted as a regression problem after embedding input vectors onto permutation matrices, with a rank-1 constraint on the weight matrix M and additional constraints on its left and right singular vectors corresponding respectively to the linear model $w \in \mathcal{W}$ and the target quantile $f \in \mathcal{F}$, up to a scaling factor.

This intriguing interpretation of target quantile optimisation as constrained matrix regression raises several comments.

- The mapping Ψ in (3.7) is the well-known *permutation representation* of the symmetric group S_p [Diaconis, 1988; Serres, 1977], where each vector $x \in \mathbb{R}^p$ is seen as a permutation $\pi_x \in S_p$ defined by the ranking of its entries. In particular, this representation is irreducible when restricted to the set $\Sigma = \{f \in \mathbb{R}^p \,:\, \sum_{i=1}^p f_i = 0\}$ [Serres, 1977, exercice 2.6], which implies that for any quantile $f \in \Sigma$ (in particular any f that solves (3.5)), the set of quantile normalised vectors $\{\Phi_f(x) \,:\, x \in \mathbb{R}^p\}$ spans the full subspace Σ.

- Besides the permutation representation, other embeddings of S_p onto $\mathbb{R}^{p\times p}$ exist and have been proposed in machine learning. For example, Jiao and Vert [2015] considered mapping $x \in \mathbb{R}^p$ to a $p \times p$ binary matrix with (i, j)-th entry equal to 1 whenever the i-th entry of x is smaller than the j-th entry, and showed how Frobenius-norm regularised linear models can be estimated efficiently thanks to the kernel trick because the inner product between two $p \times p$ matrices corresponding to two vector embeddings can be computed in $O(p \ln(p))$ with an efficient implementation of the Kendall τ statistics. It can be observed that the permutation representation Ψ used by SUQUAN is also trivially amenable to benefit from the kernel trick: to compute the inner product between $\Psi(x)$ and $\Psi(x')$ for two vectors x and x', one just needs to sort the entries of each vector independently, in $O(p \ln(p))$, and count in $O(p)$ how many entries are ranked at the same position. However, the permutation representation is extremely sparse (p non-zero values among $p(p-1)$ zeros) and only controlling the Frobenius norm of M (in order to benefit from the kernel trick) may not be sufficient to fight possible overfitting.

- $\mathcal{M}$ is not a convex set, and SUQUAN is therefore not a convex optimisation problem. A possible variant of SUQUAN would be to relax the rank constraint and replace it for example by a trace norm constraint, which is known to be a natural convex surrogate for the rank [Srebro and Shraibman, 2005].

3.5 Algorithms

The SUQUAN formulation (3.8) is a nonconvex optimisation problem since the set of rank-1 matrices $\mathcal{M}$ is not convex. To approximatively solve it, we now propose two strategies. The first one, SUQUAN-SVD, does not really attempt to solve (3.8) but instead to directly find a

Algorithm 3.1 SUQUAN-SVD

Input: $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^p \times \{-1, 1\}$
Output: $f \in \mathcal{F}_0$ target quantile

1: $M_{LDA} \leftarrow 0 \in \mathbb{R}^{p \times p}$
2: $n_{+1} \leftarrow |\{i : y_i = +1\}|$
3: $n_{-1} \leftarrow |\{i : y_i = -1\}|$
4: **for** $i = 1$ to n **do**
5: Compute Π_{x_i} (by sorting x_i)
6: $M_{LDA} \leftarrow M_{LDA} + \frac{y_i}{n_{y_i}} \Pi_{x_i}$
7: **end for**
8: $(\sigma, w, f) \leftarrow SVD(M_{LDA}, 1)$

good target quantile $f \in \mathcal{F}_0$ for binary classification problems. The second one aims to find an approximate solution to (3.8) by performing alternate optimisation in f and w, as the problem is biconvex.

3.5.1 SUQUAN-SVD

In the case where $\mathcal{F} = \mathcal{F}_0$, i.e., when we do not constrain f to be non-decreasing, and $\Omega(\beta) = ||\beta||^2$, then the set $\mathcal{M}$ of candidate matrices in (3.8) is exactly the set of rank-1 matrices. In that case, (3.8) amounts to finding a rank-1 matrix that approximatively solves a linear regression or classification problem. Let us consider the binary classification setting, when the training set is composed of pairs $(x_i, y_i)_{i=1,\ldots,n}$ with $y_i \in \{-1, +1\}$. In that case, a simple linear classifier (without rank constraint) is the one obtained by linear discriminant analysis with identity covariance: $M_{LDA} = \mu_+ - \mu_-$, where μ_+ and μ_- are respectively the means of the matrices Π_{x_i} for the positive and negative classes. Consequently, a good rank-1 candidate classifier is the closest rank-1 matrix to M_{LDA}, namely $u\sigma v^\top$ where u and v are the left and right singular vectors of M_{LDA} associated to the largest singular value σ. Hence we recover a target quantile function by keeping only the first right singular vector of M_{LDA}, which can then be used as target quantile for quantile normalising the training points before running any linear classification method. Algorithm 3.1 summarises the method. Computing Π_{x_i} on line 5 involves an $O(p \ln(p))$ sorting of the entries of x_i, and therefore computing M_{LDA}, which is a linear combination of n permutation matrices, requires $O(np \ln(p))$ operations. Then computing the right largest singular vector (line 8) of M_{LDA} typically costs another $O(p^2)$ operations using a naive power iteration method. However, if $n \leq p$, we can exploit the fact that the product of a permutation matrix by a vector is just an $O(p)$ operation (just order the vector according to the permutation), so that the power iteration to compute the first singular vector only takes $O(np)$. Computing the right largest singular vector therefore has an $O(\min(p^2, np))$ complexity. Hence the complexity of SUQUAN-SVD is $O(np \ln(p))$, which is the same as the complexity of the quantile normalisation.

3.5.2 SUQUAN-BND and SUQUAN-SPAV

We now focus on approximate algorithms to solve (3.8) in the case where $\mathcal{F} = \mathcal{F}_{BND}$ or $\mathcal{F} = \mathcal{F}_{SPAV}$. Using the biconvexity of (3.8) in w and f, we propose an alternate optimisation

Algorithm 3.2 SUQUAN-BND and SUQUAN-SPAV

Input: $(x_1, y_1), \ldots, (x_n, y_n), f_{init} \in \mathcal{I}_0, \lambda \in \mathbb{R}$
Output: $f \in \mathcal{I}_0$ target quantile

1: **for** $i = 1$ to n **do**
2: $\quad rank_i, order_i \leftarrow \text{sort}(x_i)$
3: **end for**
4: $w, b \leftarrow \underset{w,b}{\text{argmin}} \frac{1}{n} \sum_{i=1}^n \ell_i \left(w^\top f_{init}[rank_i] + b\right) + \lambda ||w||^2$
(standard linear model optimisation)
5: $f \leftarrow \underset{f \in \mathcal{F}_{BND}}{\text{argmin}} \frac{1}{n} \sum_{i=1}^n \ell_i \left(f^\top w[order_i] + b\right)$
(isotonic optimisation problem using PAVA as prox)
OR
$f \leftarrow \underset{f \in \mathcal{F}_{SPAV}}{\text{argmin}} \frac{1}{n} \sum_{i=1}^n \ell_i \left(f^\top w[order_i] + b\right)$
(smoothed isotonic optimisation problem using SPAV as prox)

scheme in w and f. Algorithm 3.2 summarises the procedure. Starting from an initial non-decreasing target quantile $f_{init} \in \mathcal{I}_0$, it outputs a new target quantile f obtained by minimising once (3.8) in w for $f = f_{init}$ fixed, then minimising in f for w fixed. Each alternative optimisation is particularly simple and efficient. For a given f, the optimisation in (w, b) amounts to solving a standard linear model optimisation over the samples $(\Pi_{x_1} f, \ldots, \Pi_{x_n} f)$. For a given w, the optimisation in f differs according to the regularisation type. With $\mathcal{F}_{\text{BND}}$, the optimisation in f is an isotonic optimisation problem (because of the constraints in $\mathcal{F}_{\text{BND}}$ that entries in f should be non-decreasing) involving the samples $\left(\Pi_{x_1}^\top w, \ldots, \Pi_{x_n}^\top w\right)$, which we solve by accelerated proximal gradient optimisation, borrowing the pool adjacent violators algorithm [Barlow et al., 1972, PAVA] as proximal operator to project onto the set of monotonically increasing vectors in $O(p)$. With $\mathcal{F}_{\text{SPAV}}$, this is a smoothed isotonic optimisation problem via ℓ_2 regularisation. Again, we solve this problem by accelerated proximal gradient optimisation but this time borrowing the Smoothed Pool Adjacent Violators [Sysoev and Burdakov, 2016, SPAV] as proximal operator which costs $O(p^2)$ operations; in this case we solve a penalised version (as opposed to a constrained version) of the problem, inducing a second regularisation parameter γ. Interestingly, the computation of each matrix-vector products $\Pi_{x_i} f$ and $\Pi_{x_i}^\top w$ before each alternative optimisation is just an $O(p)$ operation, after the sample x_i has been sorted once at the first iteration in $O(p \ln(p))$. Indeed, for a given x, if we note $order(x)$ the permutation which rearranges the entries of x in increasing order, and $rank(x)$ the ranks of the entries of x, then we simply have $(\Pi_x f)_j = f_{rank(x)_j}$ and $(\Pi_x^\top w)_j = w_{order(x)_j}$, for $j = 1, \ldots, p$, which we simply denote as $\Pi_x f = f[rank(x)]$ and $\Pi_x^\top w = w[order(x)]$ in Algorithm 3.2. Note that the procedure can be iterated to produce a sequence of target quantiles although we found in our experiments below that the performance did not change significantly after the first iteration. Note also that, contrary to SUQUAN-SVD, this algorithm requires an initial non-decreasing target quantile function. By default we suggest to use the median of the data quantile functions, which is often the default used in standard QN normalisation.

3.6 Experiments

3.6.1 Simulated data

We first test the ability of SUQUAN to overcome unwanted changes in quantile distributions on simulated datasets. For that purpose we fix $f \in \mathbb{R}^p$ to be the quantile distribution of the normal distribution, and simulate each sample $x_1, \ldots, x_n \in \mathbb{R}^p$ by randomly permuting the entries of f. We then generate binary labels $y_1, \ldots, y_n \in \{-1, 1\}$ using the logistic model $P(Y = 1 \,|\, X = x) = \frac{1}{1+\exp(-w^\top x)}$, where w is randomly sampled from a standard multivariate normal distribution. We then compare four methods to estimate w from n observations:

- Ridge logistic regression estimated on the correct data $(x_i, y_i)_{i=1,\ldots,n}$.
- Ridge logistic regression estimated on the corrupted data $(\Phi_g(x_i), y_i)_{i=1,\ldots,n}$, where g is a corrupted quantile distribution.
- SUQUAN-BND and SUQUAN-SPAV estimated on the corrupted data $(\Phi_g(x_i), y_i)_{i=1,\ldots,n}$.

While the true target f quantile is normal, we test four corrupted target quantiles g, derived from the cauchy, exponential, uniform and bimodal gaussian distributions. We assess the performance of the estimation by the area under the curve (AUC) on an independently generated test set of 1000 samples. The hyperparameters controlling the ℓ_2 penalty on w (λ) and the smoothness penalty on f (γ) for SUQUAN-SPAV were chosen thanks to an inner 5 times 3-fold cross-validation. The grid of values tested for λ ranges from 10^{-5} to 10^5 in log scale and from 10^0 to 10^4 in log scale for γ.

Figure 3.1 shows the performance of the different methods as a function of n, the number of training samples. In the case $n \ll p$, all methods perform almost equally badly in terms of AUC, including linear regressions on the true and on the corrupted datasets. However, SUQUAN-SPAV is able to learn a target quantile which is closer in terms of Euclidean distance to the true target quantile than the initial corrupted target quantile. When the number of samples increases while the number of features is kept fixed, the performances of both SUQUAN-BND and SUQUAN-SPAV clearly outperforms that of linear regression on the corrupted dataset. In particular, the AUC curves show that SUQUAN-SPAV is almost as good as linear regression performed on the true dataset whatever the dimensionality is. Moreover, both SUQUAN-BND and SUQUAN-SPAV improve their estimates of the true target quantile when the number of samples increases. Overall, these results confirm that SUQUAN can improve the performances of a linear model by recovering a good estimate of the true target quantile function, and illustrate the detrimental impact of a bad choice for the target quantile function.

3.6.2 CIFAR-10 dataset

We next test SUQUAN on an image classification task. Since our objective is to study the impact of QN with different target quantile functions, we do not aim to reach state-of-the-art classification results with complex features extracted from images, but instead assess the performance of simple linear models on pixel intensities. Here changing the target quantile can be thought of as a variant of the histrogram matching procedure. We consider the CIFAR-10 benchmark dataset [Krizhevsky, 2009] which consists of 32×32 tiny colour images from 10

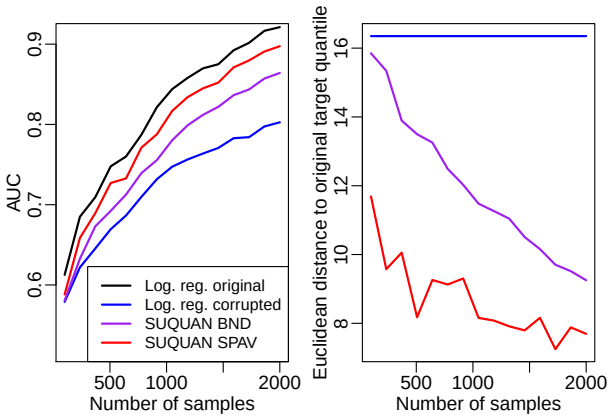


Figure 3.1 – Performance on simulated data. The number of features is fixed to p = 1000 while the number of training samples varies from 100 to 2000, and the results are averaged over four experiments with different corrupted quantile functions. The left plot shows the test AUC for logistic regressions applied to the original (black) and corrupted (blue) data as well as SUQUAN-BND (purple) and SUQUAN-SPAV (red). The right plot shows the Euclidean distance between the original target quantile and the target quantile used to corrupt the data (blue), the target quantile learned with SUQUAN BND (purple), and the target quantile learned with SUQUAN-SPAV (red).

different classes. The dataset is divided into 50,000 training images (5,000 of each class) and 10,000 test images (1,000 of each class). To simplify the setting, we consider independently all 45 binary classification problems derived from the 10 classes. For each of these 45 problems, images were first converted to grayscale and represented as vectors of grey intensities. Therefore for each binary problem we have 10,000 training samples, 2,000 test samples, and 1,024 features per image.

We compare SUQUAN on these 45 classification tasks to a logistic regression model for which data has been quantile normalised beforehand with various target quantiles. Among these target quantiles we test the median of the empirical distribution of the samples, the target quantile derived from the uniform distribution which amounts to performing histogram matching, as well as the target quantiles derived from the cauchy, exponential and gaussian distribution in order to have diversity in the target quantiles chosen. SUQUAN as well as the logistic regression are fitted with an ℓ_2 penalty on the weights w. Hyperparameters are selected using a 5 times 3 fold cross-validation on the train set. The grid of values tested for λ ranges from 10^{-5} to 10^5 in log scale and from 10^0 to 10^4 in log scale for γ.

The distributions of test AUC obtained for each method across all 45 classification problems are shown in Fig. 3.2a. SUQUAN-BND yields the best average performance and outperforms all logistic regression models learned with fixed target quantiles. Moreover, if we compare the performances of SUQUAN-BND to that of the logistic regression with the median as target quantile (Fig. 3.2b), we see that the improvements yielded by SUQUAN-BND are consistent across datasets. These observations therefore confirm the benefit of optimising the target quantile at the same time as the model weights, and support the idea that fixing a pre-defined target quantile can hurt the performance of a linear model. Interestingly, the simplified version of SUQUAN, i.e., SUQUAN-SVD, also creates a target quantile function which outperforms all other fixed target quantiles. In order to illustrate what the learned target quantiles look like

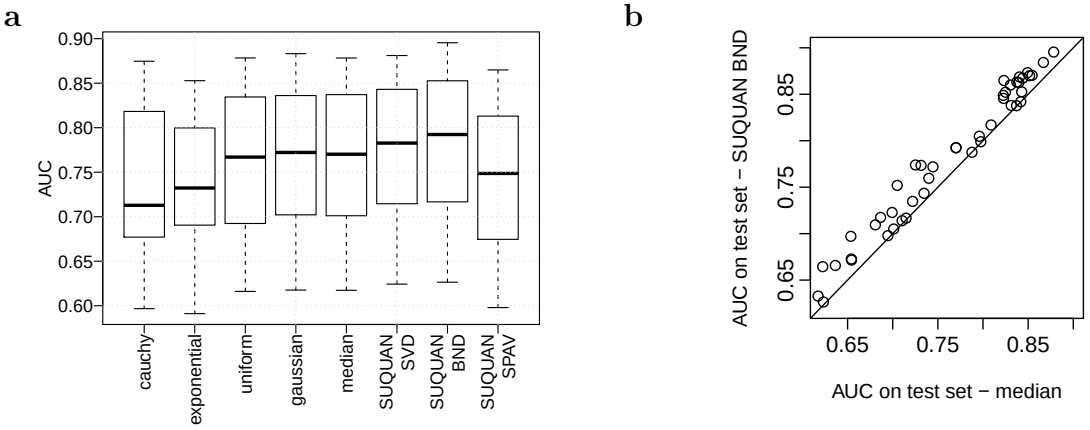


Figure 3.2 – **Performance on CIFAR-10.** (a) Each box-plot summarises the test AUC of a method on the 45 binary classification tasks. For the first seven boxplots on the left, the data was first normalised using a target quantile either drawn from a distribution or estimated by SUQUAN-SVD, and a logistic regression was fitted to the normalised data. The last two cases correspond to directly applying SUQUAN-BND or SUQUAN-SPAV to the data. (b) Comparison of the test AUC obtained with a logistic regression on data previously quantile normalised with the median on the one hand, and SUQUAN-BND on the original data on the other hand. Each point corresponds to one of the 45 binary classification tasks from CIFAR-10.

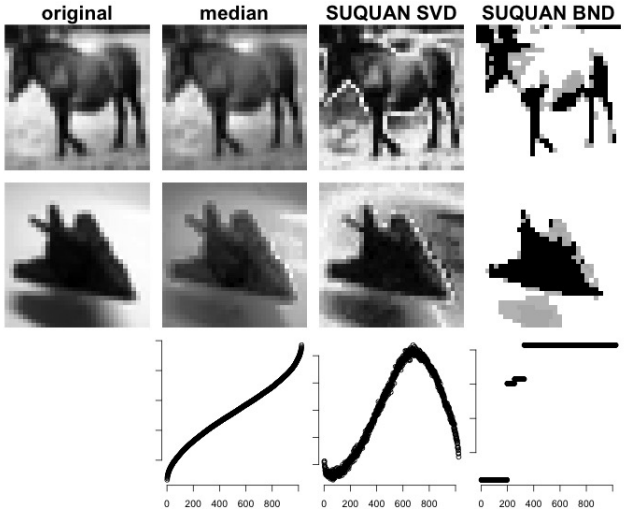


Figure 3.3 – **Target quantiles for the "airplane" versus "horse" binary classification task.** The first (resp. second) row represents one sample image from the "horse" (resp. "airplane") class in its original form or normalised with the median target quantile across all images , the target quantile from SUQUAN-SVD, or the one learned with SUQUAN BND. The third row shows the shape of the target quantiles f in each case.

Dataset name	# patients	# positives	% positives
GSE1456	141	37	0.26
GSE2034	271	104	0.38
GSE2990	106	32	0.30
GSE4922	225	73	0.32

Table 3.1 – Gene expression datasets used in this study. The dataset name corresponds to the accession number in the GEO database.

for both SUQUAN-BND and SUQUAN-SVD, we show in Fig. 3.3 the normalised images from the 'horse' versus 'airplane' classification task according to the different methods. We note two things: first, as the target quantile learned with SUQUAN-SVD can be non-monotonic, black pixels in the original image can become white in the normalised image and conversely. Interestingly here this inversion tends to occur at the edges of the objects and therefore plays a role which mimics a simple edge detector; second, SUQUAN-BND learns a target quantile with only few steps, and therefore tends to 'binarise' the image, which probably brings out salient features. Finally, we also observe that SUQUAN-SPAV has bad performances on these 45 binary classification tasks, suggesting that the smoothness constraint on the target quantile is detrimental in this case. We hypothesise this may be due to the inherent structure of images, and also to the fact that in a $n \gg p$ setting, constraining the model too much is not necessary.

3.6.3 Gene expression data

Genomic data are often subject to many types of unwanted variations that corrupt the recorded data, including but not restricted to sample preparation protocols, temperature, or measurement tools. To test the relevance of SUQUAN in this context, we focus on the problem of breast cancer prognosis from gene expression data, and collected 4 publicly available datasets describing gene expression profiles in human breast cancer tumours together with survival information from the GEO database [Barrett et al., 2011]. For each of these 4 datasets we retrieved the raw data (CEL files) which we summarised (using the median polish procedure) to obtain gene expressions. Each dataset contains the expression level of 22,283 genes measured using the same microarray technology and the number of breast cancer patients (or samples) ranges from 106 to 271 patients (Table 3.1). In each dataset, we split the patients into two classes: those who relapsed within 6 years of diagnosis and those who did not. The precise description of the datasets is summarised in Table 3.1. The problem is therefore to predict the class of a patient (relapse or not) given its gene expression values, which is a binary classification task.

We again compare the performances of SUQUAN to that of logistic regression on previously quantile normalised data with various target quantiles namely cauchy, exponential, uniform, gaussian and median. We also fit logistic regressions on the raw data and on the data preprocessed with Robust Multi-Array Average [Irizarry et al., 2003, RMA]. RMA is a widely used preprocessing method for gene expression microarrays which notably includes a background correction step and a quantile normalisation step with the median as target quantile. Experiments are performed in a 5-times 3-fold external cross-validation setting and the performances reported are the average over these 15 folds. Both models (SUQUAN and logistic regression) are fitted with an ℓ_2 norm penalty on w. Hyperparameters are optimised by 5-times 3-fold inner cross-validation. The grid of values tested for λ ranged from 10^{-5} to 10^{1} in log scale and from 10^{0} to

	LOGISTIC REGRESSION							SUQUAN		
	RAW	RMA	CAUCHY	EXP.	UNIF.	GAUS.	MEDIAN	SVD	BND	SPAV
GSE1456	65.94	68.73	59.56	68.86	68.72	69.00	69.06	57.60	71.44	69.60
GSE2034	74.52	75.42	61.91	74.53	75.22	76.45	74.92	52.61	70.50	76.11
GSE2990	57.01	60.43	54.72	61.25	56.25	58.66	59.72	52.51	59.22	59.94
GSE4922	58.52	58.86	55.24	58.81	55.66	60.01	59.18	52.39	61.82	61.41
AVERAGE	64.00	65.86	57.86	65.86	63.96	66.03	65.72	53.78	65.75	**66.77**

Table 3.2 – AUC for SUQUAN and logistic regression with various data normalisation procedures applied to four gene expression datasets.

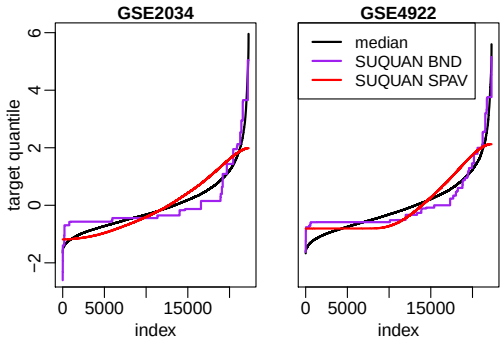


Figure 3.4 – Example of target quantiles learned for two gene expression datasets and an arbitrary split in train/test sets.

10^4 in log scale for γ.

Table 3.2 summarises the performance of each method on each dataset. Looking at the mean performance across the four datasets, we observe that the performance of the logistic regression varies according to the target quantile used, which underlines the fact that the choice of the target quantile is important. In particular, RMA (65.86) is one of the top performing preprocessing methods along with quantile normalisation with the median (65.72), exponential (65.86) and gaussian (66.03) target quantiles. Moreover, SUQUAN-SPAV (66.77) outperforms all other methods on average. This increase in performance is significant according to a one-sided paired Wilcoxon signed rank statistical test (P-value $\leq 5\times10^{-2}$) for all logistic regressions except those fitted with RMA and the gaussian as target quantile for which the P-values $P = 6.9\times10^{-2}$ and $P = 5.9 \times 10^{-2}$ are just above the significance threshold of 5%. We would like to mention that for cancer prognosis from gene expression data, it is very unlikely that any method will ever outperform the baseline by more than a few percents. Illustrations of typical target quantiles learned with SUQUAN-SPAV are shown on Fig. 3.4. Interestingly, while in the large n small p configuration (i.e on CIFAR) SUQUAN-BND was the best method, here in a small n large p configuration SUQUAN-SPAV is better on average than SUQUAN-BND. This may be due to the fact that the smoothness constraint on f which is implemented in SUQUAN-SPAV is a useful additional regularisation to prevent overfitting. Finally, we observe that SUQUAN-SVD is by far the worst method on these gene expression datasets, probably due to numerical instabilities when computing the singular vectors of large sparse matrices that appear in the $n \ll p$ setting.

3.7 Discussion

QN is an ubiquitous normalization method used throughout several application fields to remove unwanted variations in the recorded data before performing any analysis. However, the choice of the target quantile function is most often empirical and driven by field-specific standard choices. We presented a model, SUQUAN, that allows to learn the optimal target quantile function while performing a given task such as classification or regression. We showed that SUQUAN can be interpreted as a constrained matrix regression problem where sample vectors are embedded as permutation matrices.

The idea of optimising the target quantile function jointly with other parameters lends itself well to further investigations. For example, by changing the objective function of SUQUAN, one may consider other applications such as optimising the quantile function in order to improve clustering or visualisation of the data after QN, or the signal-to-noise ratio to detect differentially expressed genes. Regarding SUQUAN itself, a better understanding of the statistical properties of learning low rank linear models on the permutation representation of the symmetric group, as well as extensions from rank-1 to low rank matrices, are interesting future work.

Another remaining challenge is to develop non linear extensions of SUQUAN, using for example kernels. Such an extension is not straightforward since the optimisation is not regularised by an L2 norm of the linear model, which would be needed for a simple ‘ kernel trick ’ extension. Instead it is regularised by a rank constraint on the model, which we empirically observed to be crucial. Another way to think about ‘ kernelising’ the model would be to replace the representation of a permutation as the permutation matrix by something else (i.e., keep the model linear but in another representation), which leads to the question of defining more general kernel or feature representation for the symmetric group, a topic of broader interest in machine learning.

Chapter 4

WHInter: A Working set algorithm for High-dimensional sparse second order Interaction models.

Contents

4.1 Introduction . . . 77
4.2 Preliminaries . . . 79
4.2.1 Setting and notations . . . 79
4.2.2 Basic working set algorithm . . . 80
4.3 The WHInter algorithm . . . 81
4.3.1 Overview . . . 81
4.3.2 The Branch bound η . . . 81
4.3.3 Updating the working set . . . 84
4.4 Simulation study . . . 86
4.5 Results on real world data . . . 87
4.6 Related work . . . 89
4.6.1 Safe pattern pruning . . . 89
4.6.2 Prioritisation of updates in coordinate descent . . . 90
4.7 Discussion . . . 91

Abstract

Learning sparse linear models with two-way interactions is desirable in many application domains such as genomics. ℓ_1-regularised linear models are popular to estimate sparse models, yet standard implementations fail to address specifically the quadratic explosion of candidate two-way interactions in high dimensions, and typically do not scale to genetic data with hundreds of thousands of features. Here we present WHInter, a working set algorithm to solve large ℓ_1-regularised problems with two-way interactions for binary design matrices. The novelty of WHInter stems from a new bound to efficiently identify working sets while avoiding to scan all features, and on fast computations inspired from solutions to the maximum inner product search problem. We apply WHInter to simulated and real genetic data and show that it is more scalable and two orders of magnitude faster than the state of the art.

Résumé

Pouvoir apprendre des modèles linéaires parcimonieux avec des interactions de variables deux à deux est désirable pour de nombreux domaines d'applications comme par exemple pour la génomique. Les modèles linéaires régularisés avec une norme ℓ_1 sont répandus pour l'estimation de modèles parcimonieux, toutefois les implémentations standard ne parviennent pas à gérer l'explosion quadratique du nombre d'interactions candidates en grande dimension, et sont typiquement inapplicables à des données génétiques qui contiennent plusieurs centaines de milliers de variables. Nous présentons ici WHInter, un algorithme de type 'working set' qui permet de résoudre des problèmes régularisés ℓ_1 comportant de nombreux termes d'interactions, pour des matrices de design binaires. Le caractère innovant de WHInter découle de la dérivation d'une nouvelle borne qui permet d'identifier efficacement les 'working sets' tout en évitant de parcourir toutes les variables, ainsi que de calculs rapides inspirés des solutions du problème de recherche du produit scalaire maximal. Nous appliquons WHInter à des données simulées et à des données génétiques réelles, et montrons que WHInter gère mieux l'augmentation du nombre de variables et est jusqu'à deux ordres de grandeurs plus rapide que l'état de l'art.

4.1 Introduction

In application domains where the number of features exceeds the number of available samples, sparsity-inducing regularisers have a long history of success. Genomic prediction of complex phenotypes, biomedical imaging, astronomy or finance are a few examples. In particular the least squares with ℓ_1 regularisation, known as the LASSO [Tibshirani, 1996], has been extensively studied. It enjoys desirable statistical properties, since the number of samples required for exact support recovery of a sparse model scales as the logarithm of the number of features, under some assumptions [Wainwright, 2009]. It also enjoys practical advantages, notably the interpretability of the learned models and the availability of fast solvers.

Indeed, a lot of research effort has been devoted to accelerating solvers for sparsity constrained problems in high dimension. A central idea is to exploit the sparsity of the solution to develop algorithms that do not spend too much time on optimising coefficients that will end up being 0. For example, safe screening rules identify features which are guaranteed to be inactive at the optimum so that their corresponding coefficients can be safely zeroed and set aside from the pool of coefficients to update [El Ghaoui et al., 2012; Fercoq et al., 2015; Raj et al., 2016; Wang et al., 2013; Xiang et al., 2011; Xiang and Ramadge, 2012]. Dynamic screening rules [Bonnefoy et al., 2015] such as the GAP safe rules [Fercoq et al., 2015] are particularly useful since more and more coefficients can be safely zeroed while the solver approaches the optimal solution. In spite of this, safe rules tend to be conservative, thereby limiting the potential speed-up. To remedy this drawback, new working set heuristics have been proposed. Working set algorithms iteratively solve subproblems, either problems restricted to a subset of features in the primal or to a subset of constraints in the dual, until convergence. Working set methods allow to focus coefficient updates on a set of features which can be significantly smaller than that yielded by safe rules. However this comes at a cost, that of checking the optimality conditions for all features at each iteration. BLITZ [Johnson and Guestrin, 2015] is a recently proposed working set algorithm that has been shown to have state-of-the-art performance for ℓ_1-regularised problems. Interestingly, the choice of the working sets in BLITZ can be seen as an aggressive use of the GAP safe rules [as noted in Massias et al., 2017] where the size of the working set is chosen to maximise the progress towards convergence. BLITZ can therefore be combined with the GAP safe rules (or the FLEX constraint elimination according to Johnson et al. terminology) at no cost. A direct comparison between BLITZ and the GAP safe rules by Ndiaye et al. [2017] illustrates the effectiveness of the working set approach. Further developments have also focused on coordinate descent (CD) to avoid wasteful coordinate updates, which represent most of the time spent by the solver [Fujiwara et al., 2016; Johnson and Guestrin, 2017].

The problem of fitting sparse linear models with two-way interactions has also attracted attention during the past decade. By two-way interactions we mean the entry-wise multiplication between two features; this is for example important in genomics to detect possible epistasis between genes. Surprisingly, very few of these works have links with the aforementioned literature. A majority of them focus on the design of sparsity-inducing penalties which enforce heredity assumptions and apply to moderate-dimensional settings ($p < 1,000$) [Bien et al., 2013; Haris et al., 2016; Lim and Hastie, 2015; Radchenko and James, 2010]. Heredity assumptions state that an interaction can be included in the model only if one or both of its corresponding main effects are included. We note however that `glinternet` [Lim and Hastie, 2015] was applied to higher dimensional problems and in particular to a dataset with roughly $p = 27,000$

main effects, although the size of the learned model is not specified and the running time for the experiment is not reported by the authors. Interestingly, `glinternet` uses an active set strategy. Comparatively few works have been devoted to learning sparse regression models with interactions when the number of interactions is higher. Most of them are heuristics which start by selecting main effects and then incorporate interactions generated under the heredity constraint in a possibly iterative fashion. The simplest form of such heuristics consists in fitting a sparse linear model with the main effects only, and then fitting a second sparse linear model on all previously selected main effects and their interactions. This has been used in practice for example by Wu et al. [2009]. Iterative refinements have been proposed where the LASSO is fit several times, and each time the set of candidate interactions considered is updated either by subsets, with the interactions between the K most relevant main effects selected at the previous fit [Bickel et al., 2010], or in a greedy fashion, where new interactions are included in the model as soon as a new main effect enters the LASSO path [Shah, 2016]. In a similar vein, Hao and Zhang [2014] is based on a greedy model selection procedure instead of several LASSO fits. While these heuristics can deal with higher-dimensional problems than previous methods and enjoy some desirable statistical properties, they do not provide exact solutions and do not enjoy statistical properties as strong as those of the LASSO estimator.

An interesting link between the literature on interactions and that of solver acceleration with sparsity inducing norms has been made recently by Nakagawa et al. [2016]. In the case where variables are binary or with values in $[0, 1]$, they propose an approach called Safe Pattern Pruning (SPP) which is able to provide the optimal solution of the LASSO with two-way interactions for fairly high-dimensional problems, with no heredity constraint. Typically, for a problem with 1,000 samples and 10,000 main effects, SPP can provide solutions for a grid of regularisation parameters within one or two hours on a laptop with one core. SPP relies on the recently developed GAP safe screening rules. More precisely, the authors propose a safe pattern pruning criterion that can safely discard subsets of interactions from the model to speed up convergence. The performance of SPP is however hindered by several factors. One of them is that safe screening rules can be quite conservative even in the sequential setting. This property is inherited and amplified by the SPP criterion which can lead to heavy computations. Moreover, the GAP safe rules rely on a dual feasible point which is expensive to compute especially when the number of interactions is huge.

Inspired by SPP and the acceleration of solvers for sparsity constrained problems we propose a scalable algorithm, WHInter, to compute the optimal solution of ℓ_1-regularised linear problems with two-way interactions. WHInter is a working set method that efficiently delineates working sets among all interactions and main effects thanks to two contributions. First, we introduce a cheap and effective bound to rule out subsets of interactions that are guaranteed to be outside of the working set. Second, the identification of the working set among the remaining features is cast as a variant of the Maximum Inner Product Search (MIPS) problem to alleviate the afferent computational load. We find that WHInter is up to two orders of magnitude faster than SPP. For example, a problem with roughly 700 samples and 100,000 main effects can be solved for a grid of regularisation parameters in half an hour on a laptop with one core compared to more than 30 hours with SPP. This improvement in the scalability opens up new horizons in several application fields. The rest of the chapter is organised as follows. In section 2, we present useful knowledge and notations used throughout the paper. In section 3 we describe in details our algorithm and our main contributions. In section 4, we evaluate WHInter on simulated datasets

and finally in Section 5, we report results on a toxicogenomics prediction task.

4.2 Preliminaries

4.2.1 Setting and notations

For any integer $d \in \mathbb{N}$, we note $[\![d]\!] = \{1, \dots, d\}$ and $\mathbf{1}_d \in \mathbb{R}^d$ the d-dimensional vector of 1's. For any vector $\mathbf{u} = (\mathbf{u}_1, \dots, \mathbf{u}_d) \in \mathbb{R}^d$, we note $\| \mathbf{u} \|_1 = \sum_{i=1}^d | \mathbf{u}_i |$, $\| \mathbf{u} \|_2 = \left(\sum_{i=1}^d \mathbf{u}_i^2 \right)^{1/2}$, $\mathrm{supp}(\mathbf{u}) = \{i \in [\![d]\!] : \mathbf{u}_i \neq 0\}$ and $\| \mathbf{u} \|_0 = | \mathrm{supp}(\mathbf{u}) |$. For any two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, $\mathbf{u} \odot \mathbf{v}$ is the vector of entry-wise products, i.e., $(\mathbf{u} \odot \mathbf{v})_i := \mathbf{u}_i \mathbf{v}_i$ for $i = 1, \dots, d$. For any matrix $\mathbf{M}$, we denote by $\mathbf{M}_{i,j}$ its (i,j)-th entry, $\mathbf{M}_j$ its j-th column and by $\boldsymbol{m}_i$ its i-th row. For any $\mathbf{u} \in \mathbb{R}^d$ and $\mathcal{I} \subset [\![d]\!]$, $\mathbf{u}_\mathcal{I} = (\mathbf{u}_i)_{i \in \mathcal{I}}$, and similarly, if $\mathbf{M}$ is a matrix with d columns, $\mathbf{M}_\mathcal{I}$ is the sub-matrix with $|\mathcal{I}|$ columns $\mathbf{M}_\mathcal{I} = (\mathbf{M}_i)_{i \in \mathcal{I}}$.

Throughout the text we consider a design matrix $\mathbf{X} \in \{0,1\}^{n \times p}$ corresponding to n samples and p binary features, together with a response vector $\mathbf{y} \in \mathbb{R}^n$. We define an expanded design matrix $\mathbf{Z} \in \{0,1\}^{n \times D}$, with $D = p(p+1)/2$, which contains all p features from $\mathbf{X}$ plus the $p(p-1)/2$ interaction features. For clarity purposes, we define a symmetric indexing function $\tau : [\![p]\!]^2 \mapsto [\![D]\!]$ that uniquely assigns to every main effect and interaction an index in the expanded matrix $\mathbf{Z}$ such that $\mathbf{Z}_{\tau(j,k)} = \mathbf{Z}_{\tau(k,j)} := \mathbf{X}_j \odot \mathbf{X}_k$. In particular $\mathbf{Z}_{\tau(i,i)} = \mathbf{X}_i \odot \mathbf{X}_i = \mathbf{X}_i$ represents the i^{th} main effect. Since $\mathbf{X}$ is a binary matrix, the interaction feature $\mathbf{X}_j \odot \mathbf{X}_k$ corresponds to a logical AND between features $\mathbf{X}_i$ and $\mathbf{X}_j$. We organise the main effects and interactions in a simple tree as depicted in Fig. 4.1 so as to reflect the property that $\forall (j,k) \in [\![p]\!]^2, \mathbf{Z}_{\tau(j,k)} \leq \mathbf{X}_j$ and $\mathbf{Z}_{\tau(j,k)} \leq \mathbf{X}_k$. In the sequel, the set composed of a main effect and its interactions with all other main effects will be referred to as *a branch* and for any $j \in [\![p]\!]$, we note $\mathrm{branch}(j) = \{\tau(j,k) : k \in [\![p]\!]\}$.

We consider the convex optimisation problem:

$$\min_{(\mathbf{w},b) \in \mathbb{R}^D \times \mathbb{R}} P_{\mathbf{Z},\lambda}(\mathbf{w}, b) := F\left(\mathbf{Z}\mathbf{w} + b\mathbf{1}_n\right) + \lambda \|\mathbf{w}\|_1 := \sum_{i=1}^n f_i\left(\mathbf{z}_i \mathbf{w} + b\right) + \lambda \|\mathbf{w}\|_1 , \tag{4.1}$$

where $\lambda > 0$ is a regularisation parameter and, for any $i \in [\![n]\!]$, $f_i : \mathbb{R} \mapsto [-\infty, +\infty]$ is a loss function parametrised by $\mathbf{y}_i$ and assumed to be convex and differentiable. Table 4.1 provides

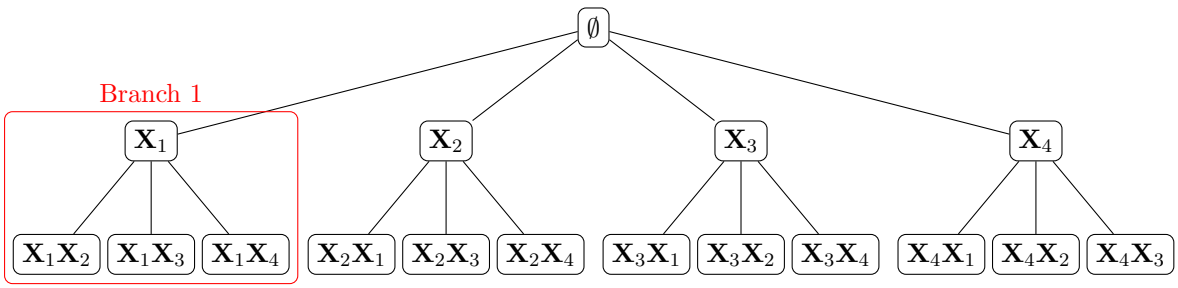


Figure 4.1 – Organisation of the main effects and interactions in a tree, depicted for 4 main effects.

examples of classical loss functions in classification and regression. A dual formulation of (4.1) reads:

$$\max_{\boldsymbol{\theta} \in \mathbb{R}^n} D_{\mathbf{Z},\lambda}(\boldsymbol{\theta}) := -\sum_{i=1}^{n} f_i^*\left(-\boldsymbol{\theta}_i\right) \quad \text{s.t.} \quad \begin{cases} \left|\mathbf{Z}_i^\top \boldsymbol{\theta}\right| \leq \lambda & \forall i \in [\![D]\!]\,, \\ \mathbf{1}_n^\top \boldsymbol{\theta} = 0\,, \end{cases} \tag{4.2}$$

where f_i^* is the Fenchel-Legendre transform of the loss f_i, i.e., the function $f_i^* : \mathbb{R} \mapsto [-\infty, +\infty]$ defined by $f_i^*(u) = \sup_{v \in \mathbb{R}} uv - f_i(v)$. For the derivation of the dual problem, we refer the reader to Johnson and Guestrin [2015, Appendix E]. The constraint $\mathbf{1}_n^\top \boldsymbol{\theta} = 0$ comes from the bias term $b\mathbf{1}_n$ in the primal problem (4.1). We denote by $(\mathbf{w}^*, b^*)$ and $\boldsymbol{\theta}^*$ a set of primal and dual optimal solutions to problems (4.1) and (4.2) respectively. Strong duality holds and therefore $(\mathbf{w}^*, b^*)$ and $\boldsymbol{\theta}^*$ satisfy Fermat's rules [Ndiaye et al., 2017]:

$$\boldsymbol{\theta}^* = -\nabla F(\mathbf{Z}\mathbf{w}^* + b^* \mathbf{1}_n)\,, \tag{4.3}$$

and

$$\forall i \in [\![D]\!]\,, \quad \mathbf{Z}_i^\top \boldsymbol{\theta}^* \in \begin{cases} \{-\lambda, \lambda\} & \text{if } \mathbf{w}_i^* \neq 0\,, \\ [-\lambda, \lambda] & \text{if } \mathbf{w}_i^* = 0\,. \end{cases} \tag{4.4}$$

	$f_i(u)$	$f_i'(u)$	$f_i^*(u)$
LASSO	$\frac{1}{2}(\mathbf{y}_i - u)^2$	$u - \mathbf{y}_i$	$\frac{1}{2}(\mathbf{y}_i + u)^2 - \frac{1}{2}\mathbf{y}_i^2$
Logistic regr.	$\log(1 + \exp(-\mathbf{y}_i u))$	$-\frac{u}{\mathbf{y}_i}\log(-\frac{u}{\mathbf{y}_i}) + (1 + \frac{u}{\mathbf{y}_i})\log(1 + \frac{u}{\mathbf{y}_i})$	$\frac{-\mathbf{y}_i}{1+\exp(\mathbf{y}_i u)}$

Table 4.1 – Summary of useful functions for the LASSO and logistic regression: loss function f_i, its derivative f_i', its Fenchel-Legendre transform f_i^*.

4.2.2 Basic working set algorithm

A general strategy to solve (4.1) is to follow a *working set* approach, as summarised in Algorithm 4.1. At each iteration, it solves (4.1) restricted to a small subset of features $\mathcal{W}$ called the working set. $\mathcal{W}$ is typically chosen as the set of features that violate the optimality condition (4.4) at the current iteration. In the sequel, we will call such features *violating features*. The algorithm converges when no violating feature remains, which occurs in a finite number of iterations as shown in Kowalski et al. [2011]. When the number of interaction features runs into the billions, Algorithm 4.1 is not tractable since the delineation of the working set (line 3 in Algorithm 4.1) requires $O(p^2 n)$ operations at each iteration.

Algorithm 4.1 Working set algorithm

Input: $\mathbf{Z} \in \{0,1\}^{n \times p}, \mathbf{y} \in \mathbb{R}^n$, $\lambda > 0$
Output: $\mathbf{w}^*, b^*$
1: Set $\boldsymbol{\theta} \leftarrow -\nabla F(\mathbf{0}_n)$, $\mathcal{W} = \emptyset$. ▷ Initialisation
2: **while** true **do**
3: $\quad \mathcal{W}' = \left\{ i \in [\![D]\!] : \left| \mathbf{Z}_i^\top \boldsymbol{\theta} \right| \geq \lambda \right\}$ ▷ Update the working set
4: $\quad$ **if** $\max_{i \in \mathcal{W}'} \left| \mathbf{Z}_i^\top \boldsymbol{\theta} \right| \leq \lambda$ **then** Break **else** $\mathcal{W} \leftarrow \mathcal{W}'$
5: $\quad \mathbf{w}^*_{\mathcal{W}}, b^* \leftarrow \underset{\mathbf{w}_{\mathcal{W}}, b}{\operatorname{argmin}} P_{\mathbf{Z}_{\mathcal{W}}, \lambda}(\mathbf{w}_{\mathcal{W}}, b)$ ▷ Solve subproblem
6: $\quad \boldsymbol{\theta} \leftarrow -\nabla F(\mathbf{Z}_{\mathcal{W}} \mathbf{w}^*_{\mathcal{W}} + b^* \mathbf{1}_n)$.
7: **end while**

4.3 The WHInter algorithm

4.3.1 Overview

WHInter is a working set algorithm that follows the general scheme of Algorithm 4.1 but implements an efficient strategy to delineate the working set among all main effects and interactions. It is described in Algorithm 4.2. The identification of the working set (line 3 in Algorithm 4.1) corresponds to lines 11-18 in Algorithm 4.2. Instead of scanning through all features to build the working set, WHInter first identifies branches that are guaranteed to contain no violating feature. These branches are identified via the evaluation of a *branch bound* $\eta(\mathbf{X}_j, \boldsymbol{\Theta}_j^{ref}, \boldsymbol{\theta}, \mathbf{m}_j^{ref})$ (line 13) which is described in Section 4.3.2. The branch bound is cheap to evaluate since it solely depends on main effects and not on their numerous interactions. Moreover, it is designed to efficiently rule out branches thanks to the exploitation of the shared structure among features in a branch, as well as the correlation among dual variables for two sufficiently close points in the optimisation path. In cases where a branch cannot be ruled out, features in the branch are considered one by one to build the working set, which is very computationally expensive. In order to reduce this cost, we cast the problem as a variant of the Maximum Inner Product Search (MIPS) problem, which is described in Section 4.3.3. If no violating feature is identified then the algorithm has converged. Otherwise, a new candidate solution is obtained by solving problem (4.1) restricted to the features in the working set, and the process is repeated until no violating feature remains. While any solver can be used to solve the restricted problem, we implemented in WHInter a coordinate descent approach with safe pruning.

4.3.2 The Branch bound η

As WHInter iterates, it produces candidate solutions $(\mathbf{w}^*, b^*)$ and corresponding dual variables $\boldsymbol{\theta}$ (lines 20 and 21 of Algorithm 4.2). For two sufficiently close iterations, or for two problems with sufficiently close regularisation parameters, the candidate solutions are likely to be *close* to one another, as well as the corresponding dual variables provided the function F does not vary too quickly. WHInter exploits this intuition to speed up the identification of the working set from an iteration to another or from one problem to another. The following results relate, for two distinct dual variables, the criteria used to identify the working set (line 3 of Algorithm 4.1).

Algorithm 4.2 WHInter

Input: $\mathbf{X} \in \{0,1\}^{n \times p}$, $\mathbf{y} \in \mathbb{R}^n$, $\lambda_1 > \cdots > \lambda_T$.
Output: $(\mathcal{W}, \mathbf{w}^*_{\mathcal{W}}, b^*)_t$ for each λ_t

Initialisation
$\boldsymbol{\theta} \leftarrow -\nabla F(\mathbf{0}_n)$
for j in $[\![p]\!]$ **do**
  $\boldsymbol{\Theta}^{ref}_j \leftarrow \boldsymbol{\theta}$
end for
$\mathcal{W}, \mathbf{m}^{ref} \leftarrow$ `update_W`$(\mathbf{X}, \boldsymbol{\theta}, [\![p]\!], \lambda_1, \emptyset)$ ▷ See Section 4.3.3
for $t = 1$ to T **do**
  # Pre-Solve
  $\mathbf{w}^*_{\mathcal{W}}, b^* \leftarrow \underset{\mathbf{w}_{\mathcal{W}}, b}{\operatorname{argmin}} P_{\mathbf{Z}_{\mathcal{W}}, \lambda_t}(\mathbf{w}_{\mathcal{W}}, b)$
  $\boldsymbol{\theta} \leftarrow -\nabla F(\mathbf{Z}_{\mathcal{W}} \mathbf{w}^*_{\mathcal{W}} + b^* \mathbf{1}_n)$.
  $\mathcal{W}, \boldsymbol{m}^{ref} \leftarrow$ `clean_W`$(\mathcal{W}, \lambda_t, \boldsymbol{\theta}, \boldsymbol{\Theta}^{ref}, \boldsymbol{m}^{ref})$
  while true **do**
    # Branch pruning
    $\mathcal{V} \leftarrow \emptyset$
    for j in $[\![p]\!]$ **do**
      if $\eta(\mathbf{X}_j, \boldsymbol{\Theta}^{ref}_j, \boldsymbol{\theta}, \mathbf{m}^{ref}_j) > \lambda_t$ **then** ▷ See Section 4.3.2
        $\mathcal{V} \leftarrow \mathcal{V} \cup \{j\}$
        $\boldsymbol{\Theta}^{ref}_j \leftarrow \boldsymbol{\theta}$
      end if
    end for
    # Identify the working set
    $\mathcal{W}', \mathbf{m}^{ref}_{\mathcal{V}} \leftarrow$ `update_W`$(\mathbf{X}, \boldsymbol{\theta}, \mathcal{V}, \lambda_t, \mathcal{W})$ ▷ See Section 4.3.3
    if $\max_{i \in \mathcal{W}'} \left| \mathbf{Z}_i^\top \boldsymbol{\theta} \right| \leq \lambda$ **then** Break **else** $\mathcal{W} \leftarrow \mathcal{W}'$
    # Solve subproblem
    $\mathbf{w}^*_{\mathcal{W}}, b^* \leftarrow \underset{\mathbf{w}_{\mathcal{W}}, b}{\operatorname{argmin}} P_{\mathbf{Z}_{\mathcal{W}}, \lambda_t}(\mathbf{w}_{\mathcal{W}}, b)$
    $\boldsymbol{\theta} \leftarrow -\nabla F(\mathbf{Z}_{\mathcal{W}} \mathbf{w}^*_{\mathcal{W}} + b^* \mathbf{1}_n)$.
    $\mathcal{W}, \boldsymbol{m}^{ref} \leftarrow$ `clean_W`$(\mathcal{W}, \lambda_t, \boldsymbol{\theta}, \boldsymbol{\Theta}^{ref}, \boldsymbol{m}^{ref})$
  end while
  $(\mathcal{W}, \mathbf{w}^*_{\mathcal{W}}, b^*)_k \leftarrow (\mathcal{W}, \mathbf{w}^*_{\mathcal{W}}, b^*)$
end for

function `clean_W`$(\mathcal{W}, \lambda, \boldsymbol{\theta}, \boldsymbol{\Theta}^{ref}, \boldsymbol{m}^{ref})$
  for i in $\mathcal{W}$ **do**
    if $\left| \mathbf{Z}_i^\top \boldsymbol{\theta} \right| < \lambda$ **then**
      Remove $\{i\}$ from $\mathcal{W}$
      for b in branch(i) **do**
        if $\mathbf{m}^{ref}_b < \left| \mathbf{Z}_i^\top \boldsymbol{\Theta}^{ref}_b \right|$ **then** $\mathbf{m}^{ref}_b \leftarrow \left| \mathbf{Z}_i^\top \boldsymbol{\Theta}^{ref}_b \right|$
  return $\mathcal{W}$, $\boldsymbol{m}^{ref}$

Lemma 4.3.1. *For any* $\mathbf{X} \in \{0,1\}^{n\times p}$, $\mathbf{v} \in \mathbb{R}^n_+$, $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^n$, $j \in [\![p]\!]$, $\mathcal{I} \subset [\![p]\!]$ *and* $\alpha \in \mathbb{R}$, *the following holds:*

$$\max_{k\in\mathcal{I}} \left| \boldsymbol{\theta}_2^\top (\mathbf{v} \odot \mathbf{X}_k) \right| \leq |\,\alpha\,| \max_{k\in\mathcal{I}} \left| \boldsymbol{\theta}_1^\top (\mathbf{v} \odot \mathbf{X}_k) \right| + \zeta(\boldsymbol{\theta}_2 - \alpha\boldsymbol{\theta}_1, \mathbf{v})\,, \tag{4.5}$$

where

$$\forall (\mathbf{u}, \mathbf{v}) \in \mathbb{R}^n \times \mathbb{R}^n_+\,, \quad \zeta(\mathbf{u}, \mathbf{v}) = \max\left(\sum_{i:\mathbf{u}_i>0} \mathbf{u}_i\mathbf{v}_i, -\sum_{i:\mathbf{u}_i<0} \mathbf{u}_i\mathbf{v}_i \right).$$

The proof of Lemma 4.3.1 is postponed to Appendix B.1. It is based on the decomposition $\boldsymbol{\theta}_2 = \alpha\boldsymbol{\theta}_1 + (\boldsymbol{\theta}_2 - \alpha\boldsymbol{\theta}_1)$, and exploits the tree structure among features in a branch. To exploit Lemma 4.3.1 in WHInter, we define for $\alpha \in \mathbb{R}$ the function

$$\forall\,(\mathbf{v}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, m) \in \mathbb{R}^n_+ \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}\,, \quad \eta_\alpha\,(\mathbf{v}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, m) = |\,\alpha\,|\, m + \zeta\,(\boldsymbol{\theta}_2 - \alpha\boldsymbol{\theta}_1, \mathbf{v})\,, \tag{4.6}$$

and we maintain an active set $\mathcal{W} \subset [\![D]\!]$, a matrix $\boldsymbol{\Theta}^{ref} \in \mathbb{R}^{n\times p}$ that contains *reference dual variables* $\boldsymbol{\Theta}^{ref}_j \in \mathbb{R}^n$ for each branch $j \in [\![p]\!]$, and the vector $\mathbf{m}^{ref} \in \mathbb{R}^p$ defined by:

$$\forall j \in [\![p]\!]\,, \quad \mathbf{m}^{ref}_j = \max_{k\in[\![p]\!]:\tau(j,k)\notin\mathcal{W}} \left| \mathbf{Z}^\top_{\tau(j,k)} \boldsymbol{\Theta}^{ref}_j \right|. \tag{4.7}$$

We now state our pruning theorem which allows to identify branches which are guaranteed to not contain any violating feature (line 13 of Algorithm 4.2):

Theorem 4.3.1 (Branch pruning)**.** *For any* $\boldsymbol{\Theta}^{ref} \in \mathbb{R}^{n\times p}$, $\mathcal{W} \subset [\![p]\!]$, $j \in [\![p]\!]$, *let* $\mathbf{m}^{ref}_j \in \mathbb{R}_+$ *be given by* (4.7)*. Then for any* $\boldsymbol{\theta} \in \mathbb{R}^n$, $\alpha \in \mathbb{R}$ *and* $\lambda > 0$, *if*

$$\eta_\alpha \left(\mathbf{X}_j, \boldsymbol{\Theta}^{ref}_j, \boldsymbol{\theta}, \mathbf{m}^{ref}_j \right) < \lambda\,, \tag{4.8}$$

then any feature from branch j *that belongs to the working set* $\left\{ i \in [\![D]\!] : \left| \mathbf{Z}_i^\top \boldsymbol{\theta} \right| \geq \lambda \right\}$ *is already in* $\mathcal{W}$*. This holds in particular if*

$$\eta_{min} \left(\mathbf{X}_j, \boldsymbol{\Theta}^{ref}_j, \boldsymbol{\theta}, \mathbf{m}^{ref}_j \right) := \min_{\alpha\in\mathbb{R}} \eta_\alpha \left(\mathbf{X}_j, \boldsymbol{\Theta}^{ref}_j, \boldsymbol{\theta}, \mathbf{m}^{ref}_j \right) < \lambda\,. \tag{4.9}$$

Proof. Take $\mathcal{I} = \{k \in [\![p]\!] : \tau(j,k) \notin \mathcal{W}\}$, $\mathbf{v} = \mathbf{X}_j$, $\boldsymbol{\theta}_1 = \boldsymbol{\Theta}^{ref}_j$ and $\boldsymbol{\theta}_2 = \boldsymbol{\theta}$ in Lemma 4.3.1. Then if (4.8) holds, we deduce from (4.5) that

$$\max_{k\in[\![p]\!]:\tau(j,k)\notin\mathcal{W}} \left| \mathbf{Z}^\top_{\tau(j,k)} \boldsymbol{\theta} \right| < \lambda\,.$$

This shows that there is no feature i in branch j such that $\left| \mathbf{Z}_i^\top \boldsymbol{\theta} \right| \geq \lambda$ *and* i is not already in $\mathcal{W}$. The fact that for fixed arguments, the function $\alpha \to \eta_\alpha$ has a minimum $\alpha^* \in \mathbb{R}$ is shown in Appendix B.2, along with with an algorithm to compute it in $O\left(\| \mathbf{X}_j \|_0 \ln \| \mathbf{X}_j \|_0\right)$ operations. Since the statement is true for any α, it is *a fortiori* true for α^*. □

Theorem 4.3.1 provides criteria (4.8) and (4.9) that can be computed for each branch j, and which if satisfied allow to skip the search for violating variables in the branch. Importantly, the features that are already in the working set $\mathcal{W}$ are not taken into account to compute the

criterion for a given branch. This subtlety allows to rule out branches even if they already contain features that were previously incorporated in the working set. Note that the reference dual variable for branch j, i.e, $\mathbf{\Theta}_j^{ref}$, is kept unchanged as long as branch j is pruned, and is otherwise updated to the latest dual variable (line 15 of Algorithm 4.2). As $\mathbf{m}_j^{ref}$ depends on the reference dual variable instead of the current one, it is solely reevaluated each time the reference residual is updated (line 18 of Algorithm 4.2) or when a feature from branch j leaves the working set (line 22 of Algorithm 4.2) .

Criterion (4.9) is the most stringent one, and therefore the most efficient one to prune branches, but it takes $O\left(\| \mathbf{X}_j \|_0 \ln \| \mathbf{X}_j \|_0\right)$ operations to compute. In order to balance computational complexity of the bound with its efficacy to prune branches, criterion (4.8) can be used as an alternative for a specific α value. One simple choice is to just take $\alpha = 1$, which leads to the criterion

$$\eta_1\left(\mathbf{X}_j, \mathbf{\Theta}_j^{ref}, \boldsymbol{\theta}, \mathbf{m}_j^{ref}\right) = \mathbf{m}_j^{ref} + \zeta\left(\boldsymbol{\theta} - \mathbf{\Theta}_j^{ref}, \mathbf{X}_j\right) < \lambda\,. \tag{4.10}$$

Alternatively, a simple heuristic to expect a more efficient pruning is to choose an α that minimises $\| \left(\boldsymbol{\theta} - \alpha \mathbf{\Theta}_j^{ref}\right) \odot \mathbf{X}_j \|_2$, i.e,

$$\alpha_{\ell_2} = \frac{\boldsymbol{\theta}^\top \left(\mathbf{\Theta}_j^{ref} \odot \mathbf{X}_j\right)}{\| \mathbf{\Theta}_j^{ref} \odot \mathbf{X}_j \|_2^2}\,. \tag{4.11}$$

$\eta_{\alpha_{\ell_2}}$ is expected to be more effective than η_1 since it is reasonable to expect that $\zeta\left(\boldsymbol{\theta} - \alpha_{\ell_2}\mathbf{\Theta}_j^{ref}, \mathbf{X}_j\right)$ is smaller than $\zeta\left(\boldsymbol{\theta} - \mathbf{\Theta}_j^{ref}, \mathbf{X}_j\right)$. Overall, computing criterion (4.9) for $\alpha = 1$ as in (4.10), or for $\alpha = \alpha_{\ell_2}$ as in (4.11), is an $O(\| \mathbf{X}_j \|)$ operation. Since computing $\zeta(\boldsymbol{\theta} - \alpha\mathbf{\Theta}_j^{ref}, \mathbf{X}_j)$ for a fixed α is also a $O(\| \mathbf{X}_j \|)$ computation, the total cost of identifying branch j as violated is $O(\| \mathbf{X}_j \|)$ for criterion (4.10), compared to $O\left(\| \mathbf{X}_j \|_0 \ln \| \mathbf{X}_j \|_0\right)$ for criterion (4.9). In Algorithm 4.2, the notation η refers to a user-defined function among $\eta_1, \eta_{\alpha_{\ell_2}}$ or η_{min}.

4.3.3 Updating the working set

When some branches $\mathcal{V} \subset [\![p]\!]$ cannot be pruned, the simultaneous updates of the working set $\mathcal{W}$ and of $\mathbf{m}_{\mathcal{V}}^{ref}$ requires scanning through all features in the branches $\mathcal{V}$ (lines 5 and 18 in Algorithm 4.2). In what follows we discuss strategies to make these updates efficient. For that purpose, let us first notice that:

$$\begin{aligned}
\forall j, k \in [\![p]\!]\ , \left|\mathbf{Z}_{\tau(j,k)}^\top \boldsymbol{\theta}\right| &= \left|\left(\mathbf{X}_j \odot \mathbf{X}_k\right)^\top \boldsymbol{\theta}\right| \\
&= \left|\left(\mathbf{X}_j \odot \boldsymbol{\theta}\right)^\top \mathbf{X}_k\right| \\
&= \left|\boldsymbol{Q}_j^\top \mathbf{X}_k\right|\ ,
\end{aligned}$$

where for any $j \in [\![p]\!]$, $\boldsymbol{Q}_j = \boldsymbol{X}_j \odot \boldsymbol{\theta}$. This allows us to write the updates of $\mathcal{W}$ and $\boldsymbol{m}_{\mathcal{V}}^{ref}$ as:

$$\begin{cases}
\mathcal{W}' & = \mathcal{W} \cup \left\{\tau(j,k) : j \in \mathcal{V}, k \in [\![p]\!], \left|\boldsymbol{Q}_j^\top \boldsymbol{X}_k\right| \geq \lambda\right\}\ , \\
\boldsymbol{m}_j^{ref} & = \max_{k:\, \left|\boldsymbol{Q}_j^\top \boldsymbol{X}_k\right| < \lambda} \left|\boldsymbol{Q}_j^\top \boldsymbol{X}_k\right|\ , \forall j \in \mathcal{V}\,.
\end{cases} \tag{4.12}$$

This highlights the fact that the updates of the working set $\mathcal{W}$ and of $\boldsymbol{m}_{\mathcal{V}}^{ref}$ can be cast as particular variants of the Maximum Inner Product Search (MIPS) problem. MIPS aims at finding a vector in a database of probes which maximises the inner product with a given query vector. If we consider $\boldsymbol{X}$ as a set of probes, and $\boldsymbol{Q}_j$ as a query, then (4.12) is a variant of MIPS where (i) the set of probe vectors satisfies some constraints and is not known upfront and (ii) the problem is a maximum *absolute* inner product search. The update of $\mathcal{W}$ involves what is sometimes referred to as *above*-λ-MIPS problems where again, maximum *absolute* inner products are considered.

The interest of casting these updates as variants of MIPS problems is to exploit the ideas developed in the literature for solving these problems efficiently. Teflioudi and Gemulla [2016] and Fontoura et al. [2011] give good overviews of MIPS solvers developed for recommender systems and information retrieval applications respectively. In both cases, the proposed methods rely on two main ideas: (i) adequate indexing techniques or data structures and (ii) pruning criteria which allow to not compute all inner products entirely. Since none of these methods can directly be applied to problem (4.12) because of its specificities, we propose an appropriate algorithm based on a simple inverted index approach, which we will refer to as IL, and which exploits the sparsity of the problem. Another option would be to leverage pruning techniques. We detail such an attempt in Appendix B.3. However, since our preliminary results with the pruning technique were not conclusive compared to IL on the simulated and real data, we will only focus on the inverted index approach below.

Algorithm 4.3 `update_W`

Input: $\mathbf{X} \in \{0,1\}^{n \times p}$, $\boldsymbol{\theta} \in \mathbb{R}^n$, $\mathcal{Q} \subset [\![p]\!]$, $\lambda \in \mathbb{R}$, $\mathcal{W} \subset [\![D]\!]$
Output: $\mathcal{W}$, $\mathbf{m}^{ref}$

for $j \in \mathcal{Q}$ **do**
  $\boldsymbol{m}_j^{ref} = 0$
  Set $\mathbf{a}_k = 0$ for all $k \in [\![p]\!]$
  for each i in supp$(\mathbf{X}_j)$ **do**
    for each k in supp$(\boldsymbol{x}_i)$ **do**
      $\mathbf{a}_k = \mathbf{a}_k + \boldsymbol{\theta}_i$
    end for
  end for
  for each k s.t. $\mathbf{a}_k \neq 0$ **do**
    if $\mathbf{m}_j^{ref} < |\,\mathbf{a}_k\,| < \lambda$ **then** set $\mathbf{m}_j^{ref} = |\,\mathbf{a}_k\,|$
    if $|\,\mathbf{a}_k\,| \geq \lambda$ and $\tau(j,k) \notin \mathcal{W}$ **then** add $\tau(j,k)$ to $\mathcal{W}$
  end for
end for
return $\mathcal{W}, \boldsymbol{m}^{ref}$

IL is detailed in Algorithm 4.3. The inverted indices consist of n lists, one for each dimension, where each list supp$(\boldsymbol{x}_i)$ records the indices of the features in $\mathbf{X}$ which have a non-zero element for the i^{th} dimension. These inverted lists can be computed once for all when WHInter starts and be reused for all MIPS problems, and therefore building the inverted lists requires a negligible additional computational cost. Algorithm 4.3 computes inner product following a *term-at-a-time* (TAAT) scheme [Fontoura et al., 2011], i.e, the inner products are accumulated simultaneously

across probes and the contribution of the i^{th} dimension to the inner products is entirely processed before moving to the next one.

4.4 Simulation study

We first test the performances of WHInter on synthetic LASSO datasets. We assess the performances of the different branch pruning bounds presented in Section 4.3.2, i.e, η_{min}, η_1 and $\eta_{\alpha_{\ell_2}}$, and further compare WHInter to a working set method that uses the bound $\zeta(\boldsymbol{\theta}, \mathbf{X}_j)$ instead of η_α, but is otherwise equivalent to WHInter. We refer to this method as $\zeta + IL$. It is expected to prune less branches than WHInter but does not require to maintain $\boldsymbol{m}^{ref}$. We also compare WHInter to SPP [Nakagawa et al., 2016] and BLITZ [Johnson and Guestrin, 2015]. In our experiments, we use a slightly modified, more efficient version of the code provided by the authors of SPP (see Appendix B.4). As for BLITZ, since the method is not tailored for interaction problems, we first compute the matrix $\boldsymbol{Z}$ which is fed as input to BLITZ. For this reason we could not solve problems when p is too large (e.g., $p = 1 \times 10^4$ in the simulations) since, even in sparse format, storing $\boldsymbol{Z}$ requires too much memory. Importantly, the performances reported for BLITZ do not include the time required to compute $\boldsymbol{Z}$ from $\boldsymbol{X}$, which clearly advantages BLITZ compared to the other methods.

We simulate five datasets $\mathbf{X} \in \{0,1\}^{n \times p}$ with varying number of features and samples: three datasets with $p = 1 \times 10^3$ fixed and $n \in \{3 \times 10^2, 1 \times 10^3, 1 \times 10^4\}$, and two more with $n = 1 \times 10^3$ fixed and $p \in \{3 \times 10^3, 1 \times 10^4\}$. The features are drawn from a Bernoulli distribution with parameter $q \in [0.1, 0.5]$ itself drawn from a uniform distribution $\mathcal{U}_{[0.1,0.5]}$. We then randomly pick a set $\mathcal{S}$ of 100 features among the main effects and interactions and compute the response as $\mathbf{y} = \mathbf{Z}_{\mathcal{S}} \mathbf{w}^*_{\mathcal{S}}$ where $\mathbf{w}^*_{\mathcal{S}} \sim \mathcal{N}(\mathbf{0}_{|\mathcal{S}|}, I_{|\mathcal{S}|})$. In all experiments, the LASSO is solved for a sequence $(\lambda_t)_{t \in [\![T]\!]}$, $T = 100$, logarithmically spaced between λ_{max} and $\max(0.01\lambda_{max}, \lambda')$ where λ_{max} is the largest value of λ for which at least one feature is selected, and λ' is the first λ_i for which 150 features or more are selected in the model. For all methods, the time to compute λ_{max} is included in the total time required to solve the regularisation path. In WHInter, λ_{max} can easily be deduced from the initialisation of $\mathbf{m}^{ref}$ since $\lambda_{max} = \max_{j \in [\![p]\!]} \boldsymbol{m}^{ref}_j$. All algorithms are implemented in C++ and compiled with the `-O3` optimisation flag. The experiments are run on a 64-bit machine with Intel Core i7 Processor 2.5 GHz, 16GB of memory and 6MB of cache.

Results are shown in Fig. 4.2. For $n = 1 \times 10^3$ (Fig. 4.2a), LASSO solutions are computed for 42, 32 and 28 values of λ for $p = 1 \times 10^3, p = 3 \times 10^3$ and $p = 1 \times 10^4$ respectively. In these cases smaller values of λ result in model sizes exceeding 150 features. For the remaining settings where $p = 1 \times 10^3$ and $n = 3 \times 10^2$ or $n = 1 \times 10^4$ (Fig. 4.2b), LASSO solutions are computed for 34 and all 100 values of λ between λ_{max} and $0.01\lambda_{max}$, respectively. All methods returned the exact same support for all values of λ (Fig. B.4).

In all settings, WHInter is the fastest method. Its better performance compared to $\zeta + IL$ highlights the benefit of using reference dual variables even if it implies to maintain $\boldsymbol{m}^{ref}$. The results also show the importance of α, since WHInter with η_{ℓ_2} is always better ($\times 1.2$ to $\times 1.8$) than WHInter with η_1 for example. Figure 4.2c confirms that the choice of α has an impact on the pruning efficiency and consequently on the performance. It shows, however, that on this experiment η_{min} does not allow to prune many more branches than η_{ℓ_2}. This explains why η_{ℓ_2} tends to outperform η_{min}, notably for large n, since the higher computational complexity of η_{min} does not sufficiently enhance the pruning. We also notice that SPP is the slowest algorithm,

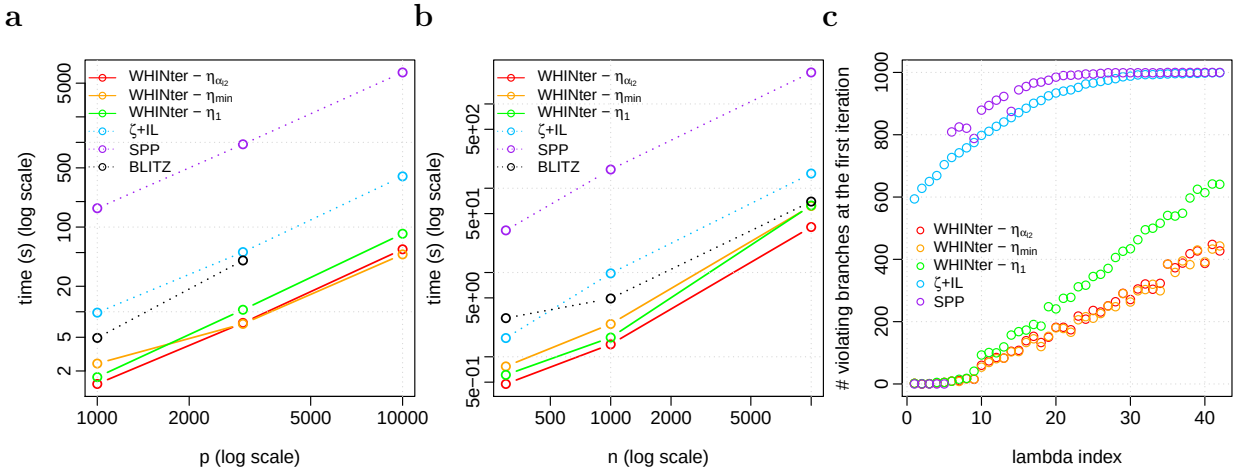


Figure 4.2 – Performance comparison on simulated data for an entire regularisation path. Comparison of WHInter with three branch pruning criteria $\eta \in \{\eta_{\alpha_2}, \eta_{min}, \eta_1\}$ to $\zeta + IL$, SPP and BLITZ. (a) Time in seconds for $n = 1 \times 10^3$ fixed and p varied. (b) Time in seconds for $p = 1 \times 10^3$ fixed and n varied. (c) Number of branches which could *not* be pruned at the first iteration, as a function of λ, for $n = p = 1 \times 10^3$.

and in particular $\zeta + IL$ is $\times 17$ faster than SPP on average. This speed-up is mostly explained by the fact that $\zeta + IL$ relies on inverted lists to update the working set while SPP identifies the safe set naively. Overall, WHInter offers a signifiant speed-up of two orders of magnitude or more compared to its safe screening counterpart.

4.5 Results on real world data

We now illustrate the performance of the different algorithms on a real-world problem, where we want to predict the cytotoxic response of 884 lymphoblastoid cell lines split into a train ($n = 620$) and a test ($n = 264$) set, and characterised by about 1.2×10^6 single nucleotide polymorphisms (SNP) that represent their genotypes. The data was released as part of the Dialogue on Reverse Engineering Assessment and Methods 8 (DREAM 8) toxicogenetics challenge [Eduati et al., 2015]. We encode the SNP data as a binary matrix were 1 stand for the presence of a minor allele on one or both copies of the chromosomes. As preprocessing we removed SNP with less than 5% of 1's and corrected the data for population structure as in Price et al. [2006]. To focus on problems of increasing scales, we first considered the SNPs of the smallest chromosome only (chr. 22), then of the largest only (chr. 1) and finally of all chromosomes together. This leads to train matrices with $n = 620$ and $p = 18,168$ SNPs for chromosome 22, $p = 89,027$ SNPs for chromosome 1 and $p = 1,166,836$ SNPs for the whole genome. Figure B.5 provides an overview of the whole genome SNPs matrix sparsity. We consider a sequence of 100 regularisation parameters λ logarithmically spaced between λ_{max} and $0.01\lambda_{max}$, and by default stop computations as soon as 150 features or more are selected. This occurs after the 12^{th}, the 11^{th} and the 9^{th} value of λ for chromosome 22, chromosome 1 and all chromosomes respectively. The time required to compute the regularisation paths are shown in Fig. 4.3.

The relative performances of the methods are the same as for the simulations. $\eta_{\alpha_{\ell_2}}$ provides

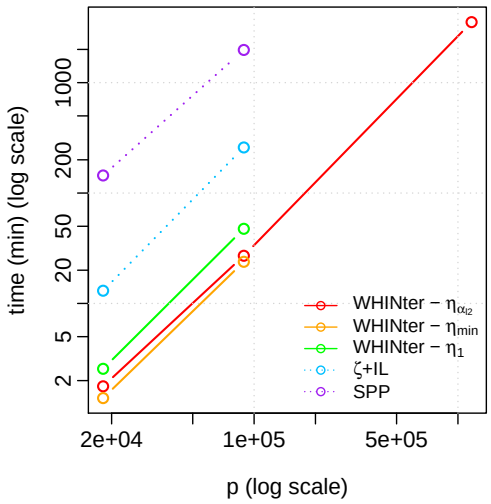


Figure 4.3 – **Performance comparison on SNPs data for an entire regularisation path**. The y-axis reports the total time (in minutes) required to compute the LASSO path for chromosome 22 (around 20,000 SNPs), chromosome 1 (around 90,000 SNPs) and the whole genome (around 1.2 million SNPs).

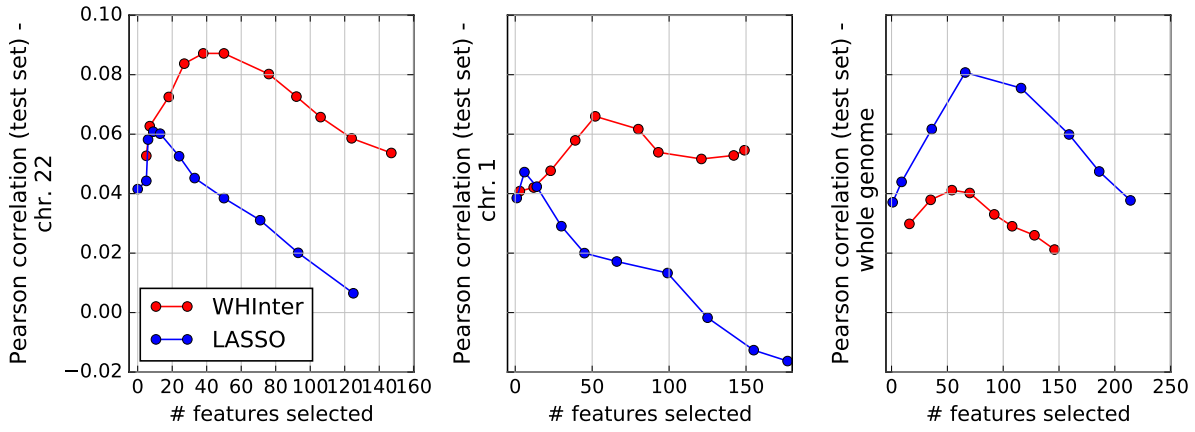


Figure 4.4 – **Predictive performance on the test set**. The y-axis reports the pearson correlation between the true and predicted response. The x-axis reports the number of selected features for the sequence of regularisation parameters tested.

a $\times 1.4$ (resp. $\times 1.8$) speed up compared to using η_1 for chromosome 22 (resp. chr. 1). and compared to SPP, there is a $\times 81$ (resp. $\times 73$) speed up for chromosome 22 (resp chr. 1). In the case of the whole genome, we only ran WHInter with $\eta_{\alpha_{\ell_2}}$ which takes two days and a half. While this can seem a lot, we recall that this corresponds to a problem with roughly 680 *billion* features. We did not run other methods on the whole genome since most of them are expected to take too long.

Out of curiosity, we also obtained preliminary results concerning the predictive performance of WHInter compared to a LASSO with no interactions on such high-dimensional problems. The results, presented in Fig. 4.4, suggest that interactions are relevant predictors for this data. For the chromosomes 1 and 22 taken independently, the predictive accuracy of WHInter is better than that of the simple LASSO for almost every value of λ. By contrast, for the whole genome, the LASSO clearly performs better, which may underline statistical issues due to the huge number of variables in this case [Donoho and Tanner, 2009].

4.6 Related work

WHInter is related to its closest competitor Safe Pattern Pruning and to strategies for prioritising updates in coordinate descent (CD) algorithms.

4.6.1 Safe pattern pruning

When the number of interaction terms becomes too large, it can be prohibitive to compute safe screening rules for every feature. SPP [Nakagawa et al., 2016] addresses this issue with a safe screening criterion applicable to entire subtrees. Safely screening subtrees allows to narrow down the set of candidate interactions without sacrificing the optimality of the obtained solution. There are two key ingredients to SPP. The first one is that interaction features should be smaller than their corresponding parent features entrywise. The second ingredient is the GAP safe screening rules. Given primal and dual feasible solutions $(\mathbf{w}, b)$ and $\boldsymbol{\theta}$, the GAP safe sphere test states that:

$$\text{If} \quad GAP(\mathbf{X}_j) = \left|\mathbf{X}_j^\top \boldsymbol{\theta}\right| + r_\lambda(\mathbf{w}, b, \boldsymbol{\theta})\|\mathbf{X}_j\|_2 < \lambda\,, \quad \text{then} \quad \mathbf{w}_j^* = 0\,. \tag{4.13}$$

where $r_\lambda(\mathbf{w}, b, \boldsymbol{\theta})$ is proportional to the square root of the dual gap. Because the GAP safe sphere test depends on the current estimates of $(\mathbf{w}, b)$ and $\boldsymbol{\theta}$, the GAP safe rules are said to be *dynamic*. While the solver proceeds towards the optimal solution, $r_\lambda(\mathbf{w}, \boldsymbol{\theta})$ decreases with the dual gap and a growing number of features can be eliminated. The main idea of SPP is to derive an upper-bound $SPPC(\mathbf{X}_j)$ on the GAP sphere test criterion that makes it applicable to all features in a subtree. Importantly this upper bound should only depend on the root of the subtree $\mathbf{X}_j$ and not on the interaction features that descend from it so as to obtain a test with relatively low computational complexity. Formally, $SPPC(\mathbf{X}_j)$ is such that:

$$\forall k \in [\![p]\!]\,, \quad GAP(\mathbf{X}_{\tau(j,k)}) \leq SPPC(\mathbf{X}_j). \tag{4.14}$$

It follows from (4.14) that if $SPPC(\mathbf{X}_j) < \lambda$, then all features in the subtree with root $\mathbf{X}_j$ can be safely discarded from the optimisation problem. We now state the Safe Pattern Pruning

criterion. Given primal and dual feasible points $(\mathbf{w}, b)$ and $\boldsymbol{\theta}$, the following holds for any feature $\mathbf{X}_j$:

$$\text{If} \quad SPPC(\mathbf{X}_j) < \lambda\,, \quad \text{then} \quad \forall k \in [\![p]\!] \quad \mathbf{w}^*_{\tau(j,k)} = 0. \tag{4.15}$$

where $SPPC(\mathbf{X}_j) = \max\left(\sum_{i:\boldsymbol{\theta}_i>0} \mathbf{X}_{ij}\boldsymbol{\theta}_i\,, -\sum_{i:\boldsymbol{\theta}_i<0} \mathbf{X}_{ij}\boldsymbol{\theta}_i\right) + r_\lambda(\mathbf{w}, b, \boldsymbol{\theta})\|\mathbf{X}_j\|_2$ and $r_\lambda(\mathbf{w}, b, \boldsymbol{\theta})$ is proportional to the square root of the dual gap. Here we only presented SPP for second-order interaction terms but it should be mentioned that the method also applies to higher order interactions.

WHInter was inspired from SPP and addresses some of its drawbacks. One of them is that it is conservative. Indeed, the number of branches that can be screened can be quite low notably when the solver is not close enough to the optimal solution. When this is the case the GAP safe rules have to be applied to many interaction features which is very time consuming since it scales as the square of the number of branches which could not be screened. Another drawback stems from the use of a dual feasible point in the safe screening criterion. Indeed, obtaining a dual feasible point that needs to satisfy twice as many constraints as the number of features (original and interactions) is expensive. In fact it is almost as expensive as the safe screening step itself. WHInter deals with both of these issues. The working set strategy discards entire branches very aggressively and does not rely on a dual feasible point. This however comes at a cost, that of checking the KKT conditions and defining a working set among roughly p^2 features at each iteration. Making this affordable is what lies at the heart of WHInter.

4.6.2 Prioritisation of updates in coordinate descent

Coordinate descent (CD) is a highly popular algorithm for solving large scale LASSO problems. CD minimises the objective function one coordinate at a time while the others are kept fixed. More specifically, if the objective function is minimised with regards to coordinate i, then the CD update for the LASSO reads:

$$\boldsymbol{u}_i \leftarrow \frac{\mathbf{X}_i^\top}{\|\mathbf{X}_i\|^2}\left(\mathbf{y} - \sum_{j\neq i} \mathbf{X}_j \mathbf{w}_j\right) \tag{4.16}$$

$$\mathbf{w}_i \leftarrow ST\left(\boldsymbol{u}_i, \frac{\lambda}{\|\mathbf{X}_i\|^2}\right) \quad \text{where} \quad ST(\boldsymbol{u}_i, \frac{\lambda}{\|\mathbf{X}_i\|^2}) = \begin{cases} \boldsymbol{u}_i - \frac{\lambda}{\|\mathbf{X}_i\|^2} & \text{if} \quad \boldsymbol{u}_i > \frac{\lambda}{\|\mathbf{X}_i\|^2} \\ \boldsymbol{u}_i + \frac{\lambda}{\|\mathbf{X}_i\|^2} & \text{if} \quad \boldsymbol{u}_i < -\frac{\lambda}{\|\mathbf{X}_i\|^2} \\ 0 \quad \text{if} \quad |\boldsymbol{u}_i| \leq \frac{\lambda}{\|\mathbf{X}_i\|^2} \end{cases} \tag{4.17}$$

While CD works well in practice, recent works ([Fujiwara et al., 2016; Johnson and Guestrin, 2017]) have highlighted that most of the time spent by the solver is wasted in 'zero updates', i.e., updates for which the weight $\mathbf{w}_i$ is equal to zero before and after being updated. Consequently faster versions of CD have been proposed under the names Sling [Fujiwara et al., 2016] or StingyCD [Johnson and Guestrin, 2017] that avoid computing these 'zero updates'. These methods rely on a cheap (constant time) test which identifies ahead of time the updates which are guaranteed to be 'zero updates' so that they can be skipped safely. As we show below, these tests have interesting similarities with our WHInter branch bounds. Indeed, as can be seen from (4.17), a 'zero update' occurs when i) the weight $\mathbf{w}_i$ is zero before the update and ii) $|\mathbf{Z}_i| \leq \frac{\lambda}{\|\mathbf{X}_i\|^2}$. Let $\boldsymbol{\theta} = \mathbf{y} - \sum_j \mathbf{X}_j \mathbf{w}_j$ be the current residual. Note that the first condition

($\mathbf{w}_i = 0$) implies that $\boldsymbol{\theta} = \mathbf{y} - \sum_{j \neq i} \mathbf{X}_j \mathbf{w}_j$ Then expanding the second condition thanks to (4.16) we get:

$$\left| x_i^\top \boldsymbol{\theta} \right| \leq \lambda \tag{4.18}$$

In order to get a constant time test from (4.18), the authors of Sling and StingyCD resort to introducing a reference vector and computing an upper bound on $\left|x_i^\top \boldsymbol{\theta}\right|$. In StingyCD, the reference vector is a reference residual $\boldsymbol{\theta}^{ref}$ and the test they propose for the LASSO can be obtained as follows:

$$\begin{aligned}
\left|\mathbf{X}_i^\top \boldsymbol{\theta}\right| &= \left|\mathbf{X}_i^\top \left(\boldsymbol{\theta}^{ref} + \boldsymbol{\theta} - \boldsymbol{\theta}^{ref}\right)\right| \\
&\leq \left|\mathbf{X}_i^\top \boldsymbol{\theta}^{ref}\right| + \left|\mathbf{X}_i^\top \left(\boldsymbol{\theta} - \boldsymbol{\theta}^{ref}\right)\right| \\
&\leq \left|\mathbf{X}_i^\top \boldsymbol{\theta}^{ref}\right| + \|\mathbf{X}_i\| \left\|\boldsymbol{\theta} - \boldsymbol{\theta}^{ref}\right\| \quad \text{(Cauchy-Schwarz)}
\end{aligned}$$

It follows that if i) $\mathbf{w}_i$ is zero before the update and ii) $\left|\mathbf{X}_i^\top \boldsymbol{\theta}^{ref}\right| + \|\mathbf{X}_i\| \left\|\boldsymbol{\theta} - \boldsymbol{\theta}^{ref}\right\| \leq \lambda$ then the update can be safely skipped. Moreover this test is a constant time test. Indeed, $\left|\mathbf{X}_i^\top \boldsymbol{\theta}^{ref}\right|$ only needs to be computed once each time the reference residual is updated, $\|\mathbf{X}_i\|$ can be computed once for all before the algorithm starts and finally $\left\|\boldsymbol{\theta} - \boldsymbol{\theta}^{ref}\right\|$ can be computed in constant time as shown in the paper. This test is the one presented in [Johnson and Guestrin, 2017] for the general LASSO, although given with different notations and viewed from a slightly different perspective.

The link between StingyCD and WHInter lies in the idea of designing a cheap 'avoidance test' based on a reference residual. The former avoids wasteful coordinate updates and the latter avoids wasteful scanning of branches which do not contain any interaction feature that can enter the working set. In both cases the low computational cost of the test comes from the fact that the more computationally demanding terms depend on the reference residual and are therefore only computed each time the residual is updated. Although both tests in StingyCD and WHInter are based on the idea of a reference residual, they differ fundamentally for three reasons. The first reason is obviously linked with the fact that WHInter is designed to deal with interaction features. In particular, the test in WHInter should apply to a whole branch but only depend on the feature at its root. The second reason is linked to the computational complexity of the tasks they are designed to avoid, which implies different balances between the affordable computational complexity of the tests and their efficiency. Last but not least, we introduce in WHInter the parameter α which enhances the 'avoidance test' and reduces the frequency at which reference residuals are updated. This idea is new and could also benefit update prioritisation in CD.

4.7 Discussion

We presented WHInter, a working set algorithm designed to solve large scale LASSO problems with interaction terms. WHInter implements a new branch pruning bound to efficiently delineate the working set among the many possible interaction variables, and a variant of MIPS solver that provides a further speed up. We showed that WHInter is up to two orders of magnitudes faster than competing approaches. While we presented WHInter for binary data, it could also be used for data rescaled in $[0, 1]$, provided that an appropriate solver is picked for the MIPS problems.

As for future work, one could exploit the recent works on approximate MIPS [Shrivastava and Li, 2014; Teflioudi and Gemulla, 2016] to obtain an additional speed up for the computationally intensive updates, and possibly rely on recent post selection-inference [Suzumura et al., 2017] frameworks to characterise the approximate solution obtained.

Chapter 5

Conclusion

In the last twenty years, the development of modern microarrays followed by that of next generation sequencing technologies have paved the way for the large-scale characterisation of biological samples at the molecular level. As a consequence, large consortia have set up to finely characterise cancer genomes, or to map human genetic variability. These technologies allow to measure thousands or millions of molecular features for each sample, such as the expression level of each gene, its mutation status, or the allele present for each SNP. These measurements are by nature high-dimensional, since the number of biological samples available rarely reaches several thousands and even less millions. Moreover, measurements are hindered by the noise and biases inherent to the technologies and sample preparation protocols. Together with the high dimensionality of the data, this poses critical statistical and computational issues. This thesis aims at tackling such issues as part of cancer genomics and GWAS applications.

In chapter 2, we explore different representations of tumour mutation profiles, and assess their relevance for survival prediction and cancer stratification tasks. The extraction of relevant information from cancer mutation data proves to be a difficult task, mainly because most mutations are passengers which do not play a role in cancer, and because driver mutations occur at medium or low frequency across patients. To tackle these issues, we propose a simple feature engineering technique, NetNorM, which integrates tumour mutation profiles with gene networks. We show that the proposed data representation allows to obtain better survival predictions than the state-of-the-art, and that patients can be successfully stratified into subgroups with significantly different survival outcomes. We also strive to decipher the underlying biology captured by the proposed feature engineering technique, and to shed a critical and constructive light on previous related work.

In chapter 3, we present a supervised quantile normalisation procedure, termed Suquan. Quantile normalisation (QN) is pervasive across many data types in computational biology. It is used as a sample normalisation procedure, which is necessary to correct for batch effects and to control for potential technical artefacts. The 'median' target quantile is the usual default for QN, although nothing supports that this is the best choice across all tasks and data types. For example, NetNorM includes a QN step characterised by a step function. For this reason, we propose a principled approach for choosing the target quantile, where a linear predictor is optimised jointly with the target quantile for a given prediction task. We show that with appropriate regularisation on the target quantile we are able to improve prediction performances, compared to a priori fixed target quantiles, for a breast cancer relapse prediction task based on microarray expression data.

Finally in chapter 4, we focus on the hard problem of estimating ℓ_1 penalised linear models when including pairwise interactions in the design matrix. Linear models with sparsity enforc-

ing penalties, such as the LASSO or sparse logistic regression, are promising tools for GWAS applications. They allow simultaneous inference and feature selection, i.e, they provide a subset of SNPs that are predictive of the phenotype, as well as phenotype predictions that can be interpreted by analysing the selected SNPs. These models are however limited by their inability to capture non-linear effects. This is an important limitation since there is extensive evidence that gene-gene interactions, at the molecular level, are the rule rather than the exception, be it through protein complexes, through pathways, or through other mechanisms. Even so, it remains a challenge to compute ℓ_1 penalised estimates when features representing interactions are taken into account. Indeed, the huge number of possible interactions between SNPs poses computational issues. We therefore propose WHInter, an algorithm that efficiently estimates sparse linear models with pairwise interactions. We show that WHInter is able to scale to typical GWAS dataset sizes, with up to $O(10^{11})$ original features and interactions. This scale up provides new opportunities, both for prediction and data exploration purposes. While we motivated WHInter with GWAS applications, we note this contribution could also be of interest to many other applications where interactions, or co-occurrences, are thought to carry predictive power.

We now conclude this thesis with an outlook into future research directions related to our contributions.

Tumour mutation profiles Tumour mutation profiles are naturally represented as binary patients by genes mutation matrix where ones indicate the presence mutations. NetNorM takes such a binary matrix as input, where silent mutations have been filtered out beforehand. There is however a growing body of work which goes far beyond separating silent versus non silent mutations, and which identifies putative drivers versus passengers [Raphael et al., 2014]. A reasonable extension of our work would therefore use a sparse matrix as input where ones have been replaced by scores representing the probability that a given mutation is a driver. Such an extension could improve survival prediction and stratification performances.

In our work, survival prediction performances and patient stratification are used as means to evaluate different representations of mutation profiles, with the underlying idea that a representation which better captures explanatory factors will produce better performances. Predicting survival from mutation data is nonetheless an ambitious task. Indeed, from mutations to survival outcomes there are successive layers of complexity, including other molecular features such as copy number variations or methylation alterations, but also the variety of treatments administered to the patients. One way to avoid such uncontrolled complexity, at least partially, would be to consider predicting gene expression instead of survival. This would shed light on the relevance of the various representations on a intermediary task.

Finally, it would be useful to explore the possibility of converting NetNorM into a more straightforward feature extraction procedure, which would explicitly build features to represent the putative explanatory factors identified with NetNorM. These factors are the mutational status of a few important genes, the total mutational burden, and the neighbourhood mutational burden computed on small subgraphs. Such a procedure would have the advantage of clarifying the nature of the explanatory factors highlighted by NetNorM, and to provide an easier-to-use and easier-to-understand feature extractor.

Supervised quantile normalisation Our contribution focuses on the demonstration of the experimental performance of Suquan, as well as on practical implementation challenges.

A theoretical analysis of the algorithm, notably regarding its generalisation ability, is however missing. It could for example be enlightening to derive an upper bound on the approximation error of the model.

Moreover, the idea of developing Suquan originally emerged from our work on mutation profiles. Indeed, NetNorM implements a quantile normalisation with a step function as target quantile while previous work used the 'median' target quantile. An experimental evaluation of Suquan on mutation data would thus be a natural addition to our contribution.

ℓ_1 penalised linear models with pairwise interactions for GWAS We see several exciting research directions to extend WHInter or improve its significance.

A natural direction would be to trade the optimality of the solution guaranteed by WHInter for some additional speed up. WHInter particularly lends itself to such tradeoff. Indeed, as presented in chapter 4, it relies on a simple inverted index approach to solve MIPS problems. There is however an increasing number of works in the literature which aims at accelerating MIPS computations either through coordinate pruning strategies, or through locality sensitive hashing (LSH) (see Teflioudi and Gemulla [2016] for a review). Coordinate pruning strategies efficiently prune the search space so that not all coordinates of a probe vector need to be scanned to figure out that it will not produce the largest inner product. They can produce exact or approximate solutions to the MIPS problems. LSH [Gionis et al., 1999] is a popular and efficient algorithm for solving the nearest neighbour search problem in a given space. While it cannot be used to directly solve the MIPS problem, recent works [Bachrach et al., 2014; Neyshabur and Srebro, 2015; Shrivastava and Li, 2014, 2015] have shown that the MIPS problem could be reformulated as a nearest neighbour search problem in a higher dimensional space. This makes previously developed LSH techniques relevant to the MIPS problem. Of note, MIPS is equivalent to the nearest neighbour search in euclidean space or cosine similarity search when all probe vectors have equal ℓ_2 norm. Therefore if one is willing to standardise the features before fitting a sparse linear model, then the MIPS problem in WHInter can be readily tackled with LSH. It would be interesting to evaluate wether those more involved approaches could provide a significant speed up in the WHInter framework. Moreover, in the case that an approximate MIPS solver is used such as LSH, it would be beneficial to derive theoretical guarantees regarding the final sparse solution obtained.

In order to increase the significance of WHInter for applications where the analysis of the selected features is important, it would be of great interest to be able to evaluate the statistical significance of the selected features. This would be particularly useful for GWAS applications. Quantifying the uncertainty in the fitted estimate is indeed important for interpretation and reproducibility matters. Until recently, it was not possible to associate p-values to the coefficients of a LASSO estimate. The difficulty of this problem arises from the fact that the subset of variables selected is data dependent. However, recent work has proposed a methodology to address this challenge [Lee et al., 2016]. It has notably been extended to sparse interaction models, obtained via Marginal Screening and Orthogonal Matching Pursuit [Suzumura et al., 2017]. An interesting future direction would thus be to see whether the work of Lee et al. [2016] on post selective inference could be extended to the LASSO with pairwise interactions, possibly relying on computational tricks that make WHInter scalable.

Another interesting direction would be to extend WHInter to sparsity inducing penalties that enforce hierarchy constraints. Weak (resp. strong) hierarchy constraints allow an interaction to

be selected in the model if at at least one (resp. both) of the corresponding main effects is also selected. Such hierarchy constraints can be enforced by relying on a group LASSO penalty (see Bien et al. [2013] for a review). Comparing the performance of ℓ_1 penalised models with sparse hierarchy enforcing models could tell whether this additional regularisation proves beneficial for real world datasets.

Finally, applications to GWAS datasets are also faced by another challenge which is the presence of linkage disequilibrium (LD) in the data. Linkage disequilibrium refers to the correlation structure that exist between SNPs. Indeed, nearby SNPs tend to be highly correlated with each other. One possibility to handle LD consists in applying LD clumping before fitting a model. However, one may wonder whether it is possible to build an LD aware spare linear model that automatically handles LD in a one step procedure, and whether such a procedure would lead to better and more robust predictions [Vilhjalmsson et al., 2015].

We hope that the works in this thesis will inspire new fruitful ideas. The road is still long and strewn with obstacles towards the ultimate goals of modelling cancer cells behaviours based on molecular data, or building clinically leveragable polygenic risk scores for important diseases. We however think that as sequencing costs continue to decrease and sample sizes continue to increase, the impact of current works will grow.

Appendix A

Supplementaries for NetNorM

A.1 Supplementary figures

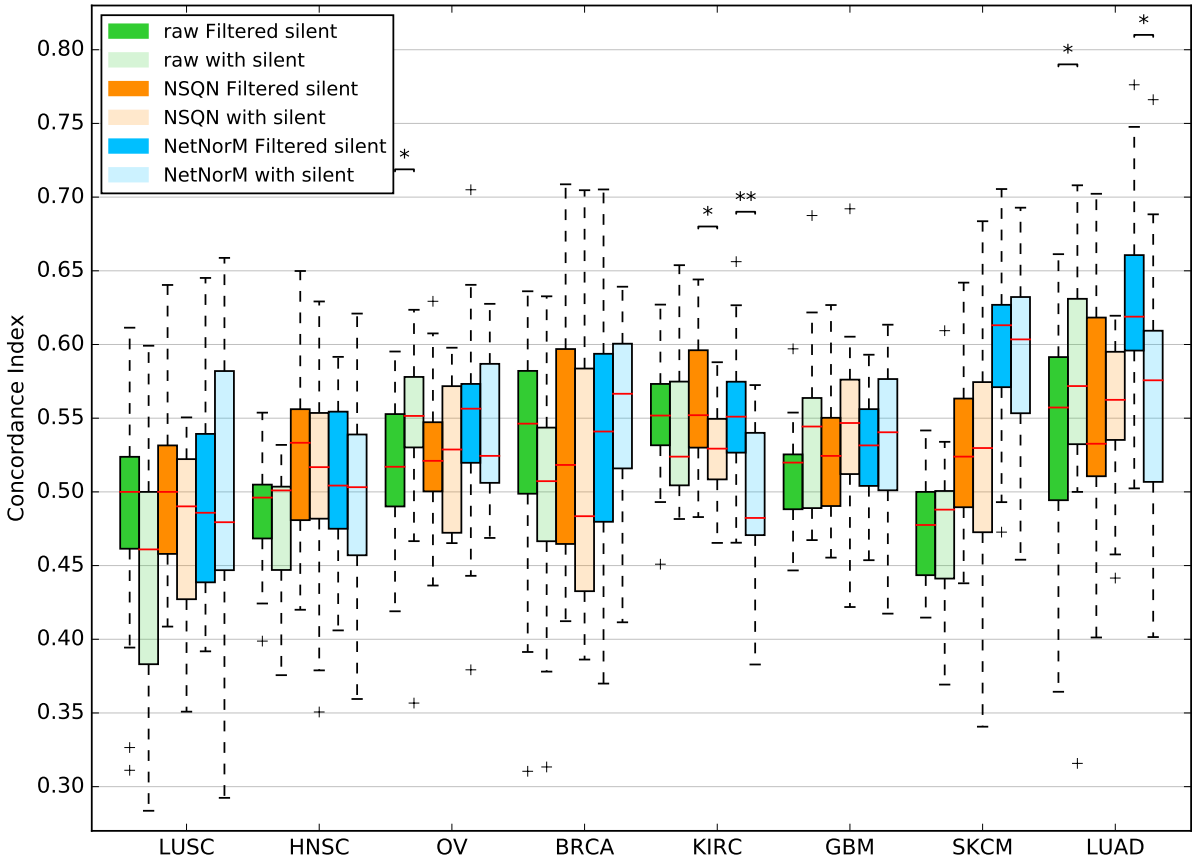


Figure A.1 – Effect of silent mutations on the survival predictive power of the raw mutation profiles, and mutation profiles processed with NSQN and NetNorM (with Pathway Commons as gene network). In the legend, 'Filtered silent' indicates that genes with silent mutations were not considered as mutated while 'with silent' indicates that genes with silent mutations were considered as mutated. For each cancer type, samples were split 20 times in training and test sets (4 times 5-fold cross-validation). Each time a sparse survival SVM was trained on the training set and the test set was used for performance evaluation. Wilcoxon signed rank tests were run to compare the performances obtained with and without silent mutations for each method and cancer type. Resulting P-values below 0.05 or 0.01 are indicated with asterisks ($P < 5 \times 10^{-2}$ (*) or $P < 1 \times 10^{-2}$ (**)).

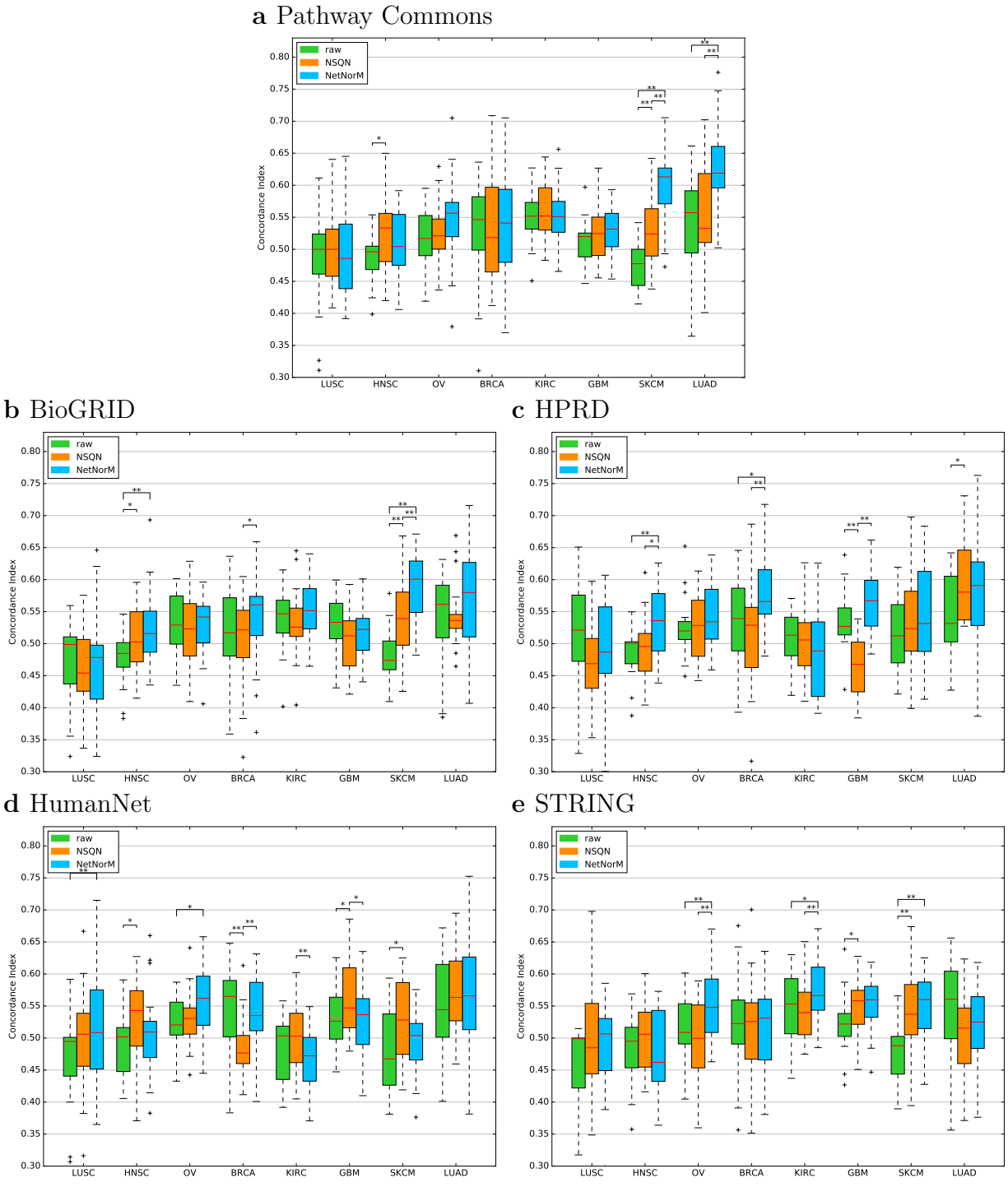


Figure A.2 – Survival predictive power of the mutation profiles processed with NSQN and NetNorM assessed with five different gene-gene interaction networks: Pathway Commons, BioGRID, HPRD, STRING and HumanNet. For STRING and HumanNet, only the top 10% most confident interactions were kept in the network. The performances obtained with the raw data slightly vary according to the network used since only the genes present in the network are considered. For each cancer type, samples were split 20 times in training and test sets (4 times 5-fold cross-validation). Each time a sparse survival SVM was trained on the training set and the test set was used for performance evaluation. The presence of asterisks indicate when the test CI is significantly different between 2 conditions (Wilcoxon signed rank test, $P < 5 \times 10^{-2}$ (*) or $P < 1 \times 10^{-2}$ (**)).

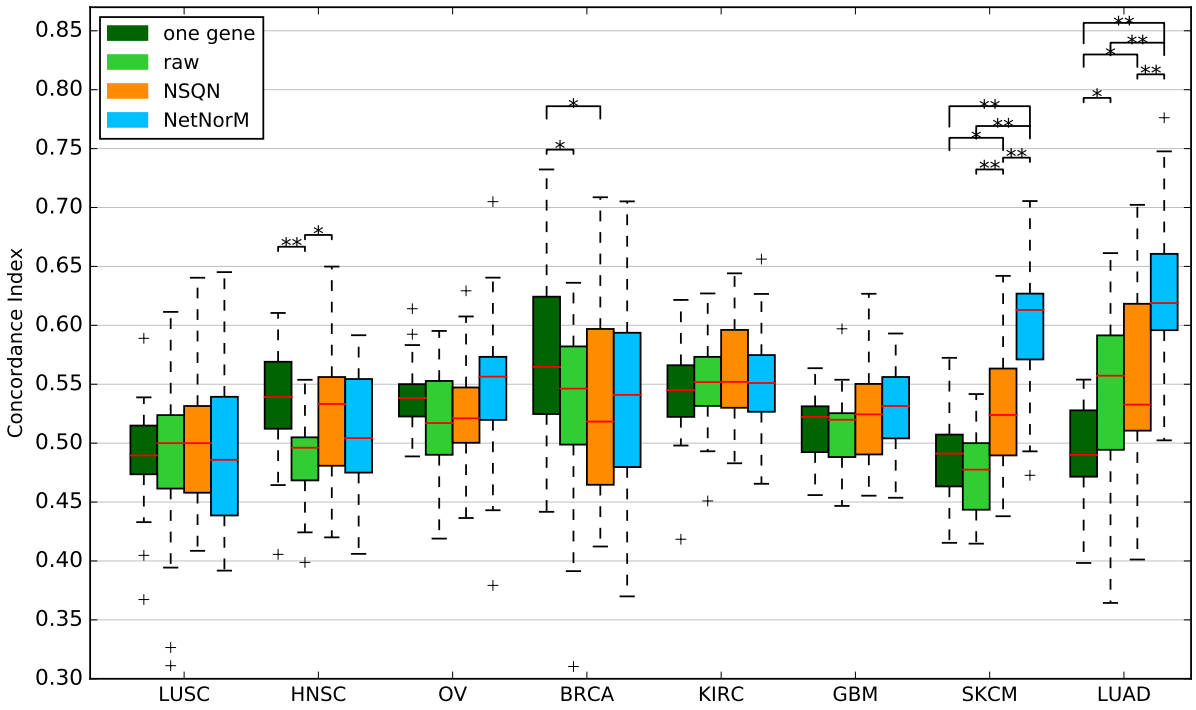


Figure A.3 – Comparison of the survival predictive power of: the most predictive gene, the raw mutation data, NSQN and NetNorM (with Pathway Commons as gene network) for 8 cancer types. For each cancer type, samples were split 20 times in training and test sets (4 times 5-fold cross-validation). In the case where only one gene was used to predict survival, the gene with the best concordance index on the training set was chosen and its performance evaluated on the test set. Otherwise, each time a sparse survival SVM was trained on the training set and the test set was used for performance evaluation. The presence of asterisks indicate when the test CI is significantly different between 2 conditions (Wilcoxon signed rank test, $P < 5 \times 10^{-2}$ (*) or $P < 1 \times 10^{-2}$ (**)).

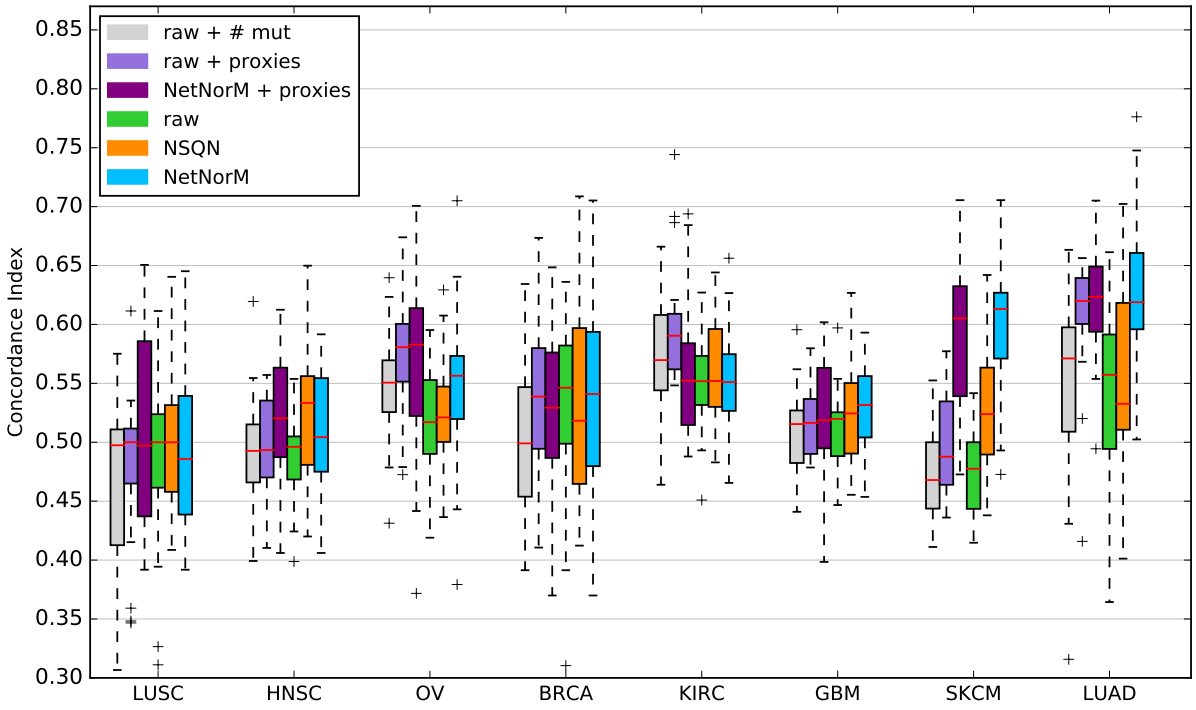


Figure A.4 – Survival predictive power of mutation data preprocessed according to five different schemes: 1) the raw data concatenated with a feature (scaled to unit variance) recording the total number of mutations in each patient (light gray); 2) the raw data concatenated with a feature called 'proxies' (scaled to unit variance) which is equal to 0 if the patient has more than k mutations (k is learned by cross-validation) and is equal to the total number of mutations otherwise (light purple), 3) the NetNorM representation concatenated with 'proxies' (purple) scaled to unit variance; 4) the raw binary mutation profiles; 5) mutation profiles processed with NSQN (orange); 6) mutation profiles processed with NetNorM (blue). Pathway Commons was used with NetNorM and NSQN. Samples were split 20 times in training and test sets (4 times 5-fold cross-validation). Each time a sparse survival SVM was trained on the training set and the test set was used for performance evaluation.

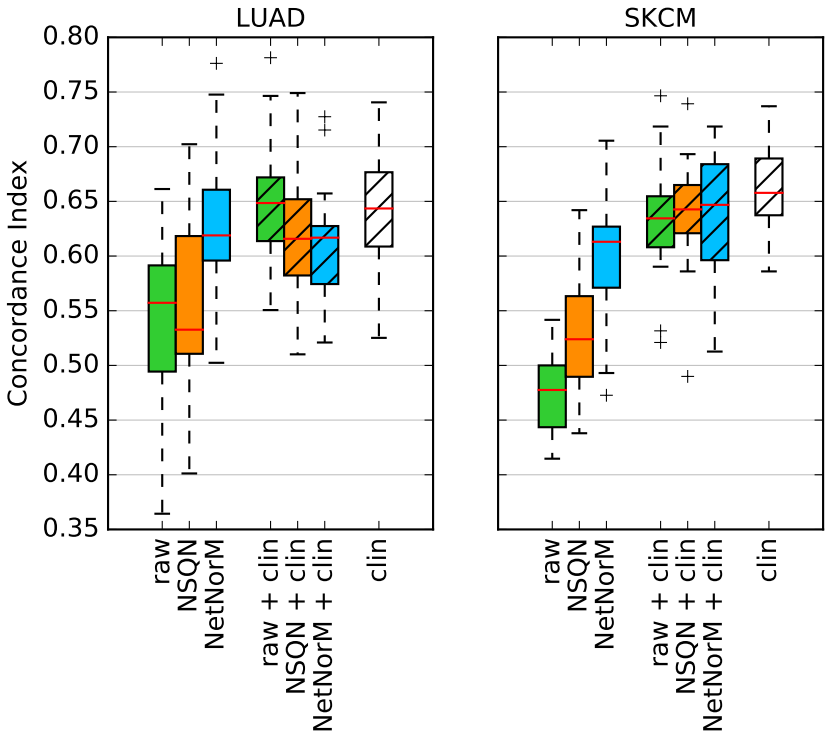
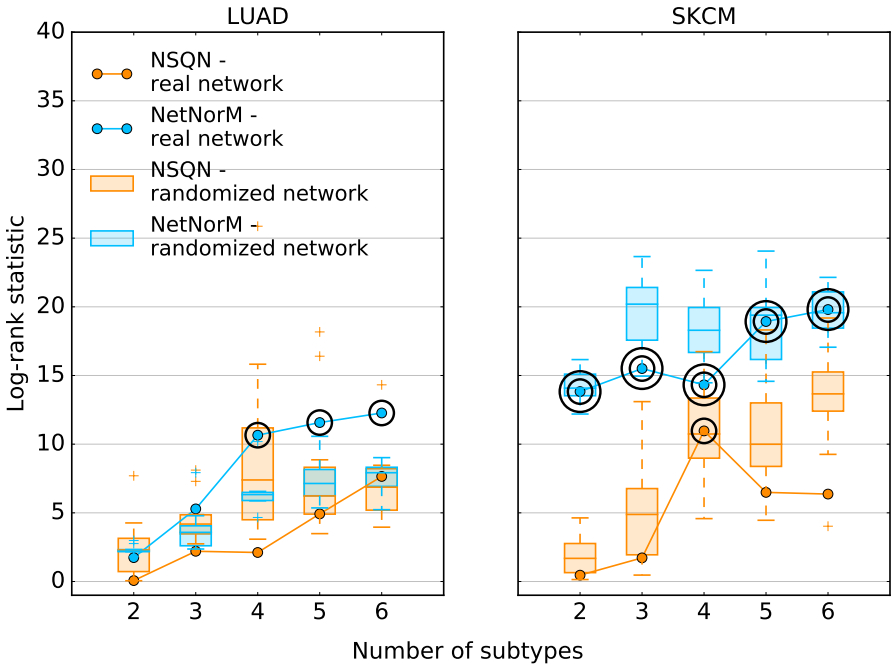


Figure A.5 – Survival predictive power of mutation data (raw binary mutations, mutations preprocessed with NSQN or NetNorM with Pathway Commons), clinical data, and the combination of both for LUAD and SKCM. The combination of both data types was obtained by concatenating the mutation features with the clinical features scaled to unit variance. For both cancers, samples were split 20 times in training and test sets (4 times 5-fold cross-validation). Each time a sparse survival SVM was trained on the training set and the test set was used for performance evaluation.

a



b

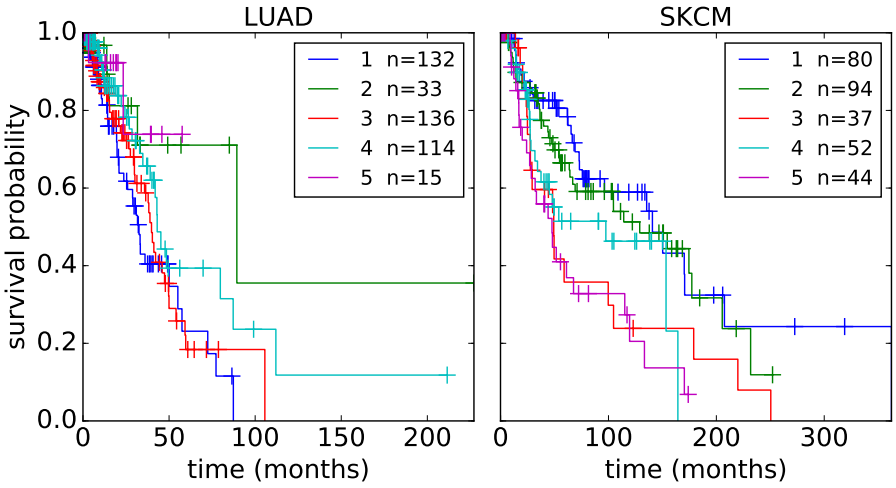


Figure A.6 – Patient stratification based on NetNorM (resp. NSQN) with hyperparameter k (resp. α) set to the value learned cross-validation for the survival prediction task instead of the default value. The stratification was obtained using NMF with consensus clustering. (a) Effect of network randomisation on patient stratification. Log-rank statistic obtained with Pathway Commons (curve) and 10 randomised versions of Pathway Commons (boxplots) with NetNorM (blue) and NSQN (orange) for LUAD and SKCM. One circle indicate a P-value $P \leq 5 \times 10^{-2}$ and two concentric circles indicate $P \leq 1 \times 10^{-2}$. (b) Kaplan Meir survival curves for NetNorM subtypes with significantly distinct survival outcomes (we illustrated the case with 5 subgroups for both LUAD and SKCM) . In the legend are indicated the subtype number followed by the number of patients in the subtype.

A.2 Supplementary tables

LUSC		HNSC		OV		BRCA		KIRC		GBM		SKCM		LUAD	
TTN	6	TP53	17	TTN	19	TP53	19	BAP1	19	TP53	10	PCDHGC5	10	ANK2	4
COL11A1	3	CACNA2D1	1	BRCA2	1	TTN	1	PBRM1	1	IDH1	6	FLNC	5	RYR2	4
FAM5C	3	MUC16	1							ITSN2	2	COL3A1	2	CRB1	4
PCDHAC2	3	NEB	1							PLEC	1	PCDHB5	1	TP53	3
ANK2	3									EDA	1	SCN11A	1	LAMA2	2
TP53	1											KIAA1217	1	HMCN1	1
RP1	1													USH2A	1
														LARP1	1

Table A.1 – Summary of the genes selected when only one gene is used to predict survival. For each gene the number of folds (out of 20 folds) where the gene is selected is indicated.

Cancer type	*min*	$Q1$	*median*	$Q3$	*max*
LUAD (Lung adenocarcinoma)	4	85	189	344	1322
SKCM (Skin cutaneous melanoma)	7	112	243	451	6272
GBM (Glioblastoma multiform)	3	31	44	130	5562
BRCA (Breast invasive carcinoma)	1	19	28	52	3189
KIRC (Kidney renal clear cell carcinoma)	7	33	43	55	108
HNSC (Head and Neck squamous cell carcinoma)	5	64	99	144	2504
LUSC (Lung squamous cell carcinoma)	2	143	189	257	1801
OV (Ovarian serous cystadenocarcinoma)	1	24	38	53	140

Table A.2 – Statistics of the distributions of patients' total number of mutations for each cancer. Only mutations in genes present in Pathway Commons are taken into account. $Q1$ and $Q3$ refer to the 1^{st} and 3^{rd} quartiles respectively. The parameter k (NetNorM) was learned by cross-validation in the supervised setting using cancer specific cross-validation grids delimited by $Q1$ and $Q3$, and with a step-size of 2.

Cancer type	k (NetNorM)	α (NSQN)
LUAD (Lung adenocarcinoma)	315	0.6
SKCM (Skin cutaneous melanoma)	140	0.25
GBM (Glioblastoma multiform)	49	0.8
BRCA (Breast invasive carcinoma)	25	0.45
KIRC (Kidney renal clear cell carcinoma)	51	0.75
HNSC (Head and Neck squamous cell carcinoma)	70	0.2
LUSC (Lung squamous cell carcinoma)	199	0.35
OV (Ovarian serous cystadenocarcinoma)	32	0.75

Table A.3 – Summary of the values of k (NetNorM) and α (NS and NSQN) learned by cross-validation for survival prediction. The values given are the medians obtained over 20 cross-validation folds performed for each dataset and each method.

gene cluster	color	nb. of genes	subgraph density	proxy mutations fraction	χ^2 test P value	enriched KEGG pathways
1	green	41	0.963	0.94	6×10^{-5}	Endocytosis, Ribosome, Phagosome
2	brown	39	1	0.974	9×10^{-15}	Spliceosome, Ribosome, RNA transport
3	cyan	27	0.983	0.945	5×10^{-8}	Ribosome, Spliceosome
4	magenta	12	0.955	0.91	1×10^{-12}	Epstein-Barr virus infection, Ribosome, Endocytosis
5	yellow	38	1	0.977	4×10^{-18}	Spliceosome, Ribosome
6	back	1	-	0.93	1×10^{-33}	Spliceosome, Ribosome, RNA transport
7	blue	80	0.989	0.935	1×10^{-8}	Ribosome, Spliceosome
8	green	143	0.679	0.847	2×10^{-2}	Salmonella infection, Prostate cancer, Estrogen signaling pathway
9	brown	185	0.03	0.093	0	cAMP Signaling pathway, PI3K-Akt signaling pathway
10	cyan	256	0.06	0.04	4×10^{-35}	Olfactory transduction, Amoebiasis
11	magenta	4	0	0.003	2×10^{-12}	[USH2A, ZFHX4, CSMD3, KRAS]
12	yellow	31	0.103	0.012	7×10^{-2}	-
13	blue	12	0.06	0.093	2×10^{-169}	-
14	green	10	0.111	0.005	4×10^{-1}	-
15	black	1	-	0	1×10^{-2}	[MUC16]
16	brown	11	1	0.956	1×10^{-9}	Endocytosis, Phagosome
17	cyan	34	1	0.971	3×10^{-18}	Ribosome, Spliceosome
18	magenta	29	0.983	0.978	5×10^{-6}	Ribosome, Spliceosome
19	yellow	2	0	0.166	8×10^{-13}	[TP53, TTN]
20	blue	2	0	0	3×10^{-1}	[LRP1B, RYR2]

Table A.4 – The gene clusters characterising LUAD patient subtypes obtained with NetNorM ($N = 5$ groups, Pathway Commons). *nb. of genes*: number of genes in a cluster, *subgraph density*: density of the subgraph whose vertices are the genes inside a cluster, *proxy mutations fraction*: number of proxy mutations out the the total number of mutations for a gene cluster across all patients.

Appendix B

Supplementaries for WHInter

B.1 Proof of Lemma 4.3.1

We detail here the proof of Lemma 4.3.1.

Proof. With the notations of Lemma 4.3.1 , we have:

$$\begin{aligned}
\max_{k\in\mathcal{I}} \left| \boldsymbol{\theta}_2^\top (\mathbf{v}\odot\mathbf{X}_k) \right| &\leq \max_{k\in\mathcal{I}} \left| \alpha\boldsymbol{\theta}_1^\top (\mathbf{v}\odot\mathbf{X}_k) + (\boldsymbol{\theta}_2-\alpha\boldsymbol{\theta}_1)^\top (\mathbf{v}\odot\mathbf{X}_k) \right| \\
&\leq |\,\alpha\,| \max_{k\in\mathcal{I}} \left| \boldsymbol{\theta}_1^\top (\mathbf{v}\odot\mathbf{X}_k) \right| + \max_{k\in\mathcal{I}} \left| (\boldsymbol{\theta}_2-\alpha\boldsymbol{\theta}_1)^\top (\mathbf{v}\odot\mathbf{X}_k) \right| \\
&\leq |\,\alpha\,| \max_{k\in\mathcal{I}} \left| \boldsymbol{\theta}_1^\top (\mathbf{v}\odot\mathbf{X}_k) \right| + \max_{\boldsymbol{x}\in\{0,1\}^n} \left| (\boldsymbol{\theta}_2-\alpha\boldsymbol{\theta}_1)^\top (\mathbf{v}\odot\boldsymbol{x}) \right| \\
&= |\,\alpha\,| \max_{k\in\mathcal{I}} \left| \boldsymbol{\theta}_1^\top (\mathbf{v}\odot\mathbf{X}_k) \right| + \zeta(\boldsymbol{\theta}_2-\alpha\boldsymbol{\theta}_1, \mathbf{v})\,.
\end{aligned}$$

□

B.2 Computing η_{min}

In this section we characterise the existence and the possibility to compute, for any fixed $(\mathbf{v}, \boldsymbol{\theta}, \boldsymbol{\theta}', m) \in \mathbb{R}_+^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}$:

$$\eta_{min}\left(\mathbf{v}, \boldsymbol{\theta}', m\right) := \min_{\alpha\in\mathbb{R}} \eta_\alpha\left(\mathbf{v}, \boldsymbol{\theta}, \boldsymbol{\theta}', m\right)\,, \tag{B.1}$$

where η_α is defined in Section 4.3.2. For that purpose, let us introduce for any $\alpha \in \mathbb{R}$ the functions:

$$\begin{cases}
\gamma_p(\alpha) &= \displaystyle\sum_{i:\boldsymbol{\theta}'_i-\alpha\boldsymbol{\theta}_i>0} \mathbf{v}_i\left(\boldsymbol{\theta}'_i-\alpha\boldsymbol{\theta}_i\right), \\
\gamma_m(\alpha) &= \displaystyle\sum_{i:\boldsymbol{\theta}'_i-\alpha\boldsymbol{\theta}_i<0} \mathbf{v}_i\left(\boldsymbol{\theta}'_i-\alpha\boldsymbol{\theta}_i\right),
\end{cases}$$

such that:

$$\eta_\alpha\left(\mathbf{v}, \boldsymbol{\theta}, \boldsymbol{\theta}', m\right) = |\alpha|\, m + \max\left(\gamma_p(\alpha), -\gamma_m(\alpha)\right)\,. \tag{B.2}$$

Let us first characterise the existence and properties of the solution to the minimisation problem (B.1).

Theorem B.2.1. *For any* $(\mathbf{v}, \boldsymbol{\theta}, \boldsymbol{\theta}', m) \in \mathbb{R}_+^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}$*, the function*

$$\alpha \in \mathbb{R} \to \eta_\alpha \left(\mathbf{v}, \boldsymbol{\theta}, \boldsymbol{\theta}', m\right)$$

is continuous, piecewise affine, convex and nonnegative. It reaches at least a minimum at a value $\alpha^* \in \mathcal{B}$ *where*

$$\mathcal{B} = \{0\} \cup \left\{ \frac{\boldsymbol{\theta}'_i}{\boldsymbol{\theta}_i} \,:\, i \in \mathit{supp}(\boldsymbol{\theta}) \cap \mathit{supp}(\mathbf{v}) \right\} \cup \{\alpha \in \mathbb{R} : \gamma_p(\alpha) = \gamma_m(\alpha)\} \,.$$

Proof. For any $i \in [\![n]\!]$, let

$$\forall \alpha \in \mathbb{R}\,, \quad \phi_i(\alpha) = \mathbf{v}_i \max\left(0, \boldsymbol{\theta}'_i - \alpha \boldsymbol{\theta}_i\right) \,.$$

Since $\mathbf{v}_i \geq 0$, $\phi_i(\alpha) = \mathbf{v}_i \max\left(0, \boldsymbol{\theta}'_i - \alpha\boldsymbol{\theta}_i\right)$ is continuous, piecewise affine, convex and nonnegative. It has a single breakpoint at $\alpha_i = \boldsymbol{\theta}'_i/\boldsymbol{\theta}_i$ if $\boldsymbol{\theta}_i \neq 0$ and $\mathbf{v}_i > 0$, and is constant otherwise. Since $\gamma_p(\alpha) = \sum_{i=1}^n \phi_i(\alpha)$, γ_p is also continuous, piecewise affine, convex and nonnegative with breakpoints in $\{\boldsymbol{\theta}'_i/\boldsymbol{\theta}_i \,:\, i \in \operatorname{supp}(\boldsymbol{\theta}) \cup \operatorname{supp}(\mathbf{v})\}$. Taking $\psi_i(\alpha) = \mathbf{v}_i \max\left(0, \alpha\boldsymbol{\theta}_i - \boldsymbol{\theta}'_i\right)$ shows similarly that $-\gamma_m(\alpha) = \sum_{i=1}^n \psi_i(\alpha)$ has the same properties. Consequently, the function $\alpha \mapsto \max\left(\gamma_p(\alpha), -\gamma_m(\alpha)\right)$ is also continuous, piecewise affine, convex and nonnegative, with possible breakpoints in

$$\left\{\boldsymbol{\theta}'_i/\boldsymbol{\theta}_i \,:\, i \in \operatorname{supp}(\boldsymbol{\theta}) \cup \operatorname{supp}(\mathbf{v})\right\} \cup \{\alpha \in \mathbb{R} : \gamma_p(\alpha) = \gamma_m(\alpha)\} \,.$$

Since $\alpha \mapsto |\,\alpha\,|$ is also continuous, piecewise affine, convex and nonnegative, and has a breakpoint for $\alpha = 0$, Theorem B.2.1 follows by observing that a continuous, piecewise affine, convex and nonnegative function necessarily reaches a minimum at one of its breakpoints. □

Let $S = |\operatorname{supp}(\boldsymbol{\theta}) \cap \operatorname{supp}(\mathbf{v})|$. Theorem B.2.1 shows that it suffices to compute the values of η_α on at most $S + 2$ values for α to find the global minimum. However, a naive computation of η_α using (B.2) takes $O(|\operatorname{supp}(\mathbf{v})|)$ for each α, hence a total complexity $O(S \times |\operatorname{supp}(\mathbf{v})|)$ to find the minimum of η_α.

This can be improved to $O(|\operatorname{supp}(\mathbf{v})| + S \ln S)$ by first sorting the $S + 1$ breakpoints $b_i = \boldsymbol{\theta}'_i/\boldsymbol{\theta}_i$ for $i \in \operatorname{supp}(\boldsymbol{\theta}) \cap \operatorname{supp}(\mathbf{v})$ and $b_{S+1} = 0$ in increasing order:

$$b_{\pi(1)} \leq b_{\pi(2)} \leq \ldots \leq b_{\pi(S+1)} \,,$$

which takes $O(S \ln S)$ time. Adding by convention $b_{\pi(0)} = -\infty$ we observe that on each interval $(b_{k-1}, b_k]$ the functions γ_p and γ_m are affine, of the form:

$$\forall \alpha \in (b_{k-1}, b_k]\,, \quad \begin{cases} \gamma_p(\alpha) & = s_p^k - \alpha t_p^k \,, \\ -\gamma_m(\alpha) & = s_m^k - \alpha t_m^k \,. \end{cases}$$

From the properties of $\gamma_p(\alpha) = \sum_{i=1}^n \phi_i(\alpha)$ and $-\gamma_m(\alpha) = \sum_{i=1}^n \psi_i(\alpha)$, we get the coefficients for $k = 1$, i.e., for the interval $(-\infty, b_{\pi(1)}]$ in $O(|\operatorname{supp}(\mathbf{v})|)$ as follows:

$$\begin{cases} s_p^1 & = \sum_{i \in \operatorname{supp}(\mathbf{v}) \,:\, \boldsymbol{\theta}_i > 0} \mathbf{v}_i \boldsymbol{\theta}'_i + \sum_{i \in \operatorname{supp}(\mathbf{v}) \,:\, \boldsymbol{\theta}_i = 0} \mathbf{v}_i \max(0, \boldsymbol{\theta}'_i) \,, \\ t_p^1 & = \sum_{i \in \operatorname{supp}(\mathbf{v}) \,:\, \boldsymbol{\theta}_i > 0} \mathbf{v}_i \boldsymbol{\theta}_i \,, \\ s_m^1 & = -\sum_{i \in \operatorname{supp}(\mathbf{v}) \,:\, \boldsymbol{\theta}_i < 0} \mathbf{v}_i \boldsymbol{\theta}'_i + \sum_{i \in \operatorname{supp}(\mathbf{v}) \,:\, \boldsymbol{\theta}_i = 0} \mathbf{v}_i \max(0, -\boldsymbol{\theta}'_i) \,, \\ t_m^1 & = \sum_{i \in \operatorname{supp}(\mathbf{v}) \,:\, \boldsymbol{\theta}_i < 0} \mathbf{v}_i \boldsymbol{\theta}_i \,. \end{cases} \tag{B.3}$$

This allows in particular to compute $\gamma_p(b_{\pi(1)})$, $\gamma_m(b_{\pi(1)})$, and therefore $\eta_{b_{\pi(1)}}$ from (B.2). We can then iteratively compute the coefficients for $k+1$ from the coefficients for k in $O(1)$ only, by observing that between the intervals $(b_{k-1}, b_k]$ and $(b_k, b_{k+1}]$, the only change in slope and intercept of γ_p is due to the function $\phi_{\pi^{-1}(k)}$, when $\pi^{-1}(k) \neq S+1$. Let $i = \pi^{-1}(k)$. When $\boldsymbol{\theta}_i > 0$, the slope of ϕ_i increases by $\mathbf{v}_i\boldsymbol{\theta}_i$ and its intercept decreases by $\mathbf{v}_i\boldsymbol{\theta}'_i$ at b_i. When $\boldsymbol{\theta}_i > 0$, its slope increases by $-\mathbf{v}_i\boldsymbol{\theta}_i$ and its intercept increases by $\mathbf{v}_i\boldsymbol{\theta}'_i$. This translates into the following recursive formula for the coefficients of γ_p:

$$s_p^{k+1} = \begin{cases} s_p^k - \mathbf{v}_i\boldsymbol{\theta}'_i & \text{if } \boldsymbol{\theta}_i > 0\,, \\ s_p^k + \mathbf{v}_i\boldsymbol{\theta}'_i & \text{if } \boldsymbol{\theta}_i < 0\,, \end{cases}$$

and

$$t_p^{k+1} = t_p^k - \mathbf{v}_i \,|\, \boldsymbol{\theta}_i \,|\,.$$

A similar analysis on γ_m leads to the following recursion:

$$s_m^{k+1} = \begin{cases} s_m^k - \mathbf{v}_i\boldsymbol{\theta}'_i & \text{if } \boldsymbol{\theta}_i > 0\,, \\ s_m^k + \mathbf{v}_i\boldsymbol{\theta}'_i & \text{if } \boldsymbol{\theta}_i < 0\,, \end{cases}$$

and

$$t_m^{k+1} = t_m^k - \mathbf{v}_i \,|\, \boldsymbol{\theta}_i \,|\,.$$

We can thus iteratively compute the coefficients on each interval, and thus the values of η_α on each breakpoint, with complexity $O(1)$ per breakpoint. Since $\alpha \mapsto \eta_\alpha$ is convex, we stop at the first k such that $\eta_{b_{\pi(k+1)}} \geq \eta_{b_{\pi(k)}}$. From the equations of γ_p and γ_m on $(b_{\pi(k)}, b_{\pi(k+1)}]$ we can additionally check if there is a crossing point $\bar{\alpha} \in (b_{\pi(k)}, b_{\pi(k+1)}]$ such that $\gamma_p(\bar{\alpha}) = \gamma_m(\bar{\alpha})$, in which case we also compute $\eta_{\bar{\alpha}}$. The global minimum of $\alpha \mapsto \eta_\alpha$ is then $\min(\eta_{b_{\pi(k)}}, \eta_{\bar{\alpha}})$.

The overall algorithm is detailed in Algorithm B.1.

Algorithm B.1 Minimise η in α

Input: $(\mathbf{v}, \boldsymbol{\theta}, \boldsymbol{\theta}', m) \in \mathbb{R}^n_+ \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}$.
Output: $\eta_{min}(\mathbf{v}, \boldsymbol{\theta}, \boldsymbol{\theta}', m)$

$S \leftarrow$ indices in $\text{supp}(\mathbf{v}) \cap \text{supp}(\boldsymbol{\theta})$
$N \leftarrow \text{length}(S)$
$b \leftarrow \left[0, \frac{\boldsymbol{\theta}'_{S[1]}}{\boldsymbol{\theta}_{S[1]}}, \ldots, \frac{\boldsymbol{\theta}'_{S[N]}}{\boldsymbol{\theta}_{S[N]}}\right]$
$\texttt{ind} \leftarrow [\texttt{none}, S[1], \ldots, S[N]]$
$\texttt{rank} \leftarrow \text{sort}(b)$ (in increasing order)
$b \leftarrow b[\texttt{rank}]$; $\texttt{ind} \leftarrow \texttt{ind}[\texttt{rank}]$
Initialise s_p, s_m, t_p, t_m via (B.3)
$\texttt{min} \leftarrow +\infty$
for i in $1 \ldots N+1$ **do**
  $\texttt{newmin} \leftarrow |b[i]|\, m + \max\left(s_p - b[i]t_p,\ s_m - b[i]t_m\right)$
  if $\texttt{newmin} < \texttt{min}$ **then**
    $\texttt{min} \leftarrow \texttt{newmin}$
    if $\texttt{ind}[i] \neq \texttt{none}$ **then**
      $t_p \leftarrow t_p - \mathbf{v}_{\texttt{ind}[i]} \left| \boldsymbol{\theta}_{\texttt{ind}[i]} \right|$
      $t_m \leftarrow t_m - \mathbf{v}_{\texttt{ind}[i]} \left| \boldsymbol{\theta}_{\texttt{ind}[i]} \right|$
      if $\boldsymbol{\theta}_{\texttt{ind}[i]} > 0$ **then**
        $s_p \leftarrow s_p - \mathbf{v}_{\texttt{ind}[i]} \boldsymbol{\theta}'_{\texttt{ind}[i]}$
        $s_m \leftarrow s_m - \mathbf{v}_{\texttt{ind}[i]} \boldsymbol{\theta}'_{\texttt{ind}[i]}$
      else
        $s_p \leftarrow s_p + \mathbf{v}_{\texttt{ind}[i]} \boldsymbol{\theta}'_{\texttt{ind}[i]}$
        $s_m \leftarrow s_m + \mathbf{v}_{\texttt{ind}[i]} \boldsymbol{\theta}'_{\texttt{ind}[i]}$
      end if
    end if
  else
    Check if there exists $\bar{\alpha} \in [b[i-1], b[i]]$ s.t. $\gamma_p(\bar{\alpha}) = \gamma_m(\bar{\alpha})$
    Return min($\texttt{newmin}$, $\eta(\alpha^{intersection})$)
  end if
end for

B.3 Alternative solver for working set updates

In this section, we present an alternative solver to the inverted list approach (Algorithm 4.3 in Section 4.3.3), which we call $MIPS1$, to compute the working set updates (4.12). It relies on a pruning technique and does not require storing extra indices for the data. The main idea of this alternative approach is to compute inner products on a progressively growing subset of dimensions, and to maintain an upper-bound on the maximum attainable score on the remaining dimensions. This allows to discard a probe as soon as its maximum attainable score drops below the maximum score achieved so far without computing the inner product in its entirety. Algorithm B.2 presents the procedure in details. It takes as input $\mathcal{Q}$ which contains the indices that define the queries of interest and outputs the updated working set $\mathcal{W}$ and $\mathbf{m}^{ref}$. For each query, we start by precomputing the partial inner product bounds $\mathbf{r}^+ \in \mathbb{R}^n$ and $\mathbf{r}^- \in \mathbb{R}^n$, where $\mathbf{r}_i^+$ and $\mathbf{r}_i^-$ are respectively the maximum and minimum attainable inner products between the query and any probe in the database on the dimensions from $i+1$ to n. Formally, $\mathbf{r}^+$ and $\mathbf{r}^-$ are defined for a given query j by:

$$\forall i \in [\![n]\!], \mathbf{r}_i^+ = \sum_{m>i;\ \boldsymbol{\theta}_m>0} \mathbf{X}_{mj}\boldsymbol{\theta}_m \tag{B.4}$$

$$\forall i \in [\![n]\!], \mathbf{r}_i^- = \sum_{m>i;\ \boldsymbol{\theta}_m<0} \mathbf{X}_{mj}\boldsymbol{\theta}_m \tag{B.5}$$

and provide an upper bound on inner products with the query $\mathbf{X}_j \odot \boldsymbol{\theta}$ as follows:

$$\begin{aligned}
\forall k \in [\![p]\!], \quad (\mathbf{X}_j \odot \boldsymbol{\theta})^\top \mathbf{X}_k &= \sum_{m\leq i} \mathbf{X}_{mj}\boldsymbol{\theta}_m\mathbf{X}_{mk} + \sum_{m>i} \mathbf{X}_{mj}\boldsymbol{\theta}_m\mathbf{X}_{mk} \\
&\leq \sum_{m\leq i} \mathbf{X}_{mj}\boldsymbol{\theta}_m\mathbf{X}_{mk} + \sum_{m>i;\ \boldsymbol{\theta}_m>0} \mathbf{X}_{mj}\boldsymbol{\theta}_m \\
&= \sum_{m\leq i} \mathbf{X}_{mj}\boldsymbol{\theta}_m\mathbf{X}_{mk} + \mathbf{r}_i^+
\end{aligned}$$

The bound involving $\mathbf{r}^-$ can be obtained analogously. These bounds simply assume there is a probe vector which has ones in front of every positive entry of the query and none in front of its negative entries, or the reverse. Once these bounds have been precomputed, the inner product between the query and a probe is computed up to a certain dimension, and every $n_c \in \mathbb{N}$ dimensions we check whether there is a possibility that the inner product being computed becomes larger than the current maximum, or larger than λ. If it is impossible, then the probe can be safely discarded and the algorithm proceeds with the next probe. If not, the inner product is computed on n_c more dimensions and a new check is performed. For all our simulations and real data experiments, we set n_c to a default of 20. If a probe cannot be discarded then the algorithm updates when appropriate the active set $\mathcal{W}$ and/or the current maximum absolute inner product obtained $\mathbf{m}_j^{ref}$. For the pruning to be effective, we reorder the dimensions $1 \dots n$ so that queries are sorted in decreasing order in absolute value. As a consequence, the partial inner product bounds $\mathbf{r}_i^+$ and $\mathbf{r}_i^-$ are computed with the $n-i$ smallest entries in absolute value of the queries which makes them tighter than with any other ordering of the dimensions.

Algorithm B.2 MIPS1

Input: $\mathbf{X} \in [0,1]^{n \times p}$, $\boldsymbol{\theta} \in \mathbb{R}^n$, $\mathcal{Q} \subset [\![p]\!]$, $\lambda \in \mathbb{R}$, $\mathcal{W} \subset [\![D]\!]$
Param: $n_c \in \mathbb{N}$
Output: $\mathcal{W}$, $\mathbf{m}^{ref}$.

Reorder the dimensions $1 \dots n$ such that $\boldsymbol{\theta}$ is sorted in descending order in absolute value and reorder the dimensions of $\mathbf{X}$ accordingly.
Reorder the columns of $\mathbf{X}$ in descending order of vector size.
for $j \in \mathcal{Q}$ **do** $\mathbf{m}_j^{ref} \leftarrow 0$
for $j \in \mathcal{Q}$ **do**
  Compute $\mathbf{r}^+ \in \mathbb{R}^n$ and $\mathbf{r}^- \in \mathbb{R}^n$ via (B.4) and (B.5).
  for $k \in [\![p]\!]$ **do**
    if $k \in \mathcal{Q}$ and $k > j$ then `continue`
    $d \leftarrow 0$ (inner product initialization); c $= 0$ (counter initialization);
    for $i \in \text{supp}(\boldsymbol{X}_j)$ **do**
      $d \leftarrow d + \mathbf{X}_{ij}\mathbf{X}_{ik}\boldsymbol{\theta}_i$
      $c \leftarrow c + 1$.
      if c `mod` $n_c = 0$ **then**
        if $\left|(d + \mathbf{r}_i^+)\right| < \min(\mathbf{m}_j^{ref}, \lambda)$ `and` $\left|(d + \mathbf{r}_i^-)\right| < \min(\mathbf{m}_j^{ref}, \lambda)$ **then** go to next probe.
      end if
    end for
    if $\mathbf{m}_j^{ref} < |\, d \,| < \lambda$ **then** set $\mathbf{m}_j^{ref} = |\, d \,|$
    if $|\, d \,| \geq \lambda$ and $\tau(k,j) \notin \mathcal{W}$ **then** add $\tau(k,j)$ to $\mathcal{W}$
  end for
end forreturn $\mathcal{W}, \boldsymbol{m}^{ref}$

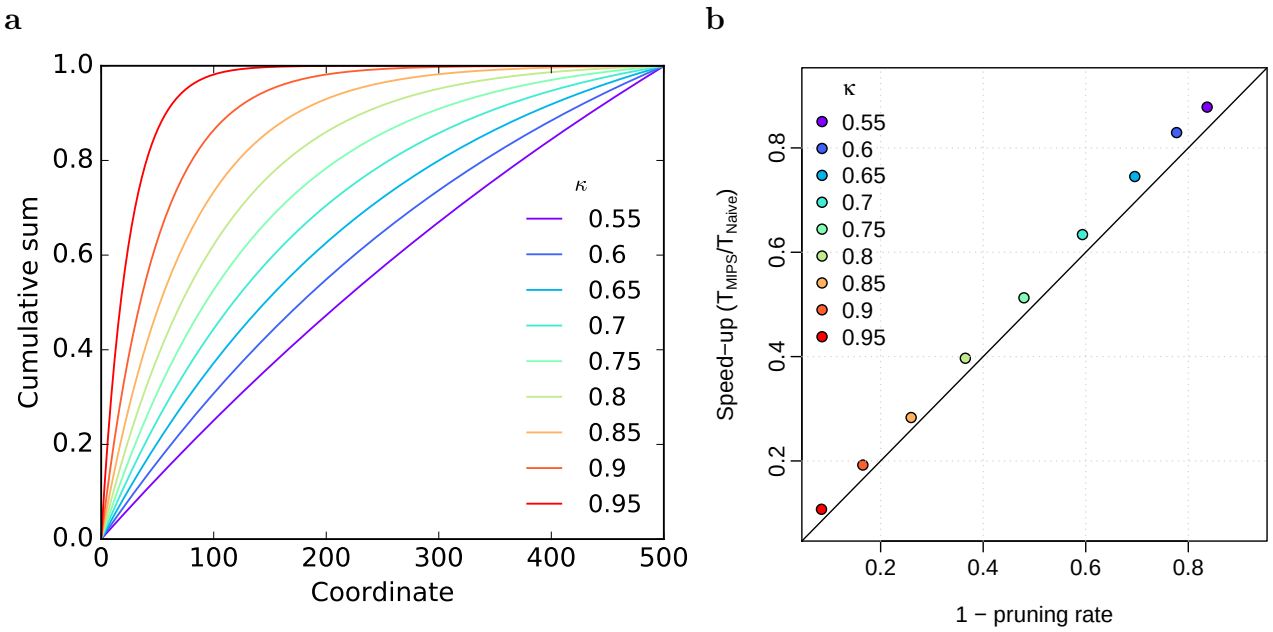


Figure B.1 – Performances of MIPS1 on simulated data. (a) Cumulative sum of the vector obtained by sorting the positive entries of $\boldsymbol{\theta}_\kappa$ in decreasing order. (b) Speed-up obtained with $MIPS1$ compared to $Naive$ for different vectors $\boldsymbol{\theta}_\kappa$ as a function of the pruning rate. The pruning rate is defined as the average proportion of coordinates in the queries which are pruned.

We now compare $MIPS1$ to its naive counterpart (which we will call $Naive$ from now on) on several benchmark datasets in order to assess the speed-up obtained with the pruning. To be more specific, $Naive$ is implemented similarly to $MIPS1$ except the lines specific to pruning, i.e., lines 5, 12 and 13 in Algorithm B.2, are removed. The benchmark datasets we use are designed in such a way that the pruning rate achievable varies. To do this, we simulate a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, with $n = p = 1000$, where the features are drawn from a Bernoulli distribution, whose parameter is itself drawn from a uniform distribution $\mathcal{U}_{[0.1,0.5]}$. Then $\boldsymbol{\theta} \in \mathbb{R}^n$ is built in such a way that the cumulative sum of the vectors obtained by sorting $\boldsymbol{\theta}_{|\boldsymbol{\theta} \geq 0}$ and $|\boldsymbol{\theta}_{|\boldsymbol{\theta} < 0}|$ follows the function $f(x) = \frac{1}{1-e^{-\mu}} \left(1 - e^{-\mu x}\right), x \in \{0, 1\}$ for a given parameter $\mu \in \mathbb{R}^+$. The area under this cumulative sum, which is $\kappa(\mu) = \frac{1}{1-e^{-\mu}} - \frac{1}{\mu} \in [0.5, 1]$, characterises the different vectors $\boldsymbol{\theta}_\kappa$ obtained with different values of μ. Figure B.1a shows how the cumulative sums are modified with μ. The interest of simulating different $\boldsymbol{\theta}_\kappa$ is that the rate of pruning achievable increases with κ: the closer κ is to 1, the higher the pruning rate. In the experiments presented hereafter, all p features were taken as queries, i.e., $\mathcal{Q} = [\![p]\!]$, and we took $\lambda = +\infty$ and $\mathcal{W} = \emptyset$. The results are presented in Fig. B.1b. The pruning rate, which we define as the average number of non-zero coordinates of the queries which were pruned out of their total number of non-zero coordinates, widely varies from 8% for $\kappa = 0.55$ to 84% for $\kappa = 0.95$. Moreover, the speed-up obtained with $MIPS1$ compared to $Naive$ is almost equal to 1 minus the pruning rate. That means $MIPS1$ is twice as fast as $Naive$ when it can prune half of the total number of coordinates.

We now compare the performance of $Naive$, $MIPS1$ and IL on the benchmark datasets (Fig. B.2). $MIPS1$ is the only method whose speed depends on κ since it is the only method

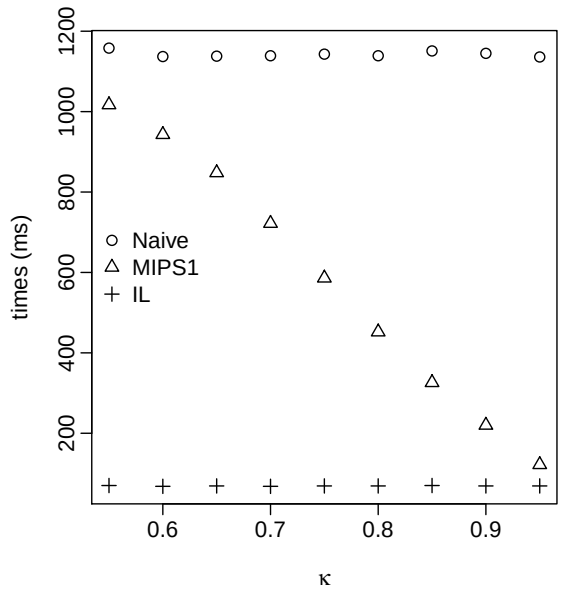


Figure B.2 – Performance comparisons on simulated data. Time (in ms) taken by $Naive$, $MIPS1$ and IL to solve Maximum Inner Product Search problems with responses characterised by different κ.

to implement pruning. It has the same performance in terms of speed as $Naive$ for the lowest pruning rate, while it is as fast as IL for the highest pruning rates. For vectors $\boldsymbol{\theta}$ following classical distributions such as the gaussian distribution, $\kappa \approx 0.7$ and $MIPS1$ is therefore expected to be $\times 1.6$ times faster than $Naive$ but $\times 11$ times slower than IL. An analysis of the complexity of $MIPS1$ and IL can help to understand these results. For a given query, $MIPS1$ requires to compute inner products (although partially) with all p vectors in the database. In our implementation, the vectors are encoded as sparse vectors, i.e., the vector $\mathbf{X}_j$ is represented by the list of its non-zero indices. If we assume that the number of non-zero elements in the query is $|q|$ and that the total number of non-zero elements of the vectors in $\mathbf{X}$ in nnz, then $MIPS1$ has a $O(p|q| + nnz)$ complexity to compute the p inner products with the query. By contrast, the inverted index approach has a $O(|q|\frac{nnz}{n})$ complexity, where $\frac{nnz}{n}$ is the average length of an inverted index. As the number of non-zero elements $|q|$ in the query will typically be a fraction of the total number of samples n, the inverted index approach is expected to be faster than $MIPS1$ even though the pruning in $MIPS1$ can make it faster. This however may not be the case with dense data instead of sparse data.

B.4 SPP: depth-first vs breadth-first

The Safe Pattern Pruning algorithm presented in [Nakagawa et al., 2016] deals with pairwise interactions but also higher-order interactions, and relies on a depth-first search scheme to explore the tree of patterns. However in our setting where we only consider pairwise interactions, we find that it is more efficient to implement a breadth-first search scheme for SPP. Indeed, the breadth-first search first identifies all the branches which can be screened. Then with this knowledge, we can restrict the number of interactions which are visited to those which only involve main effects whose corresponding branch was not screened. Basically, if we consider a

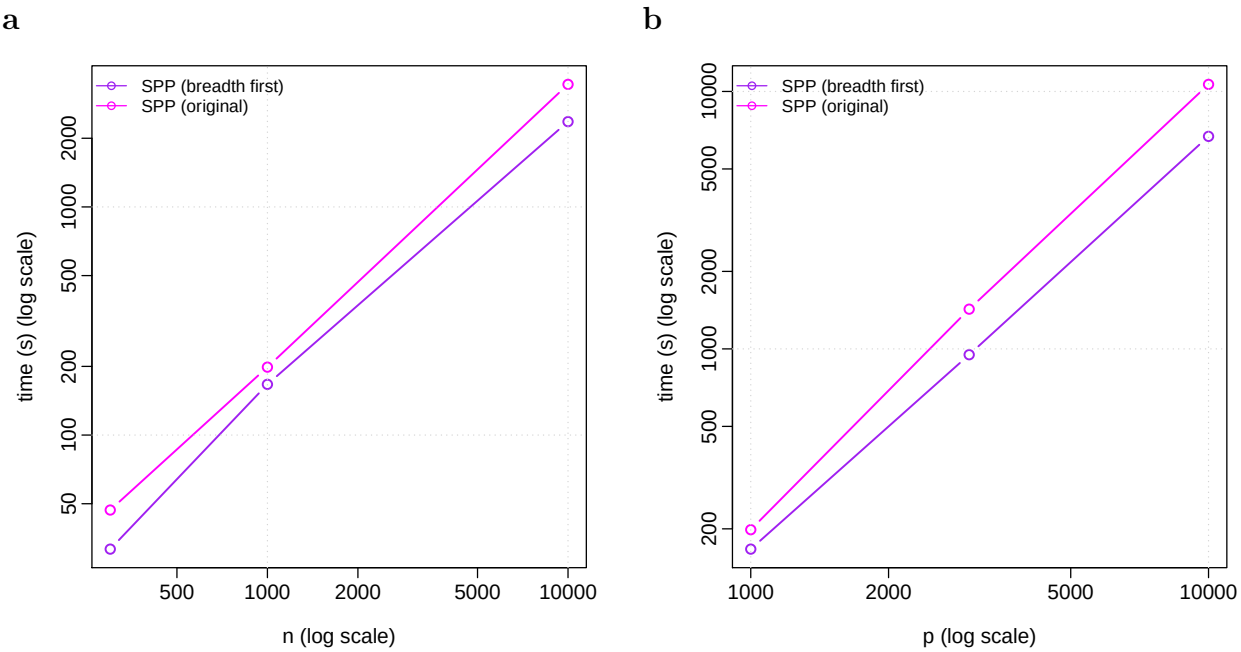


Figure B.3 – Safe Pattern Pruning performance on simulated data for an entire regularisation path. The breadth-first search SPP (which is adapted to order-2 interactions only) is in purple and the original depth-first search SPP (which is adapted to order-2 interactions and more) is in magenta. (a) Time in seconds for $p = 1000$ fixed and n varied. (b) Time in seconds for $n = 1000$ fixed and p varied.

case where p_s branches were screened among p branches, then the total number of nodes visited will be $p + \frac{(p-p_s)(p-p_s-1)}{2}$. Figure B.3 illustrates the difference in performance obtained with the original SPP and the breadth-first search version in the case of pairwise interactions. The speed up obtained with the breadth-first search version ranges from $\times 1.2$ for $n = p = 1000$ to $\times 1.6$ for $n = 1000, p = 10000$. We therefore use the breadth-first search version of SPP as a comparison baseline in all our experiments.

B.5 Suppplementary figures

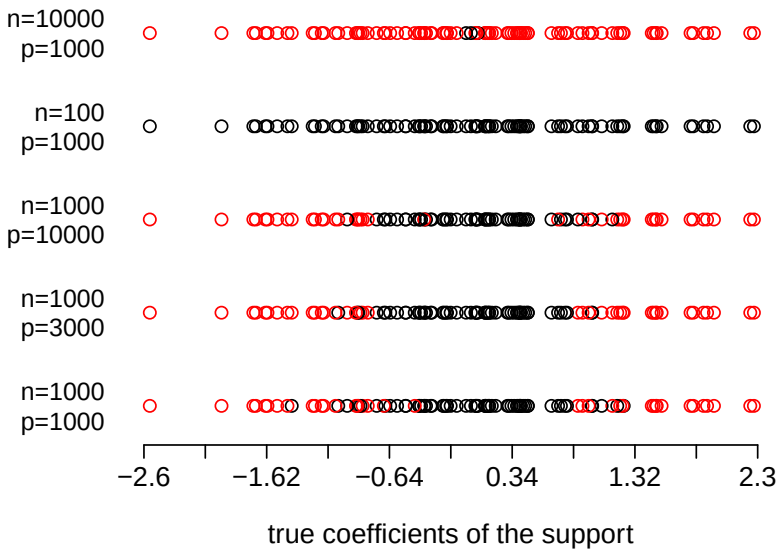


Figure B.4 – Support recovery depending on the number of samples and main effects. Each line corresponds to a different simulation with number of samples n and number of main effects p. Red circles indicates that the feature with true coefficient indicated on the x-axis has been selected in the support whereas black points indicate that it has not been selected in the support. The support shown is the one obtained for the smallest value of λ tested, which is the first one for which 150 features or more have been selected. We recall that all methods returned the exact same support for all values of λ.

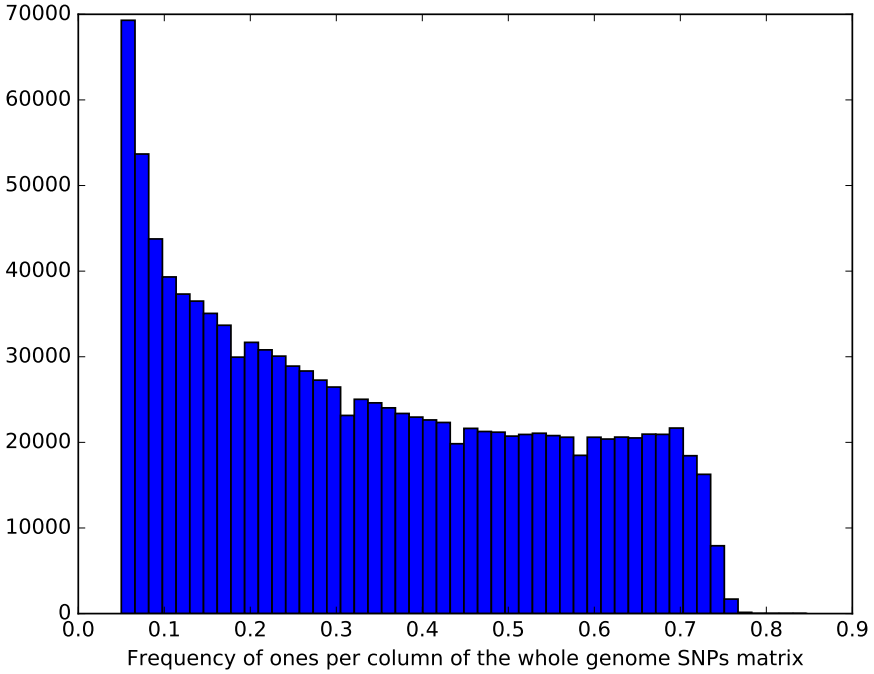


Figure B.5 – Whole genome SNPs matrix sparsity Frequency at which a minor allele is present in one of both copies of the chromosomes for each SNP of the whole genome. This corresponds to the frequency of ones in the columns of the whole genome SNPs matrix.

Bibliography

L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, S. A. Aparicio, S. Behjati, A. V. Biankin, G. R. Bignell, N. Bolli, A. Borg et al. Signatures of mutational processes in human cancer. *Nature*, 500(7463):415–421, 2013. 7

D. Amaratunga and J. Cabrera. Analysis of Data From Viral DNA Microchips. *J. Am. Stat. Assoc.*, 96(456):1161–1170, 2001. 61

S. A. Andres, G. N. Brock and J. L. Wittliff. Interrogating differences in expression of targeted gene sets to predict breast cancer outcome. *BMC Cancer*, 13(1):326, 2013. 41

S. L. Anzick. AIB1, a Steroid Receptor Coactivator Amplified in Breast and Ovarian Cancer. *Science (80-.).*, 277(5328):965–968, 1997. 41

N. Aronszajn. Theory of Reproducing Kernels. *Trans. Am. Math. Soc.*, 68(3):337, 1950. 17

D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proc. 18th Annu. ACM-SIAM Symp. Discret. algorithms*, pages 1027–1035, 2007. 56

E. A. Ashley. Towards precision medicine, 2016. 5

P. L. Auer and G. Lettre. Rare variant association studies: Considerations, challenges and opportunities. *Genome Med.*, 7(1):16, 2015. 11

S. Babaei, M. Hulsman, M. Reinders and J. de Ridder. Detecting recurrent gene mutation in interaction network context using multi-scale graph diffusion. *BMC Bioinformatics*, 14(1):29, 2013. 33

Y. Bachrach, Y. Finkelstein, R. Gilad-Bachrach, L. Katzir, N. Koenigstein, N. Nice and U. Paquet. Speeding up the Xbox recommender system using a euclidean transformation for inner-product spaces. In *Proc. 8th ACM Conf. Recomm. Syst.*, pages 257–264, 2014. 95

E. Barillot, L. Calzone, P. Hupé, J.-P. Vert and A. Zinovyev. *Computational systems biology of cancer*. CRC Press, 2012. 33

R. E. Barlow, D. Bartholomew, J. M. Bremner and H. D. Brunk. *Statistical inference under order restrictions; the theory and application of isotonic regression*. Wiley, New-York, 1972. 67

T. Barrett, D. B. Troup, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy et al. NCBI GEO: archive for functional genomics data sets - 10 years on. *Nucleic Acids Res.*, 39(suppl 1):D1005—-D1010, 2011. 71

A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009. 24

R. Bellman. *Adaptive control processes: A guided tour*. Princeton University Press, 1961. 14

Y. Bengio, A. Courville and P. Vincent. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013a. 19

Y. Bengio, Y. Mesnil, Grégoire Dauphin and S. Rifai. Better mixing via deep representations. In *Proc. 30th Int. Conf. Mach. Learn.*, pages 552–560, 2013b. 20

P. J. Bickel, Y. Ritov and A. B. Tsybakov. hierarchical selection of variables in sparse high-dimensional regression. In *Borrow. strength theory powering Appl. Festschrift Lawrence D. Brown*, pages 56–69. Institute of Mathematical Statistics, 2010. 78

J. Bien, J. Taylor and R. Tibshirani. A lasso for hierarchical interactions. *Ann. Stat.*, 41(3): 1111–1141, 2013. 77, 96

S. Bilodeau, M. H. Kagey, G. M. Frampton, P. B. Rahl and R. A. Young. SetDB1 contributes to repression of genes encoding developmental regulators and maintenance of ES cell state. *Genes \& Dev.*, 23(21):2484–2489, 2009. 61

N. J. Birkbak, B. Kochupurakkal, J. M. G. Izarzugaza, A. C. Eklund, Y. Li, J. Liu, Z. Szallasi, U. A. Matulonis, A. L. Richardson et al. Tumor mutation burden forecasts outcome in ovarian cancer with BRCA1 or BRCA2 mutations. *PLoS One*, 8(11):e80023, 2013. 33

J. R. M. Black and S. J. Clark. Age-related macular degeneration: genome-wide association studies to translation. *Genet. Med.*, 18(4):283–289, 2015. 10

B. M. Bolstad, R. A. Irizarry, M. Åstrand and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003. 28, 61, 62

A. Bonnefoy, V. Emiya, L. Ralaivola and R. Gribonval. Dynamic Screening: Accelerating First-Order Algorithms for the Lasso and Group-Lasso. *IEEE Trans. Signal Process.*, 63(19): 5121–5132, 2015. 77

J. H. Bullard, E. Purdom, K. D. Hansen and S. Dudoit. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics*, 11(1):94, 2010. 61

E. Bullmore and O. Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nat. Rev. Neurosci.*, 10(3):186–198, 2009. 21

B. Carvalho, H. Bengtsson, R. P. Speed and R. A. Irizarry. Exploration, normalization, and genotype calls of high-density oligonucleotide SNP array data. *Biostatistics*, 8(2):485–499, 2007. 61

E. G. Cerami, B. E. Gross, E. Demir, I. Rodchenkov, b. Babur, N. Anwar, N. Schultz, G. D. Bader and C. Sander. Pathway Commons, a web resource for biological pathway data. *Nucleic Acids Res.*, 39(Database issue):D685–D690, 2010. 36

P. B. Chapman, A. Hauschild, C. Robert, J. B. Haanen, P. Ascierto, J. Larkin, R. Dummer, C. Garbe, A. Testori et al. Improved Survival with Vemurafenib in Melanoma with BRAF V600E Mutation. *N. Engl. J. Med.*, 364(26):2507–2516, 2011. 10

A. Chatr-aryamontri, R. Oughtred, L. Boucher, J. Rust, C. Chang, N. K. Kolas, L. O'Donnell, S. Oster, C. Theesfeld et al. The BioGRID interaction database: 2017 update. *Nucleic Acids Res.*, 45(Database issue):D369–D379, 2016. 38

M. Chidgey and C. Dawson. Desmosomes: a role in cancer? *Br. J. Cancer*, 96(12):1783–1787, 2007. 41

J.-P. Chilès and P. Delfiner. *Geostatistics: Modeling Spatial Uncertainty, 2nd Edition.* Wiley, 2012. 61

L. Chin and J. W. Gray. Translating insights from the cancer genome into clinical practice. *Nature*, 452(7187):553–563, 2008. 33

A. Cho, J. E. Shim, F. Supek, B. Lehner and I. Lee. MUFFINN: cancer gene discovery via network analysis of somatic mutation data. *Genome Biol.*, 17(1):129, 2016. 53

F. R. K. Chung. *Spectral Graph Theory.* American Mathematical Soc., 1997. 21

G. Ciriello, M. L. Miller, B. A. Aksoy, Y. Senbabaoglu, N. Schultz and C. Sander. Emerging landscape of oncogenic signatures across human cancers. *Nat. Genet.*, 45(10):1127–1133, 2013. 6

N. Cloonan, A. R. R. Forrest, G. Kolle, B. B. A. Gardiner, G. J. Faulkner, M. K. Brown, D. F. Taylor, A. L. Steptoe, S. Wani et al. Stem cell transcriptome profiling via massive-scale mRNA sequencing. *Nat. Methods*, 5(7):613–619, 2008. 61

E. A. Collisson, J. D. Campbell, A. N. Brooks, A. H. Berger, W. Lee, J. Chmielecki, D. G. Beer, L. Cope, C. J. Creighton et al. Comprehensive molecular profiling of lung adenocarcinoma. *Nature*, 511(7511):543–550, 2014. 41, 48

P. Creixell, J. Reimand, S. Haider, G. Wu, T. Shibata, M. Vazquez, V. Mustonen, A. Gonzalez-Perez, J. Pearson et al. Pathway and network analysis of cancer genomes. *Nat. Methods*, 2 (3):1–6, 2015. 8, 33

I. Daubechies, M. Defrise and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.*, 57(11):1413–1457, 2004. 24

H. Davies, G. R. Bignell, C. Cox, P. Stephens, S. Edkins, S. Clegg, J. Teague, H. Woffendin, M. J. Garnett et al. Mutations of the BRAF gene in human cancer. *Nature*, 417(6892): 949–954, 2002. 10

S. Dharanipragada and M. Padmanabhan. A nonlinear unsupervised adaptation technique for speech recognition. In *Proc. 6th Int. Conf. Spok. Lang. Process.*, Beijing, China, 2000. 61

P. Diaconis. *Group representations in probability and Statistics.* Lecture Notes–Monograph Series. Institut of Mathematical Statistics, Hayward, CA, 1988. 65

M.-A. Dillies, A. Rau, J. Aubert, C. Hennequet-Antier, M. Jeanmougin, N. Servant, C. Keime, G. Marot, D. Castel et al. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Brief. Bioinform.*, 14(6):671–683, 2013. 61

C. M. Dimitrakopoulos and N. Beerenwinkel. Computational approaches for the identification of cancer genes and pathways. *Wiley Interdiscip. Rev. Syst. Biol. Med.*, 9(1):e1364, 2017. 8

L. Ding, G. Getz, D. A. Wheeler, E. R. Mardis, M. D. McLellan, K. Cibulskis, C. Sougnez, H. Greulich, D. M. Muzny et al. Somatic mutations affect key pathways in lung adenocarcinoma. *Nature*, 455(7216):1069–1075, 2008. 41, 48

D. L. Donoho and J. Tanner. Observed universality of phase transition in high-dimenisonal geometry, with applications for modern data analysis and signal processing. *Philos. Trans. R. Soc. London A Math. Phys. Eng. Sci.*, 367(1906):4273–4293, 2009. 89

Y. Drier, M. Sheffer and E. Domany. Pathway-based personalized analysis of cancer. *Proc. Natl. Acad. Sci.*, 110(16):6388–6393, 2013. 19

F. Dudbridge. Power and Predictive Accuracy of Polygenic Risk Scores. *PLoS Genet.*, 9(3): e1003348, 2013. 12

R. L. Dusek and L. D. Attardi. Desmosomes: new perpetrators in tumour suppression. *Nat. Rev. Cancer*, 11(5):317–323, 2011. 41

F. Eduati, L. M. Mangravite, T. Wang, H. Tang, J. C. Bare, R. Huang, T. Norman, M. Kellen, M. P. Menden et al. Prediction of human population responses to toxic compounds by a collaborative competition. *Nat. Biotechnol.*, 33(9):933–940, 2015. 87

B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, H. Ishwaran, K. Knight, J. M. Loubes, P. Massart, D. Madigan et al. Least angle regression. *Ann. Stat.*, 32(2):407–499, 2004. 23

L. El Ghaoui, V. Viallon and T. Rabbani. Safe feature elimination in sparse supervised learning. *Pacific J. Optim.*, 8(4):667–698, 2012. 25, 77

J. Fan and J. Lv. Sure independence screening for ultrahigh dimensional feature space. *J. R. Stat. Soc. Ser. B (Statistical Methodol.*, 70(5):849–911, 2008. 25

O. Fercoq, A. Gramfort and J. Salmon. Mind the Duality Gap: Safer Rules for the Lasso. In *Proc. 32nd Int. Conf. Mach. Learn.*, pages 333–342, 2015. 77

M. Fontoura, V. Josifovski, J. Liu, S. Venkatesan, X. Zhu and J. Y. Zien. Evaluation Strategies for Top-k Queries over Memory-Resident Inverted Indexes. *Proc. VLDB Endow.*, 4(12):1213–1224, 2011. 85

J. Friedman, T. Hastie, H. Höfling and R. Tibshirani. Pathwise coordinate optimization. *Ann. Appl. Stat.*, 1(2):302–332, 2007. 24

J. Friedman, T. Hastie and R. Tibshirani. Regularization Paths for Generalized Linear Models via Coordinate Descent. *J. Stat. Softw.*, 33(1):1–22, 2010. 24

Y. Fujiwara, Y. Ida, H. Shiokawa and S. Iwamura. Fast Lasso Algorithm via Selective Coordinate Descent. In *Proc. 30th Conf. Artif. Intell.*, pages 1561–1567, 2016. 77, 90

P. A. Futreal, L. Coin, M. Marshall, T. Down, T. Hubbard, R. Wooster, N. Rahman and M. R. Stratton. A census of human cancer genes. *Nat. Rev. Cancer*, 4(3):177–183, 2004. 7

A. Gionis, P. Indyk and R. Motwani. Similarity Search in High Dimensions via Hashing. In *Proc. 25th Int. Conf. Very Large Data Bases*, 1999. 95

R. C. Gonzalez and R. E. Woods. *Digital Image Processing (3rd Edition)*. Prentice Hall, 2008. 61

A. Gonzalez-Perez, C. Perez-Llamas, J. Deu-Pons, D. Tamborero, M. P. Schroeder, A. Jene-Sanz, A. Santos and N. Lopez-Bigas. IntOGen-mutations identifies cancer drivers across tumor types. *Nat. Methods*, 10(11):1081, 2013. 41

C. Greenman, P. Stephens, R. Smith, G. L. Dalgliesh, C. Hunter, G. Bignell, H. Davies, J. Teague, A. Butler et al. Patterns of somatic mutation in human cancer genomes. *Nature*, 446(7132):153, 2007. 33

D. F. Gudbjartsson, G. B. Walters, G. Thorleifsson, H. Stefansson, B. V. Halldorsson, P. Zusmanovich, P. Sulem, S. Thorlacius, A. Gylfason et al. Many sequence variants affecting diversity of adult human height. *Nat. Genet.*, 40(5):609–615, 2008. 11

D. Hanahan and R. A. Weinberg. Hallmarks of cancer: The next generation. *Cell*, 144(5): 646–674, 2011. 33

N. Hao and H. H. Zhang. Interaction Screening for Ultra-High Dimensional Data. *J. Am. Stat. Assoc.*, 109(507):1285–1301, 2014. 78

A. Haris, D. Witten and N. Simon. Convex Modeling of Interactions With Strong Heredity. *J. Comput. Graph. Stat.*, 25(4):981–1004, 2016. 77

T. Hastie and W. Stuetzle. Principal curves. *J. Am. Stat. Assoc.*, 84(406):502–516, 1989. 19

T. Hastie, R. Tibshirani and J. Friedman. *The Elements of Statistical Learning*. Springer series in statistics New York, 2001. 16

G. Hemani, S. Knott and C. Haley. An Evolutionary Perspective on Epistasis and the Missing Heritability. *PLoS Genet.*, 9(2):e1003295, 2013. ISSN 15. 12

S. C. Hicks and R. A. Irizarry. quantro: a data-driven approach to guide the choice of an appropriate normalization method. *Genome Biol.*, 16(1):117, 2015. 61

M. R. Hidalgo, C. Cubuk, A. Amadoz, F. Salavert, J. Carbonell-Caballero and J. Dopazo. High throughput estimation of functional cell activities reveals disease mechanisms and predicts relevant clinical outcomes. *Oncotarget*, 8(3):5160–5178, 2017. 18

F. Hilger and H. Ney. Quantile based histogram equalization for noise robust large vocabulary speech recognition. *IEEE Trans. Audio. Speech. Lang. Processing*, 14(3):845–854, 2006. 61

K. A. Hoadley, C. Yau, D. M. Wolf, A. D. Cherniack, D. Tamborero, S. Ng, M. D. Leiserson, B. Niu, M. D. McLellan et al. Multiplatform Analysis of 12 Cancer Types Reveals Molecular Classification within and across Tissues of Origin. *Cell*, 158(4):929–944, 2014. 6

A. E. Hoerl and R. W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67, 1970. 16

M. Hofree, J. P. Shen, H. Carter, A. Gross and T. Ideker. Network-based stratification of tumor mutations. *Nat. Methods*, 10(11):1108, 2013. 8, 26, 28, 33, 34, 36, 46, 51, 54, 55

H. Horn, M. S. Lawrence, J. X. Hu, E. Worstell, N. Ilic, Y. Shrestha, E. Kim, A. Kamburov, A. Kashani et al. A comparative analysis of network mutation burdens across 21 tumor types augments discovery from cancer genomes. *bioRxiv*, 2015. doi: 10.1101/025445. 53

J. P. Hou and J. Ma. DawnRank: discovering personalized driver genes in cancer. *Genome Med.*, 6(7):56, 2014. 33

N. Howlader, A. Noone, M. Krapcho, D. Miller, K. Bishop, C. Kosary, M. Yu, J. Ruhl, Z. Tatalovich et al. SEER Cancer Statistics Review, 1975-2014, National Cancer Institute, 2017. 12

D. W. Huang, R. a. Lempicki and B. T. Sherman. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat. Protoc.*, 4(1):44–57, 2009. 58

T. J. Hudson, W. Anderson, A. Aretz and A. D. Barker. International network of cancer genome projects. *Nature*, 464(7291):993, 2010. 33

R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf and T. P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level datas. *Biostatistics*, 4(2):249–264, 2003. 61, 71

P. Jia and Z. Zhao. VarWalker: Personalized Mutation Network Analysis of Putative Cancer Genes from Next-Generation Sequencing Data. *PLoS Comput. Biol.*, 10(2):e1003460, 2014. 33

Y. Jiao and J.-P. Vert. The Kendall and Mallows Kernels for Permutations. In *Proc. 32nd Int. Conf. Mach. Learn.*, JMLR:W&CP, pages 1935–1944, 2015. 65

Y. Jiao, M. R. Hidalgo, C. Çubuk, A. Amadoz, J. Carbonell-Caballero, J. P. Vert and J. Dopazo. Signaling Pathway Activities Improve Prognosis for Breast Cancer. *bioRxiv*, 2017. doi: 10.1101/132357. 18

T. Johnson and C. Guestrin. Blitz: A Principled Meta-Algorithm for Scaling Sparse Optimization. In *Proc. 32nd Int. Conf. Mach. Learn.*, pages 1171–1179, 2015. 25, 77, 80, 86

T. B. Johnson and C. Guestrin. StingyCD: Safely Avoiding Wasteful Updates in Coordinate Descent. In *Proc. 34th Int. Conf. Mach. Learn.*, pages 1752–1760, 2017. 77, 90, 91

C. Kandoth, M. D. McLellan, F. Vandin, K. Ye, B. Niu, C. Lu, M. Xie, Q. Zhang, J. F. McMichael et al. Mutational landscape and significance across 12 major cancer types. *Nature*, 503(7471):333–339, 2013. 33

M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi and M. Tanabe. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Res.*, 44(D1):D457–D462, 2016. 49, 58

M. Kasowski, F. Grubert, C. Heffelfinger, M. Hariharan, A. Asabere, S. M. Waszak, L. Habegger, J. Rozowsky, M. Shi et al. Variation in transcription factor binding among humans. *Science (80-.).*, 328(5975):232–235, 2010. 61

R. J. Klein, C. Zeiss, E. Y. Chew, J.-Y. Tsai, R. S. Sackler, C. Haynes, A. K. Henning, J. P. SanGiovanni, S. M. Mane et al. Complement factor H polymorphism in age-related macular degeneration. *Science (80-.).*, 308(5720):385–389, 2005. 10

S. Köhler, S. Bauer, D. Horn and P. N. Robinson. Walking the Interactome for Prioritization of Candidate Disease Genes. *Am. J. Hum. Genet.*, 82(4):949–958, 2008. 33

M. Kowalski, P. Weiss, A. Gramfort and S. Anthoine. Accelerating ISTA with an active set strategy. In *OPT 2011 4th Int. Work. Optim. Mach. Learn.*, page 7, 2011. 80

A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 2009. 68

I. Kuperstein, L. Grieco, D. P. A. Cohen, D. Thieffry, A. Zinovyev and E. Barillot. The shortest path is not the one you know: Application of biological network resources in precision oncology research. *Mutagenesis*, 30(2):191–204, 2015. 33

T. Lahusen, R. T. Henke, B. L. Kagan, A. Wellstein and A. T. Riegel. The role and regulation of the nuclear receptor co-activator AIB1 in breast cancer. *Breast Cancer Res. Treat.*, 116(2): 225–237, 2009. 41

E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860, 2001. 3

M. S. Lawrence, P. Stojanov, P. Polak, G. V. Kryukov, K. Cibulskis, A. Sivachenko, S. L. Carter, C. Stewart, C. H. Mermel et al. Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature*, 499(7457):214–218, 2013. 33, 49

M. S. Lawrence, P. Stojanov, C. H. Mermel, J. T. Robinson, L. a. Garraway, T. R. Golub, M. Meyerson, S. B. Gabriel, E. S. Lander et al. Discovery and saturation analysis of cancer genes across 21 tumour types. *Nature*, 505(7484):495–501, 2014. 7, 33

M. Le Morvan and J. P. Vert. Supervised Quantile Normalisation. *ArXiv e-prints*, 2017.

M. Le Morvan and J. P. Vert. WHInter: A Working set algorithm for High-dimensional sparse second order Interaction models. *ArXiv e-prints*, 2018.

M. Le Morvan, A. Zinovyev and J. P. Vert. NetNorM: Capturing cancer-relevant information in somatic exome mutation data with gene networks for cancer stratification and prognosis. *PLoS Comput. Biol.*, 13(6):e1005573, 2017.

I. Lee, U. M. Blom, P. I. Wang, J. E. Shim and E. M. Marcotte. Prioritizing candidate disease genes by network-based boosting of genome-wide association data. *Genome Res.*, 21(7):1109–1121, 2011. 38

J. D. Lee, D. L. Sun, Y. Sun and J. E. Taylor. Exact post-selection inference, with application to the lasso. *Ann. Stat.*, 44(3):907–927, 2016. 95

M. D. M. Leiserson, F. Vandin, H.-T. Wu, J. R. Dobson, J. V. Eldridge, J. L. Thomas, A. Papoutsaki, Y. Kim, B. Niu et al. Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nat. Genet.*, 47(2):106–114, 2014. 33, 34

G. Lettre, A. U. Jackson, C. Gieger, F. R. Schumacher, S. I. Berndt, S. Sanna, S. Eyheramendy, B. F. Voight, J. L. Butler et al. Identification of ten loci associated with height highlights new biological pathways in human growth. *Nat. Genet.*, 40(5):584–591, 2008. 11

M. Lim and T. Hastie. Learning Interactions via Hierarchical Group-Lasso Regularization. *J. Comput. Graph. Stat.*, 24(3):627–654, 2015. 77

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60 (2):91–110, 2004. 18

B. Maher. Personal genomes: The case of the missing heritability. *Nature*, 456(7218):18–21, 2008. 11

J. Mairal and Y. Bin. Complexity Analysis of the Lasso Regularization Path. In *Proc. 29th Int. Conf. Mach. Learn.*, pages 353—-360, 2012. 24

S. G. Mallat and Z. Zhang. Matching Pursuits With Time-Frequency Dictionaries. *IEEE Trans. Signal Process.*, 41(12):3397–3415, 1993. 23

T. A. Manolio, F. S. Collins, N. J. Cox, D. B. Goldstein, L. A. Hindorff, D. J. Hunter, M. I. McCarthy, E. M. Ramos, L. R. Cardon et al. Finding the missing heritability of complex diseases. *Nature*, 461(7265):747–753, 2009. 11

E. R. Mardis. Genome sequencing and cancer. *Curr. Opin. Genet. \& Dev.*, 22(3):245–250, 2012. 33

E. R. Mardis. DNA sequencing technologies: 2006-2016, 2017. 3

I. Martincorena, K. M. Raine, M. Gerstung, K. J. Dawson, K. Haase, P. Van Loo, H. Davies, M. R. Stratton and P. J. Campbell. Universal Patterns of Selection in Cancer and Somatic Tissues. *Cell*, 171(5):1029–1041.e21, 2017. 7

I. Martinocorena and P. J. Campbell. Somatic mutation in cancer and normal cells. *Science (80-.).*, 349(6255):1483–1489, 2015. 7

M. Massias, A. Gramfort and J. Salmon. From safe screening rules to working sets for faster Lasso-type solvers. *ArXiv e-prints*, 2017. 77

M. Massias, A. Gramfort and J. Salmon. Dual Extrapolation for Faster Lasso Solvers. *ArXiv e-prints*, 2018. 25

N. Mavaddat, P. D. P. Pharoah, K. Michailidou, J. Tyrer, M. N. Brook, M. K. Bolla, Q. Wang, J. Dennis, A. M. Dunning et al. Prediction of breast cancer risk based on profiling with common genetic variants. *J. Natl. Cancer Inst.*, 107(5):1–15, 2015. 12

T. Mikolov, K. Chen, G. Corrado and J. Dean. Efficient estimation of word representations in vector space. *ArXiv e-prints*, pages 1–12, 2013. 20

S. Molau, M. Pitz and H. Ney. Histogram based normalization in the acoustic feature space. In *Proc. IEEE Work. Autom. Speech Recognit. Underst.*, Madonna di Campiglio, Trento, Italy, 2001. 61

G.-L. Moldovan and A. D. D'Andrea. How the fanconi anemia pathway guards the genome. *Annu. Rev. Genet.*, 43:223–249, 2009. 49

K. Nakagawa, S. Suzumura, M. Karasuyama, K. Tsuda and I. Takeuchi. Safe Pattern Pruning: An Efficient Approach for Predictive Pattern Mining. In *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pages 1785–1794, 2016. 78, 86, 89, 114

B. K. Natarajan. Sparse Approximate Solutions to Linear Systems. *SIAM J. Comput.*, 24(2): 227–234, 1995. 23

E. Ndiaye, O. Fercoq, A. Gramfort and J. Salmon. Gap Safe screening rules for sparsity enforcing penalties. *J. Mach. Learn. Res.*, 18(128):1–33, 2017. 77, 80

Y. Nesterov. Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems. *SIAM J. Optim.*, 22(2):341–362, 2012. 24

B. Neyshabur and N. Srebro. On Symmetric and Asymmetric LSHs for Inner Product Search. *Proc. 32nd Int. Conf. Mach. Learn.*, pages 1926–1934, 2015. 95

L. G. Nyúl and J. K. Udupa. On standardizing the MR image intensity scale. *Magn. Reson. Med.*, 42(6):1072–1081, 1999. 61

L. G. Nyúl, J. K. Udupa and X. Zhang. New variants of a method of MRI scale standardization. *IEEE Trans. Med. Imaging*, 19(2):143–150, 2000. 61

M. Olivier and P. Taniere. Somatic mutations in cancer prognosis and prediction: lessons from TP53 and EGFR genes. *Curr. Opin. Oncol.*, 23(1):88–92, 2011. 33

M. R. Osborne, B. Presnell and B. a. Turlach. A new approach to variable selection in least squares problems. *IMA J. Numer. Anal.*, 20(3):389–403, 2000. 23

S. Osher and Y. Li. Coordinate descent optimization for l1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Probl. Imaging*, 3(3):487–503, 2009. 24

N. Parikh and S. Boyd. Proximal Algorithms. *Found. Trends® Optim.*, 1(3):127–239, 2014. 24

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12 (oct):2825–2830, 2012. 24, 55

C. M. Perou, T. Sørile, M. B. Eisen, M. Van De Rijn, S. S. Jeffrey, C. A. Ress, J. R. Pollack, D. T. Ross, H. Johnsen et al. Molecular portraits of human breast tumours. *Nature*, 406 (6797):747–752, 2000. 6

P. C. Phillips. Epistasis–the essential role of gene interactions in the structure and evolution of genetic systems. *Nat. Rev. Genet.*, 9(11):855, 2008. 12

T. S. K. Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen et al. Human Protein Reference Database–2009 update. *Nucleic Acids Res.*, 37(Database issue):D767–D772, 2009. 38

A. L. Price, N. J. Patterson, R. M. Plenge, M. E. Weinblatt, N. A. Shadick and D. Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nat. Genet.*, 38(8):904–909, 2006. 87

J. K. Pritchard. Are Rare Variants Responsible for Susceptibility to Complex Diseases? *Am. J. Hum. Genet.*, 69(1):124–137, 2001. 11

Y. Qian, S. Besenbacher, T. Mailund and M. H. Schierup. Identifying disease associated genes by network propagation. *BMC Syst. Biol.*, 8(Suppl 1):S6, 2014. 33

J. Qiao, S.-J. Cui, L.-L. Xu, S.-J. Chen, J. Yao, Y.-H. Jiang, G. Peng, C.-Y. Fang, P.-Y. Yang et al. Filamin C, a dysregulated protein in cancer revealed by label-free quantitative proteomic analyses of human gastric cancer cells. *Oncotarget*, 6(2):1171–1189, 2014. 41

P. Radchenko and G. James. Variable selection using Adaptive Nonlinear Interaction Structures in High dimensions. *J. Am. Stat. Assoc.*, 105(492):1541–1553, 2010. 77

A. Raj, J. Olbrich, B. Gärtner, B. Schölkopf and M. Jaggi. Screening Rules for Convex Problems. *ArXiv e-prints*, 2016. 77

P. Ranganathan, K. L. Weaver and A. J. Capobianco. Notch signalling in solid tumours: a little bit of everything but not all the time. *Nat. Rev. Cancer*, 11(5):338–351, 2011. 41

J. Rangel, S. Torabian, L. Shaikh, M. Nosrati, F. L. Baehner, C. Haqq, S. P. L. Leong, J. R. Miller, R. W. Sagebiel et al. Prognostic significance of nuclear receptor coactivator-3 overexpression in primary cutaneous melanoma. *J. Clin. Oncol.*, 24(28):4565–4569, 2006. 41

F. Rapaport, A. Zinovyev, M. Dutreix, E. Barillot and J.-p. Vert. Classification of microarray data using gene networks. *BMC Bioinformatics*, 8(1):35, 2007. 21, 33

B. J. Raphael, J. R. Dobson, L. Oesper and F. Vandin. Identifying driver mutations in sequenced cancer genomes: Computational approaches to enable precision medicine. *Genome Med.*, 6 (1):5, 2014. 7, 8, 94

N. A. Rizvi, M. D. Hellmann, A. Snyder, P. Kvistborg, V. Makarov, J. J. Havel, W. Lee, J. Yuan, P. Wong et al. Mutational landscape determines sensitivity to PD-1 blockade in non-small cell lung cancer. *Science (80-.).*, 348(6230):124–128, 2015. 33

A. I. Robles and C. C. Harris. Clinical outcomes and correlates of TP53 mutations and cancer. *Cold Spring Harb. Perspect. Biol.*, 2(3):a001016, 2010. 38

M. H. Roh, O. Makarova, C. J. Liu, K. Shin, S. Lee, S. Laurinec, M. Goyal, R. Wiggins and B. Margolis. The Maguk protein, Pals1, functions as an adapter, linking mammalian homologues of crumbs and discs lost. *J. Cell Biol.*, 157(1):161–172, 2002. 41

Y. Samuels and T. Waldman. Oncogenic mutations of PIK3CA in human cancers. *Curr. Top. Microbiol. Immunol.*, 347(1):21–41, 2010. 8

R. Scharpf, R. Irizarry, M. Ritchie, B. Carvalho and I. Ruczinski. Using the R Package crlmm for Genotyping and Copy Number Estimation. *J. Stat. Softw.*, 40(1):1–32, 2011. 61

J.-P. Serres. *Linear Representations of Finite Groups*. Graduate Texts in Mathematics. Springer-Verlag New York, 1977. 65

M. Shah, Y. Xiao, N. Subbanna, S. Francis, D. L. Arnold, D. L. Collins and T. Arbel. Evaluating intensity normalization on MRIs of human brain with multiple sclerosis. *Med. Image Anal.*, 15(2):267–282, 2011. 61

R. D. Shah. Modelling interactions in high-dimensional data with backtracking. *J. Mach. Learn. Res.*, 17(207):1–31, 2016. 78

R. T. Shinohara, E. M. Sweeney, J. Goldsmith, N. Shiee, F. J. Mateen, P. A. Calabresi, S. Jarso, D. L. Pham, D. S. Reich et al. Statistical normalization techniques for magnetic resonance imaging. *Neuroimage (Amst).*, 6:9–19, 2014. 61

A. Shrivastava and P. Li. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). In *Adv. Neural Inf. Process. Syst.*, pages 2321–2329, 2014. 92, 95

A. Shrivastava and P. Li. Improved Asymmetric Locality Sensitive Hashing (ALSH) for Maximum Inner Product Search (MIPS). In *Proc. 31st Conf. Uncertain. Artif. Intell.*, pages 812–821, 2015. 95

T. Sorlie, C. M. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, M. B. Eisen, M. van de Rijn et al. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc. Natl. Acad. Sci.*, 98(19):10869–10874, 2001. 6

N. Srebro and A. Shraibman. Rank, Trace-Norm and Max-Norm. In *Int. Conf. Comput. Learn. Theory*, pages 545–560, 2005. 65

M. R. Stratton, P. J. Campbell and P. A. Futreal. The cancer genome. *Nature*, 458:719–724, 2009. 33

S. Suzumura, K. Nakagawa, Y. Umezu, K. Tsuda and I. Takeuchi. Selective Inference for Sparse High-Order Interaction Models. In *Proc. 34th Int. Conf. Mach. Learn.*, volume 70, pages 3338–3347, 2017. 92, 95

O. Sysoev and O. Burdakov. A smoothed monotonic regression via L2 regularization. Technical Report LiTH-MAT-R–2016/01–SE, Department of mathematics, Linköping University, 2016. 64, 67

D. Szklarczyk, A. Franceschini, S. Wyder, K. Forslund, D. Heller, J. Huerta-Cepas, M. Simonovic, A. Roth, A. Santos et al. STRING v10: Protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.*, 43(D1):D447–D452, 2015. 38

C. Teflioudi and R. Gemulla. Exact and Approximate Maximum Inner Product Search with LEMP. *ACM Trans. Database Syst.*, 42(1):5:1—-5:49, 2016. 85, 92, 95

A. Tenesa and C. S. Haley. The heritability of human disease: Estimation, uses and abuses. *Nat. Rev. Genet.*, 14(2):139–149, 2013. 11

The 1000 Genomes Project Consortium, A. Auton, L. D. Brooks, R. M. Durbin, E. P. Garrison, H. M. Kang, J. O. Korbel, J. L. Marchini, S. McCarthy et al. A global reference for human genetic variation. *Nature*, 526(7571):68–74, 2015. 3

The Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70, 2012. 33

The Cancer Genome Atlas Research Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061, 2008. 33

The Cancer Genome Atlas Research Network. Integrated genomic analyses of ovarian carcinoma. *Nature*, 474(7353):609–615, 2011. 33

The Cancer Genome Atlas Research Network, J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander et al. The Cancer Genome Atlas Pan-Cancer analysis project. *Nat. Genet.*, 45(10):1113, 2013. 33

The International HapMap Consortium. The International HapMap Project. *Nature*, 426(6968): 789–796, 2003. 3

The International HapMap Consortium. A haplotype map of the human genome. *Nature*, 437 (7063):1299, 2005. 3

The Wellcome Trust Case Control Consortium. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661, 2007. 10

R. Tibshirani. Regression Selection and Shrinkage via the Lasso. *J. R. Stat. Soc. Ser. B (Statistical Methodol.*, 58(1):267–288, 1996. 16, 23, 77

R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor and R. J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *J. R. Stat. Soc. Ser. B (Statistical Methodol.*, 74(2):245–266, 2012. 25

R. J. Tibshirani and J. Taylor. Degrees of freedom in lasso problems. *Ann. Stat.*, 40(2):1198–1232, 2012. 16

A. N. Tikhonov. On the stability of inverse problems. *Dokl. Akad. Nauk SSSR*, 39(5):195–198, 1943. 16

C. Tomasetti, L. Marchionni, M. A. Nowak, G. Parmigiani and B. Vogelstein. Only three driver gene mutations are required for the development of lung and colorectal cancers. *Proc. Natl. Acad. Sci.*, 112(1):118–123, 2015. 7

V. Van Belle, K. Pelckmans, J. Suykens and S. Van Huffel. Support vector machines for survival analysis. In *Proc. 3rd Int. Conf. Comput. Intell. Med. Healthc.*, pages 1–8, 2007. 55

F. Vandin, E. Upfal and B. J. Raphael. Algorithms for detecting significantly mutated pathways in cancer. *J. Comput. Biol.*, 18(3):507—-522, 2011. 33

O. Vanunu, O. Magger, E. Ruppin, T. Shlomi and R. Sharan. Associating genes and protein complexes with disease via network propagation. *PLoS Comput. Biol.*, 6(1):e1000641, 2010. 33

J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans et al. The sequence of the human genome. *Science (80-.).*, 291 (5507):1304–1351, 2001. 3

B. J. Vilhjalmsson, J. Yang, H. K. Finucane, A. Gusev, S. Lindstrom, S. Ripke, G. Genovese, P. R. Loh, G. Bhatia et al. Modeling Linkage Disequilibrium Increases Accuracy of Polygenic Risk Scores. *Am. J. Hum. Genet.*, 97(4):576–592, 2015. 96

P. M. Visscher, W. G. Hill and N. R. Wray. Heritability in the genomics era - Concepts and misconceptions. *Nat. Rev. Genet.*, 9(4):255–266, 2008. 11

P. M. Visscher, N. R. Wray, Q. Zhang, P. Sklar, M. I. McCarthy, M. A. Brown and J. Yang. 10 Years of GWAS Discovery: Biology, Function, and Translation. *Am. J. Hum. Genet.*, 101(1): 5–22, 2017. 10

B. Vogelstein, N. Papadopoulos, V. E. Velculescu, S. Zhou, L. A. Diaz Jr. and K. W. Kinzler. Cancer Genome Landscapes. *Science (80-.).*, 339(6127):1546–1558, 2013. 7, 8, 33

M. J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using l1-constrained quadratic programming (Lasso). *IEEE Trans. Inf. Theory*, 55(5):2183–2202, 2009. 77

J. Wang, J. Zhou, P. Wonka and J. Ye. Lasso screening rules via dual polytope projection. In *Adv. Neural Inf. Process. Syst.*, pages 1070–1078, 2013. 77

J.-W. Wang, J. J. Gamsby, S. L. Highfill, L. B. Mora, G. C. Bloom, T. J. Yeatman, T.-c. Pan, A. L. Ramne, L. A. Chodosh et al. Deregulated expression of LRBA facilitates cancer cell growth. *Oncogene*, 23(23):4089–4097, 2004. 41

K. Wang, H. Gaitsch, H. Poon, N. J. Cox and A. Rzhetsky. Classification of common human diseases derived from shared genetic and environmental determinants. *Nat. Genet.*, 49(9): 1319–1325, 2017. 11

D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393 (6684):440, 1998. 53

M. N. Weedon, H. Lango, C. M. Lindgren, C. Wallace, D. M. Evans, M. Mangino, R. M. Freathy, J. R. Perry, S. Stevens et al. Genome-wide association analysis identifies 20 loci that influence adult height. *Nat. Genet.*, 40(5):575–583, 2008. 11

B. Weigelt and J. Downward. Genomic Determinants of PI3K Pathway Inhibitor Response in Cancer. *Front. Oncol.*, 2:109, 2012. 9

D. Welter, J. MacArthur, J. Morales, T. Burdett, P. Hall, H. Junkins, A. Klemm, P. Flicek, T. Manolio et al. The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. *Nucleic Acids Res.*, 45(Database issue):D896–D901, 2014. 10

D. H. Wolpert. The Supervised Learning No-Free Lunch Theorems. In *Soft Comput. Ind.*, pages 25–42. Springer, 2002. 14

D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.*, 1(1):67–82, 1997. 14

L. D. Wood, D. W. Parsons, S. Jones, J. Lin, T. Sjöblom, R. J. Leary, D. Shen, S. M. Boca, T. Barber et al. The genomic landscapes of human breast and colorectal cancers. *Science (80-.).*, 318(5853):1108–1113, 2007. 33

T. T. Wu, Y. F. Chen, T. Hastie, E. Sobel and K. Lange. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, 2009. 78

Z. Xiang, H. Xu and P. Ramadge. Learning sparse representations of high dimensional data on large scale dictionaries. In *Adv. Neural Inf. Process. Syst.*, pages 900–908, 2011. 77

Z. J. Xiang and P. J. Ramadge. Fast lasso screening tests based on correlations. In *IEEE Int. Conf. Acoust. Speech Signal Process.*, pages 2137–2140, 2012. 77

Y. Xie, A. Kapoor, H. Peng, J.-C. Cutz, L. Tao and D. Tang. IQGAP2 Displays Tumor Suppression Functions. *J. Anal. Oncol.*, 4(2):86–93, 2015. 41

J. Yang, B. Benyamin, B. P. McEvoy, S. Gordon, A. K. Henders, D. R. Nyholt, P. A. Madden, A. C. Heath, N. G. Martin et al. Common SNPs explain a large proportion of heritability for human height. *Nat. Genet.*, 42(7):565–569, 2010. 11

Y. H. Yang and N. P. Thorne. Normalization for two-color cDNA microarray data. In D. R. Goldstein, editor, *Stat. Sci. a Festschrift Terry Speed*, volume 40 of *Lecture Notes–Monograph Series*, pages 403–418. Institute of Mathematical Statistics, 2003. 61

P. Yousefi, K. Huen, R. Aguilar Schall, A. Decker, E. Elboudwarej, H. Quach, L. Barcellos and N. Holland. Considerations for normalization of DNA methylation data by Illumina 450K BeadChip assay in population studies. *Epigenetics*, 8(11):1141–1152, 2013. 61

Y. Yuan, E. M. V. Allen, L. Omberg, N. Wagle, A. Amin-Mansour, A. Sokolov, L. a. Byers, Y. Xu, K. R. Hess et al. Assessing the clinical utility of cancer genomic and proteomic data across tumor types. *Nat. Biotechnol.*, 32(7):644–652, 2014. 34

T. Yue and H. Wang. Deep Learning for Genomics: A Concise Overview. *ArXiv e-prints*, 2018. 20

E. Zeggini, L. J. Scott, R. Saxena, B. F. Voight and F. S. Collins. Meta-analysis of genome-wide association data and large-scale replication identifies additional susceptibility loci for type 2 diabetes. *Nat. Genet.*, 40(5):638–645, 2008. 11

D. Zhou, O. Bousquet, T. N. Lal, J. Weston and B. Schölkopf. Learning with local and global consistency. In S. Thrun, L. K. Saul and B. Schölkopf, editors, *Adv. Neural Inf. Process. Syst.*, pages 321–328. MIT Press, 2004. 36

A. Zinovyev, U. Kairov, T. Karpenyuk and E. Ramanculov. Blind source separation methods for deconvolution of complex signals in cancer biology. *Biochem. Biophys. Res. Commun.*, 430(3):1182–1187, 2013. 19

O. Zuk, E. Hechter, S. R. Sunyaev and E. S. Lander. The mystery of missing heritability: Genetic interactions create phantom heritability. *Proc. Natl. Acad. Sci.*, 109(4):1193–1198, 2012. 12

Résumé

Depuis le premier séquençage du génome humain au début des années 2000, de grandes initiatives se sont lancé le défi de construire la carte des variabilités génétiques inter-individuelles, ou bien encore celle des altérations de l'ADN tumoral. Ces projets ont posé les fondations nécessaires à l'émergence de la médecine de précision, dont le but est d'intégrer aux dossiers médicaux conventionnels les spécificités génétiques d'un individu, afin de mieux adapter les traitements et les stratégies de prévention. La traduction des variations et des altérations de l'ADN en prédictions phénotypiques constitue toutefois un problème difficile. Les séquenceurs ou puces à ADN mesurent plus de variables qu'il n'y a d'échantillons, posant ainsi des problèmes statistiques. Les données brutes sont aussi sujettes aux biais techniques et au bruit inhérent à ces technologies. Enfin, les vastes réseaux d'interactions à l'échelle des protéines obscurcissent l'impact des variations génétiques sur le comportement de la cellule, et incitent au développement de modèles prédictifs capables de capturer un certain degré de complexité. Cette thèse présente de nouvelles contributions méthodologiques pour répondre à ces défis. Tout d'abord, nous définissons une nouvelle représentation des profils de mutations tumorales, qui exploite leur position dans les réseaux d'interaction protéine-protéine. Pour certains cancers, cette représentation permet d'améliorer les prédictions de survie à partir des données de mutations, et de stratifier les cohortes de patients en sous-groupes informatifs. Nous présentons ensuite une nouvelle méthode d'apprentissage permettant de gérer conjointement la normalisation des données et l'estimation d'un modèle linéaire. Nos expériences montrent que cette méthode améliore les performances prédictives par rapport à une gestion séquentielle de la normalisation puis de l'estimation. Pour finir, nous accélérons l'estimation de modèles linéaires parcimonieux, prenant en compte des interactions deux à deux, grâce à un nouvel algorithme. L'accélération obtenue rend cette estimation possible et efficace sur des jeux de données comportant plusieurs centaines de milliers de variables originales, permettant ainsi d'étendre la portée de ces modèles aux données des études d'associations pangénomiques.

Mots Clés

mutations, réseaux de gènes, normalisation par les quantiles, polymorphismes mononucléotidiques (SNPs), LASSO avec interactions

Abstract

Since the first sequencing of the human genome in the early 2000s, large endeavours have set out to map the genetic variability among individuals, or DNA alterations in cancer cells. They have laid foundations for the emergence of precision medicine, which aims at integrating the genetic specificities of an individual with its conventional medical record to adapt treatment, or prevention strategies. Translating DNA variations and alterations into phenotypic predictions is however a difficult problem. DNA sequencers and microarrays measure more variables than there are samples, which poses statistical issues. The data is also subject to technical biases and noise inherent in these technologies. Finally, the vast and intricate networks of interactions among proteins obscure the impact of DNA variations on the cell behaviour, prompting the need for predictive models that are able to capture a certain degree of complexity. This thesis presents novel methodological contributions to address these challenges. First, we define a novel representation for tumour mutation profiles that exploits prior knowledge on protein-protein interaction networks. For certain cancers, this representation allows improving survival predictions from mutation data as well as stratifying patients into meaningful subgroups. Second, we present a new learning framework to jointly handle data normalisation with the estimation of a linear model. Our experiments show that it improves prediction performances compared to handling these tasks sequentially. Finally, we propose a new algorithm to scale up sparse linear models estimation with two-way interactions. The obtained speed-up makes this estimation possible and efficient for datasets with hundreds of thousands of main effects, thereby extending the scope of such models to the data from genome-wide association studies.

Keywords

mutations, gene networks, quantile normalisation, Single Nucleotide Polymorphisms (SNPs), LASSO with pairwise interactions