DISS. ETH NO. 26676

MACHINE LEARNING FOR INTERACTION DISCOVERY IN GENETICS AND BIOENGINEERING

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES OF ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

LAETITIA PAPAXANTHOS

Normalienne de l'École normale supérieure de Paris
Ingénieure civile de l'École nationale supérieure des mines de Paris

born on 11 November 1987
citizen of France

accepted on the recommendation of

Prof. Dr. Karsten Borgwardt, examiner
Prof. Dr. Niko Beerenwinkel, co-examiner
Prof. Dr. Alfonso Valencia, co-examiner

2020

Laetitia Papaxanthos: *Machine learning for interaction discovery in genetics and bioengineering,* © 2020

To my parents, Aline and Michael.

ABSTRACT

Background. As time passes, the field of biology is constantly revolutionised by the rapid emergence of technologies that have been providing larger and more diverse datasets. The availability of these large datasets enables in return discoveries of biological mechanisms and the development of new fields such as personalised medicine. Analysing these large datasets remain however challenging, because of their size and diversity, and of the underlying complex biological mechanisms. Unravelling these mechanisms requires the development of new data analysis methods, coming from domains such as pattern mining or machine learning. Among the various challenges and questions that arise from biological data, a core problem concerns how to handle biological interactions. Biological interactions are extremely diverse and appear indispensable in studies of molecular or macroscopic phenotypes. Transcription factor binding to DNA sequences are examples of core physical interactions, while indirect interactions can also exist, such as proteins operating in the same disease pathway. Due to the diversity in interaction types, a large number of models for interactions have been proposed throughout the years. In this thesis, we will examine several ways to model such interactions in two types of datasets and closely related problems to these dataset types.

Contributions. We focused on two dataset types, genome-wide association studies and large sequence-function datasets, to explore the potential of modelling interactions for better understanding and prediction of biological mechanisms.

In the first chapter of this thesis, we will focus on applications to genome-wide association study (GWAS) data, namely finding groups of genetic variants whose interaction would be responsible for a phenotype of interest. The relevance of this application lies in the fact that it is possible that a group of genetic variants is responsible for a phenotype while none of its subgroups would alter the phenotype. Additionally, GWAS datasets are typically confounded, as its samples can have different origins or covariates such as age or height. Performing association testing in confounded datasets without any adequate correction is highly at risk as it can result in many spurious associations. Therefore, only with the ability to correct for covariate factors, can algorithms that account for interactions be widely applicable to GWAS datasets. In the first chapter of the thesis, *we present two algorithms that are able to find statistically significant interactions of genetic variants in the presence of a categorical covariate. Two types of interactions are studied, first all higher-order interactions, which, as their number scales exponentially with the number of genetic variants, generate computational and statistical challenges, and second, all contiguous genomic regions potentially at the origin of genetic heterogeneity.*

In the second chapter of this thesis, we will focus on applications to functional genomics, in particular on function prediction of DNA-regulatory sequences in bacteria. Being able to accurately predict the function of regulatory sequences is highly relevant in field such as synthetic biology or bioengineering. To this end, *we build a deep learning model in order to accurately predict the functions of the regulatory sequences of interest training on a large-scale sequence-function dataset. We additionally provide reliable uncertainty estimates for the predicted values in order understand which predictions the model is confident about, so that the corresponding sequences could be used in downstream biological tasks. Finally, we compare several interpretability methods and show that the model is able to detect sequence determinants and to measure their position-dependent influence.*

Conclusion. We show that the methods introduced in these two chapters are able to leverage non-linear interactions to improve feature selection or prediction performance, respectively. We also provide software package and webserver in order to participate openly to the community's effort and advances. It would be possible to further extend the concepts and models presented in this thesis, either to weaken assumptions, incorporate domain knowledge or tackle related but different problems of similar and crucial importance, such as data integration or molecular design. We believe that the recent advances in machine learning, bioinformatics and biology greatly hold promise in the years to come.

RÉSUMÉ

Contexte. Ces dernières années, le domaine de la biologie a été constamment révolutionné par l'émergence rapide de technologies qui fournissent des données toujours plus importantes et diversifiées. La disponibilité de ces grands ensembles de données permet en retour de découvrir des mécanismes biologiques et de développer de nouveaux domaines tels que la médecine personnalisée. L'analyse de ces grands ensembles de données reste cependant difficile, en raison de leur taille et de leur diversité, ainsi que des mécanismes biologiques complexes sous-jacents. L'élucidation de ces mécanismes nécessite le développement de nouvelles méthodes d'analyse des données, provenant de domaines tels que le data mining ou l'apprentissage automatique. Parmi les divers défis et questions qui viennent de données biologiques, un problème central concerne la manière de traiter les interactions biologiques. Les interactions biologiques sont extrêmement diverses et semblent indispensables dans l'étude des phénotypes moléculaires ou macroscopiques. Les facteurs de transcription se liant aux séquences d'ADN sont des exemples d'interactions physiques essentielles, tandis que des interactions indirectes existent aussi, comme les interactions entre protéines qui agissent dans la même voie métabolique. En raison de la diversité des types d'interactions, un grand nombre de modèles d'interactions ont été proposés au fil des ans. Dans cette thèse, nous examinerons plusieurs façons de modéliser ces interactions dans deux types de données et des problèmes connexes à ces types de données.

Contributions. Nous nous sommes concentrés sur deux types de données, les études d'association pangénomique (EAG) et les grands jeux de données composées de pairs de séquence et fonction correspondante, afin d'explorer le potentiel de la modélisation des interactions pour une meilleure compréhension et prévision de mécanismes biologiques.

Dans le premier chapitre de cette thèse, nous nous concentrerons sur les applications à des données d'étude d'association pangénomique (EAG), à savoir la découverte de groupes de variants génétiques dont l'interaction serait responsable d'un phénotype d'intérêt. La pertinence de cette application réside dans le fait qu'il est possible qu'un groupe de variants génétiques soit responsable d'un phénotype alors qu'aucun de ses sous-groupes n'altèrerait individuellement le phénotype. En outre, les données d'EAG sont généralement sous l'influence de variables confondantes, car ses échantillons peuvent avoir des origines différentes ou des variables confondantes différentes, comme l'âge ou la taille. Effectuer des tests d'association sur de telles données sans correction adéquate est très risqué parce-que cela peut entraîner la découverte de nombreuses fausses associations. Par conséquent, ce n'est qu'avec la capacité de corriger pour des facteurs confondants que les algorithmes qui

tiennent compte des interactions peuvent être largement applicables aux ensembles de données d'EAG. Dans le premier chapitre de la thèse, *nous présentons deux algorithmes qui sont capables de trouver des interactions entre variants génétiques, associées, de manière statistiquement significative, à un phénotype d'intérêt en présence d'une variable confondante catégorielle. Deux types d'interactions sont considérés, tout d'abord toutes les interactions d'ordre supérieur, qui, comme leur nombre augmente de manière exponentielle avec le nombre de variants génétiques, génèrent des défis statistiques et de temps d'exécution, et deuxièmement, toutes les régions génomiques contiguës potentiellement à l'origine du phénomène d'hétérogénéité génétique.*

Dans le deuxième chapitre de cette thèse, nous nous concentrerons sur les applications à la génomique fonctionnelle, en particulier sur la prédiction de fonctions à partir de séquences d'ADN régulatrices chez les bactéries. Pouvoir prédire avec précision la fonction de séquences régulatrices est pertinent dans des domaines tels que la biologie synthétique ou la bio-ingénierie. À cette fin, *nous construisons un modèle d'apprentissage approfondi afin de prédire avec précision les fonctions des séquences régulatrices d'intérêt, modèle entraîné sur un grand jeu de donnée de pairs séquence-fonction. Nous fournissons en outre des estimations de l'incertitude du modèle pour les valeurs prédites, afin de comprendre quelles prédictions pourraient être utilisées en aval lors d'étude de phénomènes biologiques. Enfin, nous comparons plusieurs méthodes d'interprétation et montrons que le modèle est capable de detecter des déterminants séquentiels et de mesurer leurs influences en fonction de leurs positions.*

Conclusion. Nous montrons que les méthodes présentées dans ces deux chapitres sont capables d'exploiter les interactions non linéaires pour améliorer les performances de modèles d'apprentissage automatique en terme de sélection de caractéristique ou de prédiction, respectivement. Nous fournissons également un logiciel et un serveur web afin de participer ouvertement à l'effort et aux progrès de la communauté. Il serait possible d'étendre davantage les concepts et les modèles présentés dans cette thèse, soit pour prendre en compte des hypothèses plus faibles, soit pour incorporer des informations propres au domaine étudié, soit pour aborder des problèmes connexes d'importance similaire et cruciale, comme l'intégration de données ou la génération de nouvelles molecules. Nous pensons que les récents progrès de l'apprentissage automatique, de la bioinformatique et de la biologie sont très prometteurs pour les années à venir.

ACKNOWLEDGEMENTS

I would like to deeply thank:

- Prof. Dr. Karsten Borgwardt for his constant and excellent support and advice during the PhD, for the opportunity to work in his lab as well as for the generous funding to attend many conferences in our field.

- Prof. Dr. Niko Beerenwinkel and Prof. Dr. Alfonso Valencia for accepting to act as referees in my doctoral examination, as well as Prof. Dr. Dagmar Iber for chairing the examination.

- Dr. Felipe Llinares-López for co-authoring the three significant pattern mining publications presented in this thesis. His role in the successful completion of these publications has been irreplaceable. Thank you Felipe for being an excellent collaborator at all levels and for your great support during the PhD.

- Dr. Damián Roqueiro for his expertise in bioinformatics and machine learning, and for always being honest and encouraging.

- Dr. Maximilian Nickel, in Facebook AI Research NYC, for accepting my application as intern in Fall 2018 and for his great supervision.

- Collaborators Prof. Dr. Mahito Sugiyama for his contributions and expertise in pattern mining and Dr. Dean Bodenham for his contributions and for the launch of the creation a software package that contains methods developed during the PhD.

- Collaborators Prof. Dr. Yaakov (Kobi) Benenson and Dr. Markus Jeschek for conceiving the wet lab experiments of the second project of this thesis and for the valuable discussions.

- Margaux Dastor and Dr. Corentin Briat for so many excellent dinners and other enjoyable moments, students of the ITNs in particular Dr. Mélanie Pradier, FAIR interns in particular Terrance DeVries, and the squash team for having so much fun!

- My family and friends for always being present and supportive during the PhD.

- Simon Meng for making my life so much more worthwhile during these last years. I cannot thank you enough for these beautiful moments we have shared together, and hope there are many more to come!

CONTENTS

LIST OF FIGURES

LIST OF TABLES

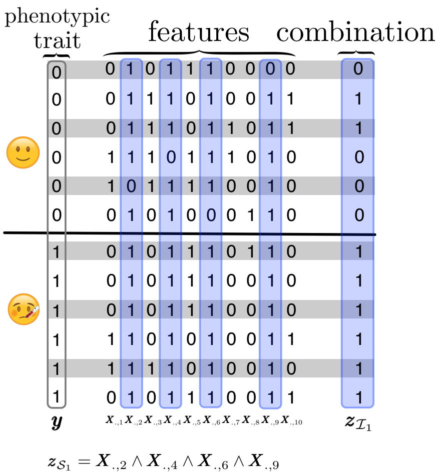
1

INTRODUCTION TO THE IMPORTANCE OF INTERACTIONS IN GENETICS AND BIOENGINEERING

Molecular interactions are omnipresent at all levels of biological systems, such as protein-protein interactions [1–3], DNA-binding proteins [4, 5] or DNA-RNA interactions [6]. Being able to decipher the role of interactions in biological phenomena can lead to a large amount of discoveries, potentially revolutionising the medical field in all its dimensions, for prevention, diagnosis and treatment. For example, in complex diseases such as asthma [7] or diabetes [8], it is possible that interacting DNA loci are responsible for the variance of the phenotypic trait or disease risk, besides existing or non-existing marginal effects (see Figure 1.1). Finding these interacting DNA loci could lead to a better molecular understanding of the diseases of interest and help to address which genes or molecules to target in drug discovery [9–11]. Additionally, interactions play a fundamental role in cancer where somatic mutations can be mutually exclusive or co-occurring, leading to different cancer pathways or different responses to treatment [12, 13]. An enhanced understanding of interactions between somatic mutations could help tackle the complexity behind cancer development to design better drugs and find effective drug combinations [14].

Our ability to collect biological data, and its ever-growing quantity and diversity, have enabled to push forward the research on biomarker discovery during the last years [15–17]. Together with information about the patients' environment and lifestyle, using the large pool of biological data available, towards a better understanding of interactions that drive diseases, could allow making personalised medicine a reality [18]. However, while the available large-scale datasets have led to the identification of several thousands of relevant biomarkers [19], less has been done in the domain of interaction discovery. A reason to explain this phenomenon is the size of the studied datasets, which fit the large-p small-n framework. For example, genome-wide association studies can contain a few hundreds to thousands of samples but several millions of DNA loci [10, 11]. In this setup, the detection of individual markers can be difficult due to the lack of statistical power and sizeable computational runtime. The detection of interacting markers is even more challenging, as the number of interactions grows exponentially with the number of individual markers. This could explain why very few methods enable an exhaustive search of biomarker interactions and why this topic remains an open problem.

However, the challenges behind modelling interactions do not only lie with the exponential number of interactions but also with the fact that often the identities of the interacting markers and the way they interact are unknown. In fact, the



$\boldsymbol{y}$	$\boldsymbol{X}_{.,1}$	$\boldsymbol{X}_{.,2}$	$\boldsymbol{X}_{.,3}$	$\boldsymbol{X}_{.,4}$	$\boldsymbol{X}_{.,5}$	$\boldsymbol{X}_{.,6}$	$\boldsymbol{X}_{.,7}$	$\boldsymbol{X}_{.,8}$	$\boldsymbol{X}_{.,9}$	$\boldsymbol{X}_{.,10}$	$\boldsymbol{z}_{\mathcal{I}_1}$
0	0	1	0	1	1	1	0	0	0	0	0
0	0	1	1	1	0	1	0	0	1	1	1
0	0	1	1	1	0	1	1	0	1	1	1
0	1	1	1	0	1	1	1	0	1	0	0
0	1	0	1	1	1	1	0	0	1	0	0
0	0	1	0	1	0	0	0	1	1	0	0
1	0	1	0	1	1	1	0	1	1	0	1
1	0	1	0	1	1	1	0	0	1	0	1
1	0	1	0	1	1	1	0	0	1	0	1
1	1	1	0	1	0	1	0	0	1	1	1
1	1	1	1	1	0	1	0	0	1	0	1
1	0	1	0	1	1	1	0	0	1	1	1

$$z_{\mathcal{S}_1} = \boldsymbol{X}_{.,2} \wedge \boldsymbol{X}_{.,4} \wedge \boldsymbol{X}_{.,6} \wedge \boldsymbol{X}_{.,9}$$

FIGURE 1.1: **Example of feature interaction**. None of the blue individual features are associated with the phenotype but the AND combination of these blue features (right) is associated.

term interaction can broadly refer to either physical interactions between molecules, such as among DNA and/or proteins molecules, or to indirect interactions between genes and/or non-coding DNA sequences, rendering its modelling complex. Due to the diversity of biological interactions, the literature refers to very different statistical and biological concepts when studying molecular interactions associated to a phenotype [20]. The commonly-used term to describe interactions between biological markers is epistasis, whose high-level definition refers to departure from additivity of the effects of two different features, at the penetrance scale, on a trait or disease. In practice, accepted models that test epistasis are additive, with additional multiplicative terms between different loci [21]. In these cases, epistasis is close to the concept of statistical interactions, which indicates departure from a specific linear model that describes the relationship of predictive factors with a phenotype of interest. As a consequence, there is a large variety of manners to model statistical interactions, and the biological meaning is dictated by the modelling choices.

Typical biomarker discovery methods often resort, in the large-p small-n framework, to univariate testing or regularised multivariate regression [22, 23]. Therefore, such methods are unable and insufficient to account for interactions between markers [11], which leads to the existence of missing heritability [24, 25]. Missing heritability conveys the idea that the variation of the trait under study is often poorly explained by the associated biomarkers. Therefore, novel methods are needed to be able to account for interacting biological markers in such a way that they describe the underlying biological mechanism, making possible to better explain the emergence of a phenotypic trait. To this end, machine learning can be used as a tool to discover or exploit these interactions, using feature selection algorithms [26] or prediction models [27], to solve biological questions. We will use throughout this thesis different types of models and features when considering statistical interactions and we will concentrate on their influence at two different biological levels.

In a first step, we will study interactions between loci in genes and/or non-coding DNA elements, and their impacts on a global phenotype, such as disease state or a physical trait under study. In this first chapter we will focus on different definitions of statistical interactions and will therefore interpret the results in terms of different types of biological interactions. However, as global phenotypes are often affected by several factors, some of them being difficult to control, we will explore a molecular phenotype, known to be mostly explainable from genetic variations alone, in the second part of the thesis. Therefore, in a second step, we will focus on nucleotide interactions in RNA sequences, using non-linear machine learning models, to determine the activity of a regulatory sequence of interest. The thesis is more precisely structured as follows.

The first chapter of this thesis, titled *Higher-order interaction discovery in genome-wide association studies*, focuses on different types of interactions between genetic variants. This chapter aims to enhance traditional genome-wide association studies (GWAS) methods to account for interactions when selecting statistically significantly associated genetic variants. Traditionally, univariate GWAS look for associations between single features and a phenotype of interest [11]. However, while this approach has had its successes, it has been hypothesised that accounting for interactions in GWAS [11] might partly alleviate this issue. These interactions are supported by biological phenomena, such as protein-protein interactions or DNA-protein interactions. As an example, we might observe that genetic features have an impact on the phenotype when both are minor alleles and no impact when only one of them is a minor allele. In this chapter, we will also focus on another specificity of GWAS studies, namely population structure or, more broadly, the presence of confounding covariates [28]. These covariates can give rise to spurious associations [28, 29], most commonly, when a feature and the phenotype are both associated to the covariate, therefore appearing marginally associated, although the feature and the phenotype are independent given the covariate. In summary, the first chapter of this thesis will present novel methods developed in order to discover statistically significant interactions of genetic variants in GWAS datasets, while correcting for a confounding covariate. This chapter will first give a broad introduction to GWAS, including a discussion on its common limitations, as well as to pattern mining techniques that have been used as building blocks of our proposed approach, to solve some of these limitations. Then, we will detail the methods we developed, and present comprehensive experimental results on both simulated and real-world GWAS data to assess the performance of our proposed approaches relative to the state of the art in the field.

The second chapter of this thesis, titled *Deep-learning enables accurate predictions of ribosome binding site activity*, aims to learn non-linear functions of multiple nucleotides in regulatory regions in order to predict their corresponding activities. With the emergence of large-scale labelled datasets in functional genomics [30, 31], there is a new opportunity to use complex, highly flexible models to fully benefit from the large available sample size. To this end, deep learning represents a promising approach due to its ability to model complex interactions between features in a data-driven fashion, as well as due to its scalability [32]. In this thesis, we will present a deep learning model tailored to the problem of predicting the activity of a specific regulatory region in bacteria, which obtains state-of-the-art performance. We will also show that we are able to assign well-calibrated uncertainty estimates to each predicted value, thereby providing guidance to design regulatory region sequences for downstream biological experiments. We will also show that, by applying SOTA interpretability techniques to the deep learning model, one can recover known motifs and positions of importance in the regulatory region of interest. This chapter is structured as follows. First, it will give a broad introduction to state-of-the-art methods in activity prediction for the regulatory region of interest and describe the wet lab experimental setting at the origin of the datasets used throughout the chapter. It will then detail the proposed approach we developed and present the experimental results we obtained in terms of prediction performance and model interpretability.

Finally, the thesis concludes with the chapter *Discussion and conclusion*, which will give a summary of the methods developed in this thesis, discuss some of their limitations and describe future research leads that could be explored towards solving some of these limitations or tackling related problems.

2

HIGHER-ORDER INTERACTION DISCOVERY IN GENOME-WIDE ASSOCIATION STUDIES

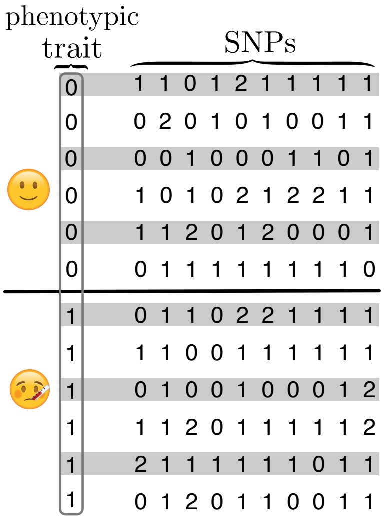
2.1 INTRODUCTION TO GENOME-WIDE ASSOCIATION STUDIES

In this section, we will first introduce genome-wide association studies, why it became popular, what do the datasets consist of and what type of analyses are traditionally performed on these datasets (Section 2.1.1). Second, we will present the main challenges and limitations when looking for associated genetic variants on GWAS datasets (Section 2.1.2).

2.1.1 *Detecting associations between genetic variants and phenotypic traits with GWAS*

Genome-wide association study (GWAS) is an experimental design that has been created towards the goal of facilitating the detection of associations between genetic variants and **phenotypic traits** in groups of samples [10, 11]. Compared to **linkage studies**, which pre-existed GWAS, a main advantage of GWAS was that it was agnostic. There was no need in GWAS to select candidate-genes a priori as studied SNPs were distributed genome-wide. GWAS was also more flexible as it was able to focus on case/control population data rather than families. As a consequence of these studies, the underlying biology of a trait or a disease would be better understood. It would allow hopefully to design a better treatment of a disease or more efficient prevention and early detection policies. The path from GWAS to a macroscopic phenomenon such as a trait or a disease is highly complex and not systematically informative about the associated genetic variant itself. For example, associated variants might not be causal of the disease or trait study. Nevertheless, GWAS can be understood as a first step towards screening genetic variants of interest among millions of other genetic variants. By finding significant associations between genetic variants and a phenotype in samples of populations, GWAS discoveries have been successful and had a major impact in autoimmune disease research and therapeutics [33, 34], in the understanding of metabolic diseases biological mechanisms [35] or identification of genes co-responsible for disease quantitative risk factors [36].

Typically, the datasets extracted from genome-wide association studies are used to test the association of individual genetic markers, such as **single-nucleotide polymorphisms** (SNPs), with a phenotypic trait of interest [37, 38]. These datasets are obtained experimentally using **SNP arrays** [39]. These SNP arrays scan from hundreds of thousands to millions of common genetic variants [10]. In the last years, the amount of data available is growing at an unprecedented rate, while keeping the amount of features orders of magnitude larger than the sample size. Therefore, statistical inference in high-dimensional spaces has become a tool of the utmost importance for practitioners in those fields. These datasets contain information on hundred of thousands to millions of SNPs. These SNPs are mostly biallelic, with a major or a minor allele. For analysis purposes, SNPs from the same loci in homologous chromosomes are combined into a categorical variable with three categories: homozygote major alleles, homozygote minor alleles and heterozygote alleles. The frequency of the second most common allele of each SNP is given by the minor allele frequency (MAF). It is usual to further binarise such encoding into a dominant encoding (homozygote major alleles are encoded as a 0 and the rest as a 1) or a recessive encoding (homozygote minor alleles are encoded as a 1 and the rest as a 0). The phenotypic trait is traditionally a binary label, i.e. presence (case) or absence (control) of a disease. However categorical, i.e. different stages of a disease, or continuous, i.e. quantitative traits such as body-mass index, variables can also be found. Figure 2.1 shows an example of GWAS dataset.



	phenotypic trait	SNPs									
	0	1	1	0	1	2	1	1	1	1	1
	0	0	2	0	1	0	1	0	0	1	1
	0	0	0	1	0	0	0	1	1	0	1
🙂	0	1	0	1	0	2	1	2	2	1	1
	0	1	1	2	0	1	2	0	0	0	1
	0	0	1	1	1	1	1	1	1	1	0
	1	0	1	1	0	2	2	1	1	1	1
	1	1	1	0	0	1	1	1	1	1	1
🤒	1	0	1	0	0	1	0	0	0	1	2
	1	1	1	2	0	1	1	1	1	1	2
	1	2	1	1	1	1	1	1	0	1	1
	1	0	1	2	0	1	1	0	0	1	1

FIGURE 2.1: **Example of a GWAS dataset**. The phenotypic trait is the presence (represented by a 1) or the absence (represented by a 0) of a disease. The ten SNPs are either homozygote with two major alleles (encoded as a 0), heterozygote (encoded as a 1) or homozygote with two minor alleles (encoded as a 2).

Common analyses of GWAS datasets consist of performing univariate tests. These univariate tests consider the effect of each SNP in isolation from the rest. Statistical tests used can be for example the χ^2 test [40], Fisher's exact test [41], the likelihood ratio test [42] or the Cochran-Mantel-Haenszel (CMH) test [43, 44] in the presence of a categorical covariate. The resulting p-values are used to assess the significance of the association between each genetic variant and the phenotypic trait.

2.1.2 *Limitations and challenges in GWAS*

However, traditional GWAS studies have shown limitations and challenges [10, 11], three of them are explained in this section, namely the existence of SNP interactions (Section 2.1.2.1), the multiple hypothesis testing problem (Section 2.1.2.2) and the existence of confounding factors (Section 2.1.2.3).

2.1.2.1 *Finding interactions*

First, traditional GWAS studies systematically miss associated non-linear interactions between genetic variants by being restricted to univariate tests. However, non-linear statistical interactions originate from biological interactions between genetic variants. These interactions are either indirect, such as interactions between proteins operating in the same pathway, or direct, such as physical interactions between proteins or between a protein and a regulatory region [20, 45]. The consequence of missing interactions is twice. First, a loss of power can occur due to the fact that interacting SNPs are not detected with univariate analyses. For example, an interaction could be significantly associated with a phenotypic trait while its individual variants would not be. Second, failing to test for genetic variant interactions can result in a loss of biological understanding of the emergence of molecular and macroscopic phenotypes. Third, the existence of biological interactions have led to the phenomenon known as **missing or phantom heritability**, which is the fact that the maximum variance of the phenotype that can be explained by a linear combinations of the individual features is in general low when interactions are not accounted for [25]. Several studies have tempted to account for interactions using exhaustive enumeration methods, filtering approaches, index structure approaches and machine-learning driven approaches [20, 46–48]. However, the exact formulation of the statistical interactions leads to different hypotheses being tested and in consequence to different types of biological mechanisms being explored. Common interaction modelling choices will be more extensively introduced Section 2.2 together with a short note on interaction discovery in cancer research.

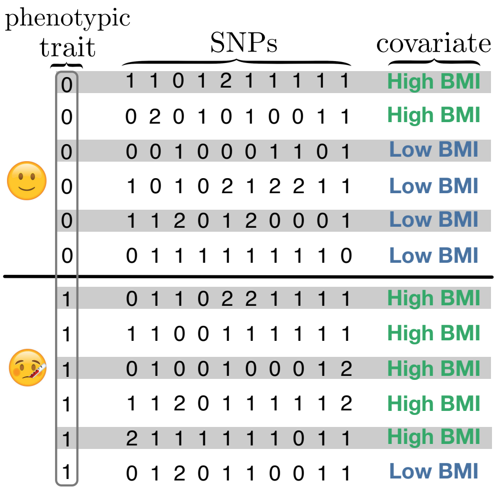
2.1.2.2 *Correcting for the multiple hypothesis problem*

Second, GWAS datasets typically fit the "large-p, small-n" framework, with hundreds or thousands of samples (n), and hundreds of thousands to millions of SNPs (p). In traditional univariate analyses, the number of statistical tests performed is as large as the number of features (here SNPs) p, which leads to the **multiple comparisons** or **multiple hypothesis testing** (MHT) problem [49, 50]. To briefly illustrate the MHT problem, let's assume a GWAS dataset contains $n = 1000$ samples and

$p = 100{,}000$ features. With univariate methods, $p = 100{,}000$ statistical tests would be performed and a p-value would be computed for each feature under the null hypothesis of independence between the feature and the phenotypic trait. Let us assume the significance threshold is set to $\delta = 0.05$. Let us also choose a test statistic that measures independence between a feature and the phenotypic trait. A null distribution can be defined that holds (asymptotically or not) in the case of independence. In this setting, a p-value smaller than the significance threshold would indicate that if we were to sample random values under the null distribution, less than 5% of the values would be more extreme than the test statistics of interest. Therefore, a p-value smaller than the significance threshold indicates that the null distribution most likely does not hold for the variant of interest. The SNP is therefore considered as associated. However, it would also be possible that such an extreme value is correctly modelled by the null distribution even if being rare. In this case, the null hypothesis would actually hold and be rejected by mistake. The SNP would be in practice not associated but would appear associated by random chance. Importantly, with a significance threshold of δ (here $\delta = 0.05$), the likeliness of the mistake would be small and correspond to a fraction δ (here 5%) of the number of tests performed under the null hypothesis. Therefore, in a dataset of 100,000 SNPs, approximately 5000 SNPs would be deemed false positives. In order to reduce the number of false positives, it is common to use multiple hypothesis correction procedures. However, standard MHT correction methods have a tendency to be too conservative and to, by contrast, yield too many false negatives [51], see Section 2.3.1. Therefore, we will introduce alternative methods that lead to an increase of statistical power compared to standard methods, see Section 2.3.1.3.

2.1.2.3 *Correcting for confounding factors*

Third, GWAS datasets often present **confounding factors** that influence both the phenotypic trait and some genetic variants. As a consequence, several spurious associations can be detected. Examples of confounding factors are population structure or **covariate variables** such as sex or BMI, as illustrated Figure 2.2. For illustration purposes, let us assume a GWAS dataset contains samples that belong to different BMI level groups (high and low BMI) and that the phenotypic trait is binary (presence or absence of a disease). It is possible that these BMI level groups show differences in prevalence of some genetic variants, which are associated with high BMI for example. When, additionally, the BMI level groups are unevenly distributed across phenotypic classes, it can result in false associations between the genetic variants and the phenotypic trait [28]. Therefore, it is necessary to model the presence of confounding factors in order to correct for their effects and eliminate potential false positive associations. An overview of methods that allow accounting for confounding factors will be presented Section 2.3.2.



phenotypic trait	SNPs										covariate
0	1	1	0	1	2	1	1	1	1	1	High BMI
0	0	2	0	1	0	1	0	0	1	1	High BMI
0	0	0	1	0	0	0	1	1	0	1	Low BMI
0	1	0	1	0	2	1	2	2	1	1	Low BMI
0	1	1	2	0	1	2	0	0	0	1	Low BMI
0	0	1	1	1	1	1	1	1	1	0	Low BMI
1	0	1	1	0	2	2	1	1	1	1	High BMI
1	1	1	0	0	1	1	1	1	1	1	High BMI
1	0	1	0	0	1	0	0	0	1	2	High BMI
1	1	1	2	0	1	1	1	1	1	2	High BMI
1	2	1	1	1	1	1	1	0	1	1	High BMI
1	0	1	2	0	1	1	0	0	1	1	Low BMI

FIGURE 2.2: **Example of a GWAS dataset with one categorical covariate**. In addition to the GWAS dataset described Figure 2.1, BMI level is accounted for as binary covariate.

Significant pattern mining (SPM) [52, 53] is a data mining subfield that consists of finding sets of features that are significantly more frequently present in a group of samples than in another one. In **pattern mining**, the features are represented with binary vectors, whose elements indicate the presence or absence of the feature in each sample of the dataset. A pattern is a subset of features and is represented by a binary vector. Each element of the representative vector indicates whether all the features of the pattern occur in the given sample. In **significant pattern mining**, the samples are further labelled with a binary encoding, splitting the samples of the population into two groups. For each pattern, the objective of SPM is to determine whether the pattern is statistically significantly enriched in samples that belong to one of the two classes. This is equivalent to testing the statistical association between the binary label and the pattern vector. Several developments have enabled SPM methods to be applicable to large-scale datasets [48, 54, 55]. The setting of significant pattern mining is particularly relevant to case/control studies in GWAS, where the features can be binarised, using a dominant or recessive encoding, and the labels are naturally binary (cases and controls).

The aim of this chapter is therefore to introduce a series of methods that are able to find interactions of genetic variants significantly associated with a phenotypic trait of interest, while correcting for both (i) the multiple hypothesis testing problem, to increase the true positive rate compared to standard MHT correction methods, and (ii) covariates, to reduce the false positive rate due to confounding. This chapter is organised as follows:

- Section 2.2, we will first introduce previous work in interaction discovery.
- Section 2.3, we will then present some significant pattern mining background work in multiple hypothesis testing correction and confounder correction.
- Section 2.4, a novel significant pattern mining method (Fast Automatic Conditional Search, `FACS`) will be described [56], which handles binary feature interactions while accounting for categorical covariates, together with a strict MHT correction. We will show that `FACS` is able to find associated interactions of genetic variants in GWAS in a proof-of-concept experiment. The work presented in this section is available in the following publication, for which the two first authors contributed equally.
  - Papaxanthos, L., Llinares-López, F., Bodenham, D. & Borgwardt, K. *Finding significant combinations of features in the presence of categorical covariates* in Advances in Neural Information Processing Systems (2016), 2271–2279
- Section 2.5, another significant pattern mining algorithm (`FastCMH`) that focuses on detecting genetic heterogeneity in GWAS data will be presented [57]. This method has been published under the following reference, work for which the two first authors contributed equally.
  - Llinares-López, F., Papaxanthos, L., Bodenham, D., Roqueiro, D., COPD Investigators & Borgwardt, K. *Genome-wide genetic heterogeneity discovery with categorical covariates* in Bioinformatics 33, i1820–i1828 (2017)
- Section 2.6, a ready-to-use software package, that includes the above significant pattern mining methods together with some predecessor methods, will be introduced [58]. The software package originates from the following publication, where the two first authors contributed equally.
  - Llinares-López, F., Papaxanthos, L., Roqueiro, D., Bodenham, D., & Borgwardt, K. *CASMAP: detection of statistically significant combinations of SNPs in association mapping* in Bioinformatics (2019)

Please note that the text of Sections 2.4 to 2.6 have been largely inspired from the existing publications they are describing.

2.2 STATE OF THE ART IN INTERACTION DISCOVERY IN GWAS

Notation: We assume n samples and p genetic variants arranged in a $n \times p$ design matrix $\boldsymbol{X}$, such that the columns of the matrix $\boldsymbol{X}_{.,i} = (X_{1i}, X_{2i}, ..., X_{ni})$ correspond to each variant $i \in [\![1,p]\!]$ and the rows $\boldsymbol{X}_{i,.} = (X_{i1}, X_{i2}, ..., X_{ip})$ correspond to each sample $i \in [\![1,n]\!]$. A subset of variants of $\boldsymbol{X}$ is written as $\mathcal{S} = \{\boldsymbol{X}_{.,i_1}, \boldsymbol{X}_{.,i_2}, ..., \boldsymbol{X}_{.,i_k}\}$ and a contiguous interval of variants of $\boldsymbol{X}$ is written as $\mathcal{I} = \{\boldsymbol{X}_{.,s}, \boldsymbol{X}_{.,s+1}, \boldsymbol{X}_{.,s+2}, ..., \boldsymbol{X}_{.,e}\}$, where $1 \leq s \leq e \leq p$, s being the starting position of the region and e the ending position. In this thesis, the terms 'genomic interval' and 'genomic region' will be used indistinguishably. Let $\boldsymbol{y}$ define the phenotypic trait vector, where y_i corresponds to the phenotype of sample i. Let μ be the empirical mean of $\boldsymbol{y}$. Additionally, we assume that $\mathbf{C}$ is a matrix of covariate variables, with n rows that correspond to samples and c columns that correspond to individual covariates.

In this section, we describe a first part of the necessary background on which the methods we propose are built. First, in Section 2.2.1, we introduce intrinsic limitations of univariate genome-wide association studies. Then, Sections 2.2.2, 2.2.3 and 2.2.4 present state-of-the-art approaches in GWAS in interaction discovery.

2.2.1 *Problem statement*

Several plausible mechanisms by which genetic variation could be linked to phenotypes cannot be captured by traditional univariate GWAS studies [24] and include (i) **low-frequency** (i.e. with a minimum allele frequency (MAF) such that $0.5\% \leq \text{MAF} < 5\%$) or rare genetic variants (i.e. $\text{MAF} < 0.5\%$), with possibly strong effects [59], (ii) **epistasis**, which can be defined as non-linear higher-order interactions between common variants, (iii) **genetic heterogeneity**, which consists of the production of single or similar phenotypes through different genetic mechanisms, therefore leading to multiple distinct genetic variants being weakly associated, and/or (iv) other environmental phenomena. Mechanisms explained by hypothesis (i) cannot be captured by common GWAS analyses as SNPs arrays mostly record common variants, present on 5% or more of a population omitting rare variants, and GWAS studies often fail to detect associations to rare variants or to variants with weak effects, due to the large number of tests performed and the relatively low sample size. In other words, the individual signal carried by some variants is too weak to be discovered in a single SNP study. Additionally, hypotheses (ii) and (iii), which existence has been corroborated by several studies [60], are missed by univariate testing of single SNPs. Finally, environmental phenomena (iv) are not captured by traditional SNP genotyping. Therefore, conditions (i) to (iv) are in general leading to a low statistical power in GWAS studies. As explained, due to the intrinsic properties of the GWAS datasets, the opportunity to study hypotheses (i) and (iv) can be discarded. In addition, from a biological point of view, there is *a priori* no reason to expect that traits should be additive. Biology consists of many non-linearities, from the saturation of enzymes with substrate concentration, the cooperative binding of proteins or redundant pathways. However, traditional GWAS studies heavily rely on univariate testing, finding individual loci and potentially missing interactive SNPs. When estimating the maximum variance of the phenotype that can be explained by a linear combination of allele counts, termed **narrow sense heritability** [24, 25], researchers have realised the existence of a missing or phantom heritability, i.e. the contribution of the individual loci to the total variation of the phenotype is rather low. This missing heritability has been explained partly by the fact that total variation of the phenotypic trait was overestimated, due to the fact that its estimation did not include non-additive terms. The existence of missing heritability corroborates the importance of hypotheses (ii) and (iii), towards which our research projects have been directed.

State-of-the-art methods that handle hypotheses (ii) and (iii) in GWAS include **aggregation tests** (region-based and gene-based) such as burden tests among others [61] (hypothesis (ii)), **epistasis studies** [46, 47, 62] (hypothesis (iii)) and more recently **significant pattern mining methods** developed with the GWAS application in mind [48] (hypothesis (ii)). A brief description of these three types of approaches is given below.

2.2.2 *Burden tests*

Burden tests evaluate the cumulative effects of multiple genetic variants in a gene or region. They are motivated by the fact that if several variants in a group are associated to the given disease or trait, statistical power will increase compared to testing individual SNPs. Several modifications have been brought to the original burden tests, such as including effect signs or null variants in the group, however most burden tests still consist of (i) summarising a region with a summary genetic score, (ii) building a corresponding score statistic and (iii) testing the null hypothesis, which assumes no association between the region of interest, as represented by the summary score, and the phenotypic trait. Following the above notations, assuming the region of interest is the SNP window $[\![i_s, i_e]\!]$, a common choice for summary genetic score can be written as $\boldsymbol{s} = \sum_{j=i_s}^{i_e} w_j \boldsymbol{X}_{.,j}$ where w_j is a fixed weight for variant j. The null hypothesis H_0 of independence between the region and the phenotypic trait is tested in the model $\boldsymbol{y} = \beta_0 + \beta \boldsymbol{s}$ (assuming no covariate variable, see Section 2.3.2). Extensions to binary $\boldsymbol{y}$ (case/control) are possible. The corresponding score statistic to test $H_0 : \beta = 0$ is then $q_{burden} = (\sum_{j=i_s}^{i_e} w_j \sum_{i=1}^{n} X_{ij}(y_i - \mu))^2$. The advantages of burden tests are two-fold: (i) the ability to aggregate additive weak effects in order to increase the corresponding signal and (ii) in the case of available prior knowledge of the regions of interest, the possibility to reduce the number of tests to a bare minimum, therefore not being penalised by a traditionally conservative multiple hypothesis testing correction (see Section 2.3.1). However, in the absence of prior biological knowledge, such as the location of genes and exons likely to be associated with the trait of interest, the positions and lengths of the studied regions would be arbitrarily chosen, therefore potentially missing regions that would be associated and leading to a loss of statistical power.

2.2.3 *Epistasis studies*

There is a large body of work covering epistasis studies. Ideally, epistasis methods should be exhaustive in terms of length of the sets of SNPs that are tested, handle homozygous and heterozygous SNPs to be applicable to human GWAS and scale to hundred of thousands of SNPs at least and thousands of samples. However, as this ideal objective is computationally and statistically extremely challenging, existing methods impose constraints on these aspects. State-of-the-art epistasis methods can be broadly classified into four categories: exhaustive enumeration methods, filtering approaches, index-structure approaches and machine-learning driven approaches [46, 47, 62, 63].

Exhaustive approaches aim at testing all sets of SNPs exhaustively. As the number of sets of SNPs scale exponentially with the number of SNPs, some methods limit themselves to testing sets of SNPs of bounded size, in general pairs of SNPs (a set of size two). Testing all pairs corresponds to the large-p small-n setting and leads to computational and statistical challenges. To this end, [64, 65] introduce methods that use computing clusters and works on graphical processing units, optimized for basic matrix operations. However, these methods are not able to handle higher-order interactions. Another approach that aims at searching for higher-order interactions is proposed in [66]. The latter uses multifactor-dimensionality reduction (MDR) as a model-free (it does not assume any inheritance model) and non-parametric (no hypothesis is made about the value of statistical parameters) approach to avoid choosing a priori a type of encoding, which might affect the results in the absence of prior knowledge. MDR first selects a set of SNPs, reduces the dimension of the table recording case-control ratio for each combination of locus/SNP encoding and uses prediction accuracy on the selected multifactor model to learn the most relevant set of SNPs. The procedure is repeated several times. However, this method encounters scalability issues when the number of features is large and is mainly applicable to balanced case-control datasets. Later, model-based MDRs have beens developed [67, 68], with a focus on pairs of SNPs most often. At last, [69, 70] propose an exhaustive search of pairwise interactions, using bitwise encodings, to save memory and computational runtime, and the likelihood ratio test, to measure departure from the additive model for each pair of SNPs.

Another group of approaches rely on a first filtering step. These approaches use a two-stage procedure, first reducing the set of SNPs based on statistical criteria [71–74] or biological criteria [75, 76] and then computing all remaining pairs exclusively or running a L2-regularised linear regression on the remaining SNPs [77].

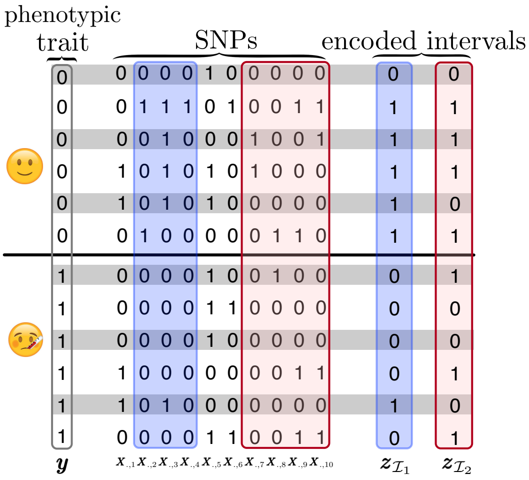
Third, index-structure approaches can be used in order to save memory and computational runtime. [78] (respectively [79]) proposes to build on the ANOVA test (respectively across different test statistics) to find all pairs of binary SNPs that are significantly associated with a phenotype, while controlling for the FWER. Its advantage in computational runtime comes from the fact that it uses an upper bound to filter out SNP-pairs having no chance to become significant, computes efficiently the upper bound and identifies redundant cases in the permutation tests. Another example is [80], which utilises the minimum spanning tree structure in a depth-first search algorithm to update contingency tables of the pairs containing a SNP of interest, without scanning all individuals. It allows performing efficient epistasis detection on homozygous and heterozygous data, controls for both FWER and FDR and is efficient in large sample studies.

Fourth, several bayesian methods have been proposed, where the objective is ideally to learn the dependency structure between SNPs, and with the phenotype. A first one [81] relies on Markov Chain Monte Carlo (MCMC) to test iteratively each marker conditioning on the previous ones, and distribute them among three groups, unlinked to the phenotype, contributing independently or contributing together with other SNPs to the phenotype. Then markers are further filtered to obtain a reduced set of important SNPs. As MCMC can present scalability issues, several extensions have been proposed afterwards. [82] considers that the SNPs are causal to the phenotype, proposing a prior to the directed acyclic graph joining SNPs and phenotype and replaces the MCMC step by a branch-and-bound strategy. [83–85] rely on a Markov-blanket strategy that consists of finding the smallest set of influencing SNPs. A notable sampling approach is the epistasis lightbulb algorithm [86, 87], which detects pairs of homozygous or heterozygous SNP interactions, by phrasing epistasis detection as a difference in correlation problem between cases and controls. The estimated maximum correlation in cases or controls can be calculated sub-quadratically in the number of SNPs, by estimating a correlation between two vectors sampling only k rows several times. Finally, [88–90] are tree-based algorithms that rely on variable importance techniques to find the sets of SNPs the most associated with a phenotype of interest.

While the studies presented above have proven their importance in disease understanding and disease risk prediction, epistasis has been extensively studied in other domains and in particular in the domain of cancer research. When studying cancer, epistasis search has integrated different sources of data, such as mutation, copy number and mRNA expression datasets. Identifying epistasis in cancer genome is at the heart of understanding cancer evolution, cancer pathways [91] and identifying effective combination therapies [14]. Epistasis is studied between somatic mutations on the same genomes and can refer to, for example, mutually exclusive mutations or to co-occurring mutations for example. Mutually exclusive mutations could happen when a driver mutation is less likely to occur when an earlier mutation has a redundant functional effect in the same molecular pathway. Inversely, a driver mutation could arise if it acts synergistically with a previous mutation. Several studies have tackled mutual exclusivity of cancer mutations [92, 93] or have modelled tumour mutational profiles and interactions between mutated genes [94] with the objective to allow genomic stratification for clinical trials and identifying drug targets. Numerous studies have also searched cancer genome for synthetic lethal genetic interactions for understanding genotype-phenotype relationship or identifying drug-targets against cancer [95–97]. These cancer studies together with the studies presented earlier show the importance of research in epistasis in order to uncover biological mechanisms and push medical discoveries forward [98, 99].

2.2.4 *Interval search algorithm*

More recently, the Fast Automatic Interval Search (FAIS) algorithm [48] enables to detect any GWAS contiguous region that would be significantly associated with a binary phenotypic trait. This method tackles the problem of genetic heterogeneity, by searching for **genomic intervals** $\mathcal{I}$ in which the occurrence of a type of sequence variant (e.g. a point mutation or minority allele) present in at least one genetic variant of the genomic region is significantly more frequent in one of the two phenotypic classes. Formally, each SNP has a binary encoding (dominant or recessive) and the encoding of the interval implements the OR operation between the SNPs that it contains: $\boldsymbol{z}_{\mathcal{I}} = \boldsymbol{X}_{.,s} \vee \boldsymbol{X}_{.,s+1} \vee ... \vee \boldsymbol{X}_{.,e}$, with $1 \leq s \leq e \leq p$, as illustrated Figure 2.3. This method's main assets are (i) the possibility to perform an exhaustive search among all contiguous intervals. The algorithm automatically finds the starting and end positions of the significantly associated intervals, without requiring any prior knowledge on the length or position of the associated regions. (ii) The method leverages the aggregation of weak effects to improve statistical power. (iii) Finally, it is able to properly correct for multiple hypothesis testing, as we will explain in the following section (2.3.1).



y	$X_{.,1}$	$X_{.,2}$	$X_{.,3}$	$X_{.,4}$	$X_{.,5}$	$X_{.,6}$	$X_{.,7}$	$X_{.,8}$	$X_{.,9}$	$X_{.,10}$	$z_{\mathcal{I}_1}$	$z_{\mathcal{I}_2}$
0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	1	0	1	0	0	1	1	1	1
0	0	0	1	0	0	0	1	0	0	1	1	1
0	1	0	1	0	1	0	1	0	0	0	1	1
0	1	0	1	0	1	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	1	0	1	1
1	0	0	0	0	1	0	0	1	0	0	0	1
1	0	0	0	0	1	1	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	1	1	0	1
1	1	0	1	0	0	0	0	0	0	0	1	0
1	0	0	0	0	1	1	0	0	1	1	0	1

$$\boldsymbol{z}_{\mathcal{I}_1} = \boldsymbol{X}_{.,2} \vee \boldsymbol{X}_{.,3} \vee \boldsymbol{X}_{.,4}$$
$$\boldsymbol{z}_{\mathcal{I}_2} = \boldsymbol{X}_{.,7} \vee \boldsymbol{X}_{.,8} \vee \boldsymbol{X}_{.,9} \vee \boldsymbol{X}_{.,10}$$

FIGURE 2.3: **Examples of genomic intervals**. Two genomic intervals and their encoding are represented in blue and red in the rightmost part of the figure. The encoding of a genomic interval takes the OR operation of the SNPs contained in the interval. In this thesis, the terms genomic interval and genomic region are going to be used interchangeably.

2.3 STATE OF THE ART IN SIGNIFICANT PATTERN MINING

In this section, we describe the necessary background on which the methods we propose are built. First, in Section 2.3.1, we present the multiple hypothesis testing problem and some state-of-the-art solutions. Genome-wide association studies lead to the multiple hypothesis testing problem as, typically, the datasets contain a large number of features, and the associations of every feature with the label are being tested individually. Second, in Section 2.3.2, we describe the effect of confounding factors on GWAS results and several approaches that have been used to correct for it when performing hypothesis testing.

2.3.1 *Multiple hypothesis testing correction*

Notation: In this section and the next one, we will use the notation provided Section 2.2 and the following notation. We denote δ the significance threshold under which we consider an association significant, δ_M the significance threshold under the multiple hypothesis testing correction method M and α the multiple hypothesis testing threshold target error. FP is the total number of false positives and TP the total number of true positives. As previously, we denote by $\mathcal{S}$ a **feature subset**, also called **pattern** in data mining, and $\mathbf{z}_{\mathcal{S}}$ the corresponding **feature combination** that encodes as a vector the feature subset. $\mathcal{P}$ describes the set of all feature subsets.

2.3.1.1 *Problem statement*

The multiple hypothesis testing problem [100] arises when several statistical tests are conducted in parallel and is exacerbated when the number of statistical tests performed is high, as seen in the introduction. It can result in a large number of false positives. In order to control for the number of false positives in multiple testing, frequentist practitioners often use metrics such as the Family-Wise Error Rate (FWER) or the False Discovery Rate (FDR) [101]. The FWER is the probability that at least one significant association is a false positive and can be written as: $P(\text{FP} > 0)$ where FP corresponds to the number of false positives. The false discovery proportion (FDP) is the proportion of false discoveries among all discoveries. Controlling its expectation, commonly referred as the false discovery rate (FDR), is given by $\mathbb{E}(\frac{FP}{\max(FP+TP,1)})$ and is a popular, less stringent alternative to the FWER control. In both cases, we aim to control these rates with a threshold α, such that $\text{FWER} \leq \alpha$ or $\text{FDR} \leq \alpha$.

In GWAS, both metrics can be used to control the error rate in multiple testing. While FDR has the main advantage over FWER that it is less conservative, by being more flexible in the number of false positives allowed, therefore leading to a higher statistical power, we chose to focus the study on the FWER for several reasons. First, controlling the FWER allows to guarantee that the probability of any false discoveries is upper bounded, which gives a meaningful measure of confidence independent of the number of total discoveries. Moreover, as we will see in the next section (2.3.1.3), existing algorithms that apply multiple hypothesis testing correction to test feature interactions in significant pattern mining aim to control the FWER. Additionally, controlling the FWER does not require any hypothesis on the joint distribution of the tests statistics performed. As encoded interactions of features can be highly correlated between each other, the respective statistical tests are very likely to also present strong correlations, therefore making this last property extremely important when working with interactions of features. In contrast, many FDR-controlling procedures require either independence on certain restricted forms of dependence, where validity is hard to verify in the context of interaction search.

2.3.1.2 *Bonferroni's family-wise error rate estimate*

As there does not exist any general closed-form expression for the FWER, the Bonferroni estimate $\widehat{\text{FWER}}_{bonf}$ is often used instead [51]. As $\widehat{\text{FWER}}_{bonf}$ is an upper bound of the true FWER, controlling the Bonferroni FWER estimate at level α implies that the true FWER is also controlled at level α, i.e. $\alpha \geq \widehat{\text{FWER}}_{bonf} \geq \text{FWER}$. In practice, the Bonferroni estimate is equal to $\widehat{\text{FWER}}_{bonf} = \delta \times p'$ where δ is the significance threshold per p-value and p' is the total number of tests performed. Therefore, applying a Bonferroni correction is equivalent to fixing the significance threshold δ to $\delta_{bonf} = \alpha / p'$. For example, if we have $p = 100$ features and test all $p' = |\mathcal{P}| = 2^p - 1 \approx 10^{30}$ interactions of features, if we wish to control the FWER at level $\alpha = 0.05$, the adjusted significance threshold would be $\delta_{bonf} \approx 5 \times 10^{-32}$, which can be overly stringent and lead to a significant loss of power. For this reason, several significant pattern mining methods restrict the search space by implicit or explicit constraints which results in a less stringent Bonferroni correction, as the number of tested patterns are reduced [102–104].

2.3.1.3 *Tarone's family-wise error rate estimate*

Another less conservative approach was introduced by Tarone [105] and consists of an adjusted Bonferroni correction for discrete test statistics. As the novel algorithms described in this manuscript build on [105], we detail Tarone's process hereafter.

Tarone primarily applied his method to univariate association tests, therefore in this section we only consider single features. Let a feature Z and a label Y be two binary random variables for which we observe n realisations $\{(z_i, y_i)\}_{i=1}^{n}$. It is possible to build a contingency table, as follows:

	y=1	y=0	total
$z = 1$	a	$x - a$	x
$z = 0$	$n_1 - a$	$n - x - n_1 + a$	$n - x$
total	n_1	$n - n_1$	n

In the contingency table, n corresponds to the total number of samples, n_1 is the number of samples whose label takes value 1, x the support of the feature, i.e. the total number of samples for which the feature of interest takes value 1, a the number of samples that belong to the positive class and for which the feature of interest $\mathbf{z}$ takes value 1. To test the null hypothesis $H_0 : Y \perp\!\!\!\perp Z$, a test statistic is chosen and a p-value is computed. Test statistics that fit Tarone's framework are, for example, Pearson's χ^2-test, the Mann-Whitney U test or Fisher's exact test. For illustration purposes, we use the latter throughout this section. The test statistic is therefore a and a corresponding p-value, in the case of a two-sided test, can be written as: $p(a|x, n_1, n) = \sum_{a' \in \mathcal{A}} Pr(a'|x, n_1, n) = \sum_{a' \in \mathcal{A}} \frac{C_{n_1}^{a'} C_{n-n_1}^{x-a'}}{C_n^{n-x}}$ with $\mathcal{A} = \{a' | Pr(a|x, n_1, n) \geq Pr(a'|x, n_1, n)\}$. If the p-value is smaller than a significance threshold δ, then the association between Z and Y is deemed significant. Notably, Tarone made the observation that there exists a **minimum attainable p-value** for each contingency table that only depends on the margins of the contingency table (n, x and n_1). Given the margins, the test statistic a can only take values in $[\![a_{min}, a_{max}]\!] = [\![\max(0, x + n_1 - n), \min(x, n_1)]\!]$. Therefore, at fixed margins, a p-value can take at most $a_{max} - a_{min} + 1$ different values. Given these observations, one can compute a minimum attainable p-value as $\Psi(x, n_1, n) = \min\{p(k|x, n_1, n) | k \in [\![a_{min}, a_{max}]\!]\}$. When studying a dataset with multiple features instead of one as in this preliminary example, a contingency table and a minimum attainable p-value can be computed for each feature that is tested.

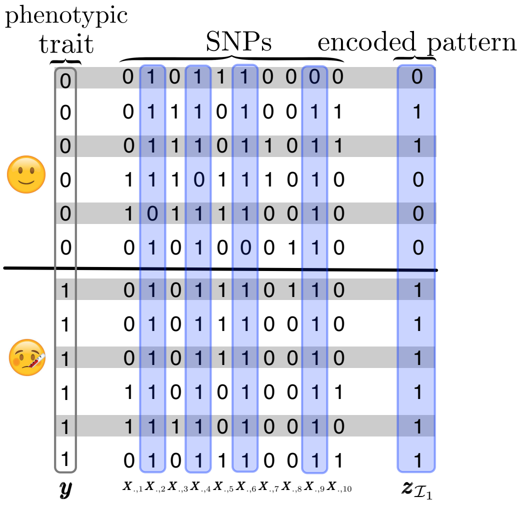
The concept of minimum attainable p-value has important implications in multiple hypothesis testing for discrete test statistics. The minimum attainable p-value quantifies the strongest association possible given the number of samples n, the number of positive cases in $\boldsymbol{y}$ and the number of samples for which $\mathbf{z}$ is active. Comparing the minimum attainable p-value to the significance threshold δ allows to quantify how strong the association could be given the margins. Therefore, if the strongest association possible were not significant, i.e. if $\Psi(x, n_1, n) > \delta$, any other outcome given the same margins could also not be significant, i.e. $p(a|x, n_1, n) \geq \Psi(x, n_1, n) > \delta$, cannot regardless of the value of the test statistic a.

More importantly, as the minimum attainable p-values are only functions of the margins of the contingency table and not of the labels $\boldsymbol{y}$, it is possible to use them in order to prune features that cannot be significant. This property is key in the context of multiple hypothesis testing correction: as features that are pruned away have a minimum attainable p-value larger than the significance threshold, they cannot be deemed significant, they cannot be false positives and finally do not need to be accounted for in the FWER. Therefore the feature is said to be **untestable** and can be ignored. By contrast, if the minimum attainable p-value of a feature is smaller than the significance threshold, the feature is considered **testable**. Therefore, the total number of tests performed $|\mathcal{P}_{tar}(\delta)|$, equal to the number of testable features, is reduced and verifies $|\mathcal{P}_{tar}(\delta)| \leq |\mathcal{P}|$. As a consequence, the Bonferroni correction is modified and the FWER estimate becomes $\widehat{\text{FWER}}_{tar} = \delta_{tar}|\mathcal{P}_{tar}(\delta_{tar})|$. Tarone leverages the reduction in statistical tests performed, $|\mathcal{P}_{tar}(\delta_{tar})| \leq |\mathcal{P}|$, to obtain an adjusted significance threshold δ_{tar} that is larger than Bonferroni's adjusted significance threshold δ_{bonf} and *de facto* leads to a gain in statistical power. To summarise, Tarone's statistical framework uses properties of some discrete statistics to increase the statistical power in the context of multiple hypothesis testing, while controlling for the FWER.

However, as the number of testable features is a function of the significance threshold δ, the calculation of δ_{tar} is less straightforward than in Bonferroni's correction, but can be expressed as $\delta_{tar} = \max\{\delta \,|\, \delta \times |\mathcal{P}_{tar}(\delta)| <= \alpha\}$. In order to find δ_{tar}, Tarone suggests to compute all the maximum achievable significance levels (the test statistics that results into the minimum attainable p-value defined above) and to incrementally increase the correction factor of the significance threshold, until, increasing one more time the correction factor leads to a number of tests performed (equivalently testable features) smaller than the correction factor.

While Tarone's framework allowed to use a less conservative Bonferroni's correction, Tarone's method (i) did not consider higher-order interactions, which would be the regime where the method is most advantageous, i.e. where the number of tests $|\mathcal{P}|$ is *a priori* very large and (ii) computed δ_{tar} in a brute-force manner, which does not scale to a large number of tests. Next, we will present an algorithm developed more recently that tackled both limitations successfully.

2.3.1.4 *Tarone's adjusted significance threshold in the case of higher-oder interactions with the* `LAMP` *algorithm*



$\boldsymbol{y}$	$\boldsymbol{X}_{.,1}$	$\boldsymbol{X}_{.,2}$	$\boldsymbol{X}_{.,3}$	$\boldsymbol{X}_{.,4}$	$\boldsymbol{X}_{.,5}$	$\boldsymbol{X}_{.,6}$	$\boldsymbol{X}_{.,7}$	$\boldsymbol{X}_{.,8}$	$\boldsymbol{X}_{.,9}$	$\boldsymbol{X}_{.,10}$	$\boldsymbol{z}_{\mathcal{I}_1}$
0	0	1	0	1	1	1	0	0	0	0	0
0	0	1	1	1	0	1	0	0	1	1	1
0	0	1	1	1	0	1	1	0	1	1	1
0	1	1	1	0	1	1	1	0	1	0	0
0	1	0	1	1	1	1	0	0	1	0	0
0	0	1	0	1	0	0	0	1	1	0	0
1	0	1	0	1	1	1	0	1	1	0	1
1	0	1	0	1	1	1	0	0	1	0	1
1	0	1	0	1	1	1	0	0	1	0	1
1	1	1	0	1	0	1	0	0	1	1	1
1	1	1	1	1	0	1	0	0	1	0	1
1	0	1	0	1	1	1	0	0	1	1	1

$$z_{\mathcal{S}_1} = \boldsymbol{X}_{.,2} \wedge \boldsymbol{X}_{.,4} \wedge \boldsymbol{X}_{.,6} \wedge \boldsymbol{X}_{.,9}$$

FIGURE 2.4: **Example of a feature subset**. A subset of features and its encoding $\boldsymbol{z}_{\mathcal{S}_1}$ is represented in blue in the rightmost part of the figure. The encoding of the feature subset takes the AND operation of the features vectors it includes.

Terada [54] introduces the Limitless Arity Multiple-testing Procedure (`LAMP`) algorithm, which uses Tarone's statistical framework in the context of significant pattern mining. Later, [55] proposes a new `LAMP` algorithm that is much faster than the original one. In the following paragraphs, we will focus the description on this second, faster version. The `LAMP` algorithm efficiently finds δ_{tar} and all significantly associated feature combinations in a dataset composed of binary variables and label. In the publication, features $\{\boldsymbol{X}_{.,i}\}_{i=1}^{n}$ have a binary encoding and feature subsets $\mathcal{S}$ are summarised using the logical AND operation on the feature vectors that compose $\boldsymbol{z}_{\mathcal{S}}$, such as shown Figure 2.4. For each sample i, the element $z_{\mathcal{S},i}$ indicates whether all features contained in $\mathcal{S}$ are active in sample i ($z_{\mathcal{S},i} = 1$) or not ($z_{\mathcal{S},i} = 0$). We note the set of feature subsets $\mathcal{P}$. The total number of features being p, there is a total of $|\mathcal{P}| = 2^p - 1$ non-empty feature subsets. `LAMP` uses Fisher's exact test to quantify the association between feature combinations and the binary label, but could be applied to the χ^2-test and to the Mann-Whitney U test. The pseudo-code of the `LAMP` algorithm is presented in Algorithms 1 and 2.

Algorithm 1 `LAMP`

Input: Dataset $\mathcal{D} = \{(\boldsymbol{X}, \boldsymbol{y})\}$, target FWER $= \alpha$
Output: $\{\mathcal{S} | p_{\mathcal{S}} \leq \delta_{tar}\}$

```
1: Initialise global variables $\delta_{tar} = 1$ and $\mathcal{P}_{tar}(\delta_{tar}) = \emptyset$
2: tarone($\emptyset$)
3: Return $\{\mathcal{S} \in \mathcal{P}_{tar}(\delta_{tar}) | p_{\mathcal{S}} \leq \delta_{tar}\}$
```

Algorithm 2 `tarone`

Input: Current subset of feature being processed $\mathcal{S}$
Output: Adjusted significance threshold δ_{tar}, set of testable feature subsets $\mathcal{P}_{tar}(\delta_{tar})$

```
if is_testable($\mathcal{S}, \delta_{tar}$) then
    Append $\mathcal{S}$ to $\mathcal{P}_{tar}(\delta_{tar})$
    $\widehat{\text{FWER}}_{tar}(\delta_{tar}) \leftarrow \delta_{tar} |\mathcal{P}_{tar}(\delta_{tar})|$
    while $\widehat{\text{FWER}}_{tar}(\delta_{tar}) > \alpha$ do
        Decrease $\delta_{tar}$
        $\mathcal{P}_{tar}(\delta_{tar}) \leftarrow \{\mathcal{S} \in \mathcal{P}_{tar}$ : is_testable($\mathcal{S}, \delta_{tar}$)$\}$
        $\widehat{\text{FWER}}_{tar}(\delta_{tar}) \leftarrow \delta_{tar} |\mathcal{P}_{tar}(\delta_{tar})|$
    end while
end if
if not is_prunable($\mathcal{S}, \delta_{tar}$) then
    for $\mathcal{S}' \in$ Children($\mathcal{S}$) do
        tarone($\mathcal{S}'$)
    end for
end if
```

The core operation of `LAMP`, described Algorithm 1, is Line 2, an efficient implementation of the routine `tarone`, detailed in Algorithm 2, that computes δ_{tar} and the corresponding set of testable patterns. This is followed by Line 3, that evaluates the statistical association of the feature combination $\boldsymbol{z}_{\mathcal{S}}$ of each testable feature subset $\mathcal{S} \in \mathcal{P}_{tar}(\delta_{tar})$ with the class labels $\boldsymbol{y}$. The routine `tarone` uses a branch-and-bound approach to efficiently compute Tarone's corrected significance threshold δ_{tar} and the set of testable feature subsets $\mathcal{P}_{tar}(\delta_{tar})$. This processes one subset $\mathcal{S}$ at a time. The subsets are arranged in a tree as in Figure 2.5, such that parents are subsets of their descendants, i.e. $\mathcal{S}' \in \text{Children}(\mathcal{S}) \implies \mathcal{S} \subset \mathcal{S}'$. The tree is explored with a depth-first search approach with the enumeration scheme proposed in [55, 106].

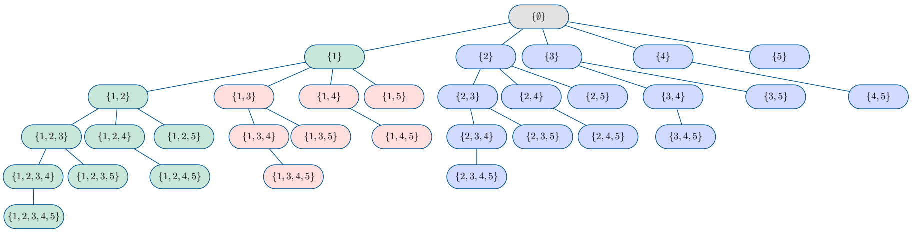


FIGURE 2.5: **Example of a depth-first search tree**. The dataset used to build the tree contains five features. Each node is a subset $\mathcal{S}$, represented by the set of indices of the features that are contained in the feature subset. Grey nodes are traversed first, then green, red and blue ones.

Before invoking `tarone`, in Line 1 of Algorithm 2 the significance threshold δ_{tar} is initialised to 1, the largest value it can take, and the set of testable feature subsets $\mathcal{P}_{tar}(\delta_{tar})$ is initialized to the empty set. The enumeration procedure is started by calling `tarone` with the empty feature subset $\mathcal{S} = \emptyset$, which acts as the root of the enumeration tree. All $2^p - 1$ non-empty feature subsets will then be explored recursively by traversing the enumeration tree depth-first. Each time a feature subset $\mathcal{S}$ is visited, Line 1 checks if the combination $\mathbf{z}_{\mathcal{S}}$ is testable. If it is testable, the feature subset is appended to the set of testable feature subsets (Line 2) and the estimated FWER is recomputed (Line 3). The FWER condition for Tarone's testability criterion is checked in Line 4. If it is violated, the threshold δ is decreased incrementally (Line 5) requiring to decrease the number of testable feature subsets $\mathcal{P}_{tar}(\delta_{tar})$ (Line 6) and to reevaluate the FWER estimation (Line 7). This step is repeated until the FWER condition of testability is again verified. Before continuing the traversal of the tree by exploring the children of the current feature subset $\mathcal{S}$, Line 8 checks if the pruning criterion applies. Only if it does not apply are all children of $\mathcal{S}$ visited recursively in Lines 9 and 10. The testability and pruning conditions in Lines 1 and 8 become more stringent as δ_{tar} decreases. Because of this, as δ_{tar} decreases along the enumeration procedure (Line 5), increasingly larger parts of the search space are pruned. Thus, the algorithm terminates when, for the current value of δ_{tar} and $\mathcal{P}_{tar}(\delta_{tar})$, all feature subsets that cannot be pruned have been visited.

Without the pruning criterion Line 8, all the subsets $\mathcal{S}$ would need to be enumerated and processed, which scales exponentially with p and therefore is a severe computational bottleneck. The pruning criterion allows to limit the number of subsets $\mathcal{S}$ that are being processed by inferring the testability properties of children of the feature subset being visited. To this end, the pruning criterion exploits the fact that the minimum attainable p-value function $\Psi(\mathcal{S})$ obeys a simple monotonicity property: $\mathcal{S} \subseteq \mathcal{S}' \implies \Psi(\mathcal{S}) \leq \Psi(\mathcal{S}')$ provided that the support of the feature subset $x_{\mathcal{S}} \leq \min(n_1, n - n_1)$. This leads to a remarkably simple pruning criterion: if a feature subset $\mathcal{S}$ is non-testable, i.e. $\Psi(\mathcal{S}) > \delta_{tar}$, and its support $x_{\mathcal{S}}$ is smaller or equal to $\min(n_1, n - n_1)$, then all children $\mathcal{S}'$ of $\mathcal{S}$, which satisfy $\mathcal{S} \subset \mathcal{S}'$ by construction of the enumeration tree, will also be non-testable and can be pruned from the search space.

To conclude, the algorithm Limitless Arity Multiple-testing Procedure is able to find all feature interactions in a binary dataset, which are significantly associated with a binary class-label, while controlling for the number of false positives with Tarone's FWER estimate, therefore obtaining a higher statistical power than a simple Bonferroni correction. Since then, alternative methods have been developed [106], in particular an extension of `LAMP` that explores permutation testing to exploit redundancy between tested patterns (as patterns are sub- or super- patterns of other ones) and gain statistical power.

2.3.1.5 *`FAIS` as first application of Tarone's FWER estimate to GWAS datasets*

The method `FAIS` implements a modification of `LAMP` in order to efficiently find significant contiguous genomic regions in GWAS, i.e. genomic intervals or regions such that the occurrence of a least one of its variants encoded as a 1 (for instance a minor allele or recessive genotype) is statistically significantly associated with the occurrence of a phenotype of interest. The main novelties introduced in FAIS are the following:

1. the regions are encoded as: $z_{\mathcal{I}} = \mathbf{X}_{.,s} \vee \mathbf{X}_{.,s+1} \vee ... \vee \mathbf{X}_{.,e}$. The objective of this encoding is to account for genetic heterogeneity, by grouping together SNPs that would have a too small effect to be detected alone but that are all associated to the same phenotype, under the assumption that they play a sufficiently similar role in giving rise to the phenotype.

2. the regions that are parsed are contiguous intervals of SNPs instead of any subset of SNPs, as the authors were interested in local regions of genetic heterogeneity, and this greatly enhances scalability, albeit that the cost of generality.

3. the intervals are visited according to a breadth-first search arrangement as shown Figure 2.6, instead of a depth-first search one, which allows making a more efficient use of pruning while being feasible, as the memory requirements are drastically less stringent when testing intervals compared to all feature subsets.

The intervals found by this algorithm share promising candidates for regions of genetic heterogeneity underlying phenotypic variation and aim to be functionally investigated.

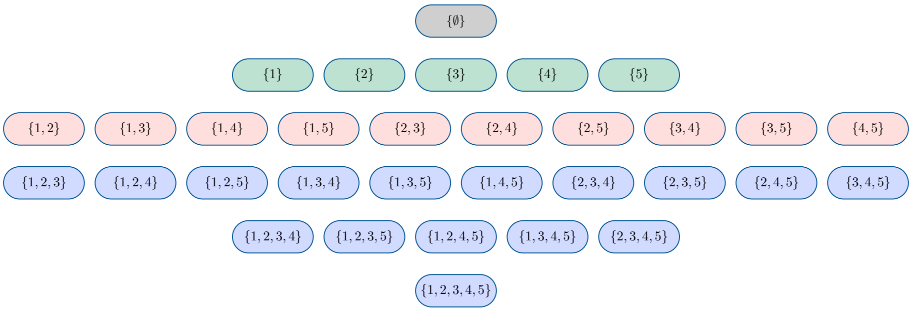


FIGURE 2.6: **Example of a breadth-first search tree**. The dataset used to build the tree contains five features. Each node is a subset $\mathcal{S}$, represented by the set of indices of the features that are contained in the feature subset. Grey nodes are traversed first, then green, red and blue ones.

2.3.2 *Correction for confounders*

Methods that correct for confounding effects are fundamental in GWAS analyses. If unaccounted for, one may find many false positives that are actually associated with a covariate and not with the phenotypic trait of interest [107]. Formally, let X be a regressor, Y a phenotypic trait and C a confounding covariate, such that $X \not\perp\!\!\!\perp Y$ but $X \perp\!\!\!\perp Y|C$ as illustrated Figure 2.7. In this case, we would like that the statistical framework treats X as false positive. For example, in clinical case/control association studies, it is common to search for subsets of genetic variants that are associated with a disease of interest. In this setting, the class labels are the health status of individuals, e.g. sick or healthy, and the features represent binary genetic variants. However, it can often be the case that the studied samples belong to several subpopulations or have different physical attributes, which moreover show differences in the prevalence of some altered genetic variants. When, additionally, these clusters are unevenly distributed across classes, it can result in false associations to the disease of interest [28]. As confounding factors are commonly present in GWAS data and practitioners aim to have a precise control of the number of false positives, it is necessary to model ancestry differences between cases and controls in the presence of population structure or to correct for any additional confounding covariate when necessary.

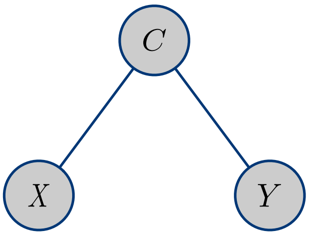


FIGURE 2.7: **Graphical model**. $X \not\perp\!\!\!\perp Y$ but $X \perp\!\!\!\perp Y|C$

In GWAS, the confounding effect is often measured with the genomic inflation factor λ [108]. This parameter is equal to the ratio between the empirical median of the test statistic divided by the theoretical median, under the global null hypothesis in a theoretical unconfounded analysis. Thus, the medians of the empirical and theoretical test statistic for the association between the genetic variants and the phenotype are taken as representatives of the two distributions. $\lambda \approx 1$ is interpreted as the two distributions being close to identical while a medium to strong divergence, i.e. $\lambda > 1$ or $\lambda < 1$, means that the two distributions are far apart. In the case of GWAS, the theoretical median under the global null is a reasonable approximation under the conservative assumption that very few genetic variants are associated to the phenotype of interest compared to the total count of genetic variants. As a consequence, an empirical median much larger than the theoretical one would indicate that many more variants than expected show some association to the phenotype. A factor that can explain genomic inflation is the presence of confounder. Therefore, an inflated genomic factor λ is often used as proxy for confounding effects on the statistical tests, effects that are potentially responsible for an unwanted large number of false positives.

Common methods that enable to reduce genomic inflation are: **regression models** to which regressors representing confounding effects are adjoined, either as continuous, discretised or dummy variables, **stratification methods**, using for example the Cochran-Mantel Hanzel (CMH) test [43, 44] and **linear-mixed models** (LMM) where confounding effects are modelled as a random effect following a gaussian distribution [29, 108, 109].

In regression models, the covariates are known and modelled explicitly as additional fixed effects. We would write such a model as: $\boldsymbol{y} = \beta_0 + \sum_{i=1}^{p} \beta_i \boldsymbol{X}_{.,i} + \sum_{i=1}^{c} \gamma_i \boldsymbol{C}_{.,i}$, where the vectors $\boldsymbol{X}_{.,i}$ correspond to genetic variants, $\boldsymbol{C}_{.,i}$ to covariates and c to the number of covariates. Most often, univariate t-tests under a FWER control are used and lead to adjusted odds ratios or p-values of the features of interest that measure the impact of the features on the phenotype after accounting for confounding factors. Additionally, multivariate regularised linear or logistic regressions are also common tools, with the downside that p-values are not commonly provided in this case as it would require to account for the use of labels in the variable selection step.

Another approach is to stratify the data according to the covariate of interest, which requires the covariate to be categorical, and then use a conditional test. For example, let $\boldsymbol{z}$ be n binary feature realisations, $\boldsymbol{y}$ n binary phenotype realisations and $\boldsymbol{c}$ the categorical covariate we would like to correct for. Let us further assume that the covariate has k categories. A conditional test that is adapted to this setting is the Cochran-Mantel-Haenszel (CMH) test, which requires to compute k contingency tables, one for each category of the covariate. Let $j \in \{1,...,k\}$, for each contingency table we denote c_j the value of the covariate, n_j the total number of samples with covariate value c_j and, among these samples, $n_{1,j}$ the number of samples that belong to the positive class and x_j the number of samples whose feature $z_i = 1$. Additionally, a_j corresponds to the number of samples that belong to the positive class and for which $z_i = 1$. Based on these values $\{n_j, n_{1,j}, x_j, a_j\}_{j=1}^k$, the p-value for feature $\boldsymbol{z}$ can be computed as:

$$p(\{n_j, n_{1,j}, x_j, a_j\}_{j=1}^k) = 1 - F_{\chi_1^2}\left(\frac{\left(\sum_{j=1}^k a_j - \frac{x_j n_{1,j}}{n_j}\right)^2}{\sum_{j=1}^k \frac{n_{1,j}}{n_j}(1 - \frac{n_{1,j}}{n_j})x_j(1 - \frac{x_j}{n_j})}\right) \tag{2.1}$$

where $F_{\chi_1^2}$ is the distribution function of a χ^2 random variable with 1 degree of freedom. Finally, the feature $\boldsymbol{z}$ will be deemed significantly associated to the binary label if the p-value $p(\{n_j, n_{1,j}, x_j, a_j\}_{j=1}^k)$ falls below a significance threshold δ, that is, if $p(\{n_j, n_{1,j}, x_j, a_j\}_{j=1}^k) \leq \delta$. As illustrated, the CMH test can be understood as a form of meta-analysis applied to k disjoint datasets $\{\mathcal{D}_j\}_{j=1}^k$, where $\mathcal{D}_j$ contains only observations for which the covariate variable $\boldsymbol{c}$ takes value c_j. For confounded features, the association might be large in the entire dataset $\mathcal{D}$, but small for conditional datasets $\mathcal{D}_j$. Thus, the CMH test will not deem such features significant.

As a third option, in linear-mixed models, the confounding effects are modelled as random effects which do not need to be observed directly. Formally, the dependence between the phenotypic trait and the regressors can be written as the sum of genetic effects and confounding influences, $\boldsymbol{y} = \beta_0 + \sum_{i=1}^p \beta_i \boldsymbol{X}_{.,i} + \boldsymbol{u}$, where the vectors $\boldsymbol{X}_{.,i}$ correspond to genetic variants and $\boldsymbol{u}$ to the confounding influence. The distribution of $\boldsymbol{u}$ is assumed Gaussian $\boldsymbol{u} \sim \mathcal{N}(0, \sigma_g^2 \boldsymbol{K})$, with $\boldsymbol{K}$ the covariance of the data, which models deviations from the usual i.i.d. scenario arising due to confounding. While LMMs are very efficient at accounting for confounding effects both implicitly and explicitly, it is either used in univariate settings neglecting feature interactions, in set-based settings where defining the sets requires prior biological information [29], or in multivariate additive settings such as LMM-Lasso [109].

The results obtained by the three types of approaches reinforce the importance of confounding correction as they are shown to reduce the number of false positives in simulation settings and the genomic inflation in real data [110, 111]. However, no method that allows to apply fixed or random effects models, that scales to interactions of any order and corrects for the multiple hypothesis testing problem in that regime, exists yet. This leaves the combination of LAMP and CMH as most promising option, which did not exist at the time of the development of our models as the CMH test does not lead to a simple pruning criteria, unlike Fisher's exact test or the χ^2 test. The next sections will therefore be devoted to explain how to combine LAMP and CMH, resulting in the first method for interaction search with covariate correction.

2.4 significant combinations of features in the presence of categorical covariates

This section introduces the Fast Automatic Conditional Search (FACS) algorithm, the first approach that allows finding all interactions of binary features statistically associated with a binary label while correcting for a categorical covariate and controlling for the FWER. Until the development of our methods, it was not possible to use LAMP while correcting for confounding covariates, as the mathematical properties of Fisher's exact test or of the χ^2 test that enabled defining a computationally efficient pruning are not verified by conditional tests. Section 2.4.1 formalises the problem statement. Section 2.4.2 provides a high-level description of the algorithm. Then, Sections 2.4.3 and 2.4.4 detail the two key steps of FACS, which are also the main algorithmic contributions of this work. Finally, simulation experiments show the statistical and computational superiority of our method compared to state-of-the-art approaches in Section 2.4.5, and Section 2.4.6 demonstrates the usefulness of such method in a proof-of-concept GWAS experiment.

2.4.1 *FACS: main objective*

The main objective of the FACS algorithm can be summarised as:
Objective: Given a dataset $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{c})$, the goal of FACS is to:

1. Compute Tarone's corrected significance threshold δ_{tar}.
2. Retrieve all feature subsets $\mathcal{S}$ whose p-value $p_{\mathcal{S}}$ is below δ_{tar} when testing the null hypothesis H_0: $Z_{\mathcal{S}} \perp\!\!\!\perp Y|C$ of conditional independence given the covariates –rather than normal independence.

For both (1) and (2), the test statistic of choice will be the CMH test, which is a conditional test statistic for discrete values, thus allowing to correct for a confounding categorical covariate as described in Section 2.3.2. The key contribution of our work is to bridge the gap between Tarone's testability criterion and the CMH test. The resulting algorithm, described in Section 2.4.2, relies on two key novel theoretical results. In Section 2.4.3, we show for the first time that Tarone's method can be applied to the CMH test. More importantly, in Section 2.4.4, we introduce a novel branch-and-bound algorithm to efficiently compute δ_{tar} without requiring the function Ψ computing Tarone's minimum attainable p-value to be monotonic. This allows us not only to apply Tarone's testability criterion to the CMH test, but to do so as efficiently as existing methods not able to handle confounding covariates.

2.4.2 *FACS: high-level description and pseudocode*

As shown in the pseudocode in Algorithm 3, conceptually, FACS's structure is very similar to LAMP's, presented Section 2.3.1.4. FACSs performs two main operations.

Algorithm 3 FACS

```
Input: Dataset $\mathcal{D} = \{(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{c})\}$, target FWER $= \alpha$
Output: $\{\mathcal{S} | p_{\mathcal{S}} \leq \delta_{tar}\}$

1: Initialise global variables $\delta_{tar} = 1$ and $\mathcal{P}_{tar}(\delta_{tar}) = \emptyset$
2: tarone_cmh($\emptyset$)
3: Return $\{\mathcal{S} \in \mathcal{P}_{tar}(\delta_{tar}) | p_{\mathcal{S}} \leq \delta_{tar}\}$
```

Algorithm 4 tarone_cmh

```
Input: Current subset of features being processed $\mathcal{S}$
Output: Adjusted significance threshold $\delta_{tar}$, set of testable subsets $\mathcal{P}_{tar}(\delta_{tar})$

if is_testable_cmh($\mathcal{S}, \delta_{tar}$) then
    Append $\mathcal{S}$ to $\mathcal{P}_{tar}(\delta_{tar})$
    $\widehat{\text{FWER}}_{tar}(\delta_{tar}) \leftarrow \delta_{tar} |\mathcal{P}_{tar}(\delta_{tar})|$
    while $\widehat{\text{FWER}}_{tar}(\delta_{tar}) > \alpha$ do
        Decrease $\delta_{tar}$
        $\mathcal{P}_{tar}(\delta_{tar}) \leftarrow \{\mathcal{S} \in \mathcal{P}_{tar}$ : is_testable_cmh($\mathcal{S}, \delta_{tar}$)$\}$
        $\widehat{\text{FWER}}_{tar}(\delta_{tar}) \leftarrow \delta_{tar} |\mathcal{P}_{tar}(\delta_{tar})|$
    end while
end if
if not is_prunable_cmh($\mathcal{S}, \delta_{tar}$) then
    for $\mathcal{S}' \in$ Children($\mathcal{S}$) do
        tarone_cmh($\mathcal{S}'$)
    end for
end if
```

Before invoking tarone_cmh, first in Line 1 of Algorithm 3 the significance threshold δ_{tar} is initialized to 1, the largest value it can take, and the set of testable feature combinations $\mathcal{P}_{tar}(\delta_{tar})$ is initialised to the empty set. Second, Line 2 of Algorithm 3 invokes the routine tarone_cmh, described in Algorithm 4. This routine uses our novel branch-and-bound approach to efficiently compute Tarone's corrected significance threshold δ_{tar} and the set of testable feature subsets $\mathcal{P}_{tar}(\delta_{tar})$, when using the CMH test. Third, using the significance threshold δ_{tar} obtained in the previous step, Line 3 of Algorithm 3 evaluates the conditional association of the feature combination $\boldsymbol{z}_{\mathcal{S}}$ of each testable feature subset $\mathcal{S} \in \mathcal{P}_{tar}(\delta_{tar})$ with the class labels, given the categorical covariate, using the CMH test as shown in Section 2.3.2. Note that, according to Tarone's testability criterion, untestable feature subsets $\mathcal{S} \notin \mathcal{P}_{tar}(\delta_{tar})$ cannot be significant and therefore do not need to be considered in this step. Since in practice $|\mathcal{P}_{tar}(\delta_{tar})| \ll 2^p - 1$, the procedure tarone_cmh is the most critical part of FACS. The routine tarone_cmh uses the enumeration scheme first proposed in [55, 112]. All 2^p feature subsets are arranged in an enumeration tree such that $\mathcal{S}' \in Children(\mathcal{S}) \Rightarrow \mathcal{S} \subset \mathcal{S}'$. In other words, the children of a feature subset $\mathcal{S}$ in the enumeration tree are obtained by adding an additional feature to $\mathcal{S}$.

The enumeration procedure is started by calling `tarone_cmh` with the empty feature subset $\mathcal{S} = \emptyset$, which acts as the root of the enumeration tree. All $2^p - 1$ non-empty feature subsets will then be explored recursively by traversing the enumeration tree depth-first, as illustrated Figure 2.5. Every time a feature subset $\mathcal{S}$ in the tree is visited, Line 1 of Algorithm 4 checks if it is testable, as detailed in Section 2.4.3. If it is, $\mathcal{S}$ is appended to the set of testable feature subsets $\mathcal{P}_{tar}(\delta_{tar})$ in Line 2. Line 3, the FWER estimate is updated. The FWER condition for Tarone's testability criterion is checked in Line 4. If it is found to be violated, the significance threshold δ_{tar} is decreased in Line 5 until the condition is satisfied again, removing from $\mathcal{P}_{tar}(\delta_{tar})$ any feature subsets made untestable by decreasing δ_{tar} in Line 6 and re-evaluating the FWER condition accordingly in Line 7. Before continuing the traversal of the tree by exploring the children of the current feature subset $\mathcal{S}$, Line 8 checks if our novel pruning criterion applies, which is described in Section 2.4.4. Only if it does not apply are all children of $\mathcal{S}$ visited recursively in Lines 9 and 10. The testability and pruning conditions in Lines 1 and 8 become more stringent as δ_{tar} decreases. Because of this, as δ_{tar} decreases along the enumeration procedure (see Line 5), increasingly larger parts of the search space are pruned. Thus, the algorithm terminates when, for the current value of δ_{tar} and $\mathcal{P}_{tar}(\delta_{tar})$, all feature subsets that cannot be pruned have been visited. Despite the structural similarity with `LAMP`, both approaches differ drastically on the two most challenging steps of `FACS`, the design of (i) an appropriate testability criterion `is_testable_cmh` and (ii) an efficient, principled pruning criterion, `is_prunable_cmh`. These are crucial in overcoming the limitations of the current state of the art and be able to account for confounding in SPM. These are now each described in detail.

2.4.3 *`FACS`: testability criterion for the CMH test*

As mentioned in Section 2.3.1.3, Tarone's testability criterion has only been applied to test statistics such as Fisher's exact test, Pearson's χ^2 test and the Mann-Whitney U test, none of which allows incorporating covariates. However, the following proposition shows that the CMH test also has a minimum attainable p-value $\Psi_{cmh}(\mathcal{S})$:

Proposition 1 *The CMH test has a minimum attainable p-value $\Psi_{cmh}(\mathcal{S})$, which can be computed in $O(k)$ time as a function of the margins $\{n_j, n_{1,j}, x_{\mathcal{S},j}\}_{j=1}^{k}$ of the k 2×2 contingency tables.*

As detailed below, the proof of Proposition 1 involves showing that $\Psi_{cmh}(\mathcal{S})$ can be computed from the k 2×2 contingency tables corresponding to the feature combination $\mathbf{z}_{\mathcal{S}}$ by optimising the p-value $p_{\mathcal{S}}$ with respect to $\left\{a_{\mathcal{S},j}\right\}_{j=1}^{k}$ while keeping the table margins $\{n_j, n_{1,j}, x_{\mathcal{S},j}\}_{j=1}^{k}$ fixed.

PROOF: Equation 2.1 in Section 2.3.2 can be rewritten as:

$$\begin{aligned}
p_{\mathcal{S}} &= 1 - F_{\chi_1^2}\left(\frac{\left(\sum_{j=1}^{k} a_{\mathcal{S},j} - \frac{x_{\mathcal{S},j}n_{1,j}}{n_j}\right)^2}{\sum_{j=1}^{k}\frac{n_{1,j}}{n_j}\left(1-\frac{n_{1,j}}{n_j}\right)x_{\mathcal{S},j}\left(1-\frac{x_{\mathcal{S},j}}{n_j}\right)}\right) \\
&= 1 - F_{\chi_1^2}\left(\frac{\left(a_{\mathcal{S},tot} - \sum_{j=1}^{k}\frac{x_{\mathcal{S},j}n_{1,j}}{n_j}\right)^2}{\sum_{j=1}^{k}\frac{n_{1,j}}{n_j}\left(1-\frac{n_{1,j}}{n_j}\right)x_{\mathcal{S},j}\left(1-\frac{x_{\mathcal{S},j}}{n_j}\right)}\right) \\
&= 1 - F_{\chi_1^2}(T_{\mathcal{S}}(a_{\mathcal{S},tot}, \mathbf{x}_{\mathcal{S}})) \qquad (2.2)
\end{aligned}$$

where $a_{\mathcal{S},tot} = \sum_{j=1}^{k} a_{\mathcal{S},j}$ and $\mathbf{x}_{\mathcal{S}} = (x_{\mathcal{S},1}, \ldots, x_{\mathcal{S},k})$. Because $F_{\chi_1^2}(\cdot)$ is a monotonically increasing function of its argument $T_{\mathcal{S}}(a_{\mathcal{S},tot}, \mathbf{x}_{\mathcal{S}})$, $p_{\mathcal{S}}$ is minimized when $T_{\mathcal{S}}(a_{\mathcal{S},tot}, \mathbf{x}_{\mathcal{S}})$ is maximized. $T_{\mathcal{S}}(a_{\mathcal{S},tot}, \mathbf{x}_{\mathcal{S}})$ depends on $a_{\mathcal{S},tot}$ as a quadratic function with positive definite Hessian, hence, $p_{\mathcal{S}}$ is minimized as a function of $a_{\mathcal{S},tot}$ at the most extreme values $a_{\mathcal{S},tot}$ can attain. Since $a_{\mathcal{S},j} \in [\![a_{\mathcal{S},j,min}, a_{\mathcal{S},j,max}]\!]$ $\forall j \in [\![1,k]\!]$, with $a_{\mathcal{S},j,min} = \max(0, x_{\mathcal{S},j} - n_j + n_{1,j})$ and $a_{\mathcal{S},j,max} = \min(x_{\mathcal{S},j}, n_{1,j})$, we have $a_{\mathcal{S},min} \leq a_{\mathcal{S},tot} \leq a_{\mathcal{S},max}$, where $a_{\mathcal{S},min} = \sum_{j=1}^{k} a_{\mathcal{S},j,min}$ and $a_{\mathcal{S},max} = \sum_{j=1}^{k} a_{\mathcal{S},j,max}$. Thus:

$$\Psi_{cmh}(\mathcal{S}) = 1 - F_{\chi_1^2}\left(T_{\mathcal{S}}^{max}(\mathbf{x}_{\mathcal{S}})\right) \qquad (2.3)$$

with $T_{\mathcal{S}}^{max}(\mathbf{x}_{\mathcal{S}}) = \max\left(T_{\mathcal{S}}\left(a_{\mathcal{S},min}, \mathbf{x}_{\mathcal{S}}\right), T_{\mathcal{S}}\left(a_{\mathcal{S},max}, \mathbf{x}_{\mathcal{S}}\right)\right)$ satisfies $p_{\mathcal{S}} \geq \Psi_{cmh}(\mathcal{S})$, for all $\mathcal{S}$ that have the same margins. Also, $\Psi_{cmh}(\mathcal{S})$ as defined above depends only on $\{n_j, n_{1,j}, x_{\mathcal{S},j}\}_{j=1}^{k}$ and can be evaluated in $O(k)$ time, which completes the proof.

2.4.4 *FACS: pruning criterion for the CMH test*

State-of-the-art methods [54, 106], all of which are limited to unconditional association testing, exploit the fact that the minimum attainable p-value function $\Psi(\mathcal{S})$, using either Fisher's exact test or Pearson's χ^2 test on a single contingency table, obeys a simple monotonicity property: $\mathcal{S} \subseteq \mathcal{S}' \Rightarrow \Psi(\mathcal{S}) \leq \Psi(\mathcal{S}')$ provided that $x_{\mathcal{S}} \leq \min(n_1, n - n_1)$. This leads to a remarkably simple pruning criterion: if a feature subset $\mathcal{S}$ is non-testable, i.e. $\Psi(\mathcal{S}) > \delta$, and its support $x_{\mathcal{S}}$ is smaller or equal to $\min(n_1, n - n_1)$, then all children $\mathcal{S}'$ of $\mathcal{S}$, which satisfy $\mathcal{S} \subset \mathcal{S}'$ by construction of the enumeration tree, will also be non-testable and can be pruned from the search space. However, such a monotonicity property does **not** hold for the CMH minimum attainable p-value function $\Psi_{cmh}(\mathcal{S})$, severely constraining the development of an effective pruning criterion.

In the next section, we show how to circumvent this limitation by introducing a novel pruning criterion based on defining a monotonic **lower envelope** $\widetilde{\Psi}_{cmh}(\mathcal{S}) \leq \Psi_{cmh}(\mathcal{S})$ of the original minimum attainable p-value function $\Psi_{cmh}(\mathcal{S})$ and prove that it leads to a valid pruning strategy. Finally, in Section 2.4.4.2, we provide an efficient algorithm to evaluate $\widetilde{\Psi}_{cmh}(\mathcal{S})$ in $O(k \log k)$ time, instead of a naive implementation whose computational complexity would scale exponentially with k, the number of categories for the covariate.

2.4.4.1 *Definition and correctness of the pruning criterion*

As mentioned above, existing unconditional significant discriminative pattern mining methods only consider feature subsets $\mathcal{S}$ with support $x_{\mathcal{S}} \leq \min(n_1, n - n_1)$ to be **potentially prunable**. Analogously, we consider as potentially prunable the set of feature subsets $\mathcal{P}_P = \left\{ \mathcal{S} \mid x_{\mathcal{S},j} \leq \min(n_{1,j}, n_j - n_{1,j}),\ \forall\, j = 1, \ldots, k \right\}$. Note that for $k = 1$, our definition reduces to that of existing work. In pattern mining, a very large proportion of all feature subsets will have small supports. Therefore, restricting the application of the pruning criterion to potentially prunable feature subsets does not cause a substantial loss of performance in practice. We can now state the definition of the lower envelope for the CMH minimum attainable p-value:

Definition 1 *Let $\mathcal{S} \in \mathcal{P}_P$ be a potentially prunable feature subset. The lower envelope $\widetilde{\Psi}_{cmh}(\mathcal{S})$ is defined as $\widetilde{\Psi}_{cmh}(\mathcal{S}) = \min \{ \Psi_{cmh}(\mathcal{S}') \mid \mathcal{S}' \supseteq \mathcal{S} \}$.*

Note that, by construction, $\widetilde{\Psi}_{cmh}(\mathcal{S})$ satisfies $\widetilde{\Psi}_{cmh}(\mathcal{S}) \leq \Psi_{cmh}(\mathcal{S})$ for all feature subsets $\mathcal{S}$ in the set of potentially prunable feature subsets. Next, we show that unlike for the minimum attainable p-value function $\Psi_{cmh}(\mathcal{S})$, the monotonicity property holds for the lower envelope $\widetilde{\Psi}_{cmh}(\mathcal{S})$:

Lemma 1 *Let $\mathcal{S}, \mathcal{S}' \in \mathcal{P}_P$ be two potentially prunable feature subsets such that $\mathcal{S} \subseteq \mathcal{S}'$. Then, $\widetilde{\Psi}_{cmh}(\mathcal{S}) \leq \widetilde{\Psi}_{cmh}(\mathcal{S}')$ holds.*

PROOF: The statement follows directly from the definition of the lower envelope for the CMH minimum attainable p-value. We have $\widetilde{\Psi}_{cmh}(\mathcal{S}) = \min\limits_{\mathcal{S}'' \supseteq \mathcal{S}} \Psi_{cmh}(\mathcal{S}'')$ and $\widetilde{\Psi}_{cmh}(\mathcal{S}') = \min\limits_{\mathcal{S}'' \supseteq \mathcal{S}'} \Psi_{cmh}(\mathcal{S}'')$, respectively. Also, $\mathcal{S}'' \supseteq \mathcal{S}' \Rightarrow \mathcal{S}'' \supseteq \mathcal{S}$. Thus, the set of feature subsets over which $\Psi_{cmh}(\mathcal{S}'')$ is minimized to compute $\widetilde{\Psi}_{cmh}(\mathcal{S}')$ is a subset of the set of feature subsets over which $\Psi_{cmh}(\mathcal{S}'')$ is minimized to compute $\widetilde{\Psi}_{cmh}(\mathcal{S})$.

Next, we state the main result of this section, which establishes our search space pruning criterion:

Theorem 1 *Let $\mathcal{S} \in \mathcal{P}_P$ be a potentially prunable feature subset such that $\widetilde{\Psi}_{cmh}(\mathcal{S}) > \delta$. Then, $\Psi_{cmh}(\mathcal{S}') > \delta$ for all $\mathcal{S}' \supseteq \mathcal{S}$, i.e. all feature subsets containing $\mathcal{S}$ are non-testable at level δ and can be pruned from the search space.*

PROOF: Let $\mathcal{S}'$ be an arbitrary feature subset containing $\mathcal{S}$, i.e. $\mathcal{S}' \supseteq \mathcal{S}$. Then we have $\Psi_{cmh}(\mathcal{S}') \geq \widetilde{\Psi}_{cmh}(\mathcal{S}') \underset{Lemma\,1}{\geq} \widetilde{\Psi}_{cmh}(\mathcal{S})$. Therefore, $\widetilde{\Psi}_{cmh}(\mathcal{S}) > \delta \Rightarrow \Psi_{cmh}(\mathcal{S}') > \delta$. This proves that all feature subsets containing $\mathcal{S}$ are non-testable at level δ.

Moreover, since during the enumeration procedure described in Algorithm 4, the significance threshold δ can only decrease, those feature subsets can be pruned from the search space.

To summarize, the pruning criterion `is_prunable_cmh` in Line 10 of Algorithm 4 evaluates to `True` if and only if $\mathcal{S} \in \mathcal{P}_P \Leftrightarrow x_{\mathcal{S},j} \leq \min(n_{1,j}, n_j - n_{1,j}) \, \forall \, j = 1, \ldots, k$ and $\widetilde{\Psi}_{cmh}(\mathcal{S}) > \delta_{tar}$.

2.4.4.2 *Evaluating the pruning criterion in $O(k \log k)$ time*

In `FACS`, the pruning criterion stated above will be applied to all enumerated feature subsets. Hence, it is mandatory to have an efficient algorithm to compute the lower envelope for the CMH minimum attainable p-value $\widetilde{\Psi}_{cmh}(\mathcal{S})$ for any potentially prunable feature subset $\mathcal{S} \in \mathcal{P}_P$.

As shown in the proof of Proposition 1, $\Psi_{cmh}(\mathcal{S})$ depends on the feature subset $\mathcal{S}$ through its k-dimensional vector of supports $\mathbf{x}_{\mathcal{S}} = (x_{\mathcal{S},1}, \ldots, x_{\mathcal{S},k})$. Also, the condition $\mathcal{S}' \supseteq \mathcal{S}$ implies that $x_{\mathcal{S}',j} \leq x_{\mathcal{S},j} \, \forall \, j = 1, \ldots, k$. As a consequence, one can rewrite Definition 1 as $\widetilde{\Psi}_{cmh}(\mathcal{S}) = \min\limits_{\mathbf{x}_{\mathcal{S}'} \leq \mathbf{x}_{\mathcal{S}}} \Psi_{cmh}(\mathbf{x}_{\mathcal{S}'})$, where the vector inequality $\mathbf{x}_{\mathcal{S}'} \leq \mathbf{x}_{\mathcal{S}}$ holds component-wise. Thus, naively computing $\widetilde{\Psi}_{cmh}(\mathcal{S})$ would require optimizing Ψ_{cmh} over a set of size $\prod_{j=1}^{k} x_{\mathcal{S},j} = O(m^k)$, where m is the geometric mean of $\left\{ x_{\mathcal{S},j} \right\}_{j=1}^{k}$. This scaling is clearly impractical, as even for moderate k it would result in an overhead large enough to outweigh the benefits of pruning.

Because of this, we proposed the last key part of `FACS`: an efficient algorithm which evaluates $\widetilde{\Psi}_{cmh}(\mathcal{S})$ in only $O(k \log k)$ time. We will arrive at our final result in two steps, contained in Lemma 2 and Theorem 2.

Lemma 2 *Let $\mathcal{S} \in \mathcal{P}_P$ be a potentially prunable feature subset. The optimum $\mathbf{x}^*_{\mathcal{S}'}$ of the discrete optimization problem $\min\limits_{\mathbf{x}_{\mathcal{S}'} \leq \mathbf{x}_{\mathcal{S}}} \Psi_{cmh}(\mathcal{S}')$ satisfies $x^*_{\mathcal{S}',j} = 0$ or $x^*_{\mathcal{S}',j} = x_{\mathcal{S},j}$ for each $j = 1, \ldots, k$.*

In short, Lemma 2 shows that the optimum $\mathbf{x}^*_{\mathcal{S}'} = \arg\min\limits_{\mathbf{x}_{\mathcal{S}'} \leq \mathbf{x}_{\mathcal{S}}} \Psi_{cmh}(\mathcal{S}')$ of the discrete optimization problem defining $\widetilde{\Psi}_{cmh}(\mathcal{S})$ is always a vertex of the discrete hypercube $[\![\mathbf{0}, \mathbf{x}_{\mathcal{S}}]\!]$. Thus, the computational complexity of evaluating $\widetilde{\Psi}_{cmh}(\mathcal{S})$ can be reduced from $O(m^k)$ to $O(2^k)$, where $m \gg 2$ for most feature subsets. Finally, building upon the result of Lemma 2, Theorem 2 below shows that one can in fact find the optimal vertex out of all $O(2^k)$ vertices in $O(k \log k)$ time.

Theorem 2 *Let $\mathcal{S} \in \mathcal{P}_P$ be a potentially testable feature subset and define $\beta^l_{\mathcal{S},j} = \frac{n_j - n_{1,j}}{n_j} \left(1 - \frac{x_{\mathcal{S},j}}{n_j}\right)$ and $\beta^r_{\mathcal{S},j} = \frac{n_{1,j}}{n_j} \left(1 - \frac{x_{\mathcal{S},j}}{n_j}\right)$ for $j = 1, \ldots, k$. Let π_l and π_r be permutations $\pi_l, \pi_r : [\![1, k]\!] \mapsto [\![1, k]\!]$ such that $\beta^l_{\mathcal{S},\pi_l(1)} \leq \ldots \leq \beta^l_{\mathcal{S},\pi_l(k)}$ and $\beta^r_{\mathcal{S},\pi_r(1)} \leq \ldots \leq \beta^r_{\mathcal{S},\pi_r(k)}$, respectively.*

Then, there exists an integer $\kappa \in [\![1,k]\!]$ *such that the optimum* $\mathbf{x}^*_{\mathcal{S}'} = \underset{\mathbf{x}_{\mathcal{S}'} \leq \mathbf{x}_{\mathcal{S}}}{\arg\min} \Psi_{cmh}(\mathcal{S}')$ *satisfies one of the two possible conditions: (I)* $x^*_{\mathcal{S}',\pi_l(j)} = x_{\mathcal{S},\pi_l(j)}$ *for all* $j \leq \kappa$ *and* $x^*_{\mathcal{S}',\pi_l(j)} = 0$ *for all* $j > \kappa$ *or (II)* $x^*_{\mathcal{S}',\pi_r(j)} = x_{\mathcal{S},\pi_r(j)}$ *for all* $j \leq \kappa$ *and* $x^*_{\mathcal{S}',\pi_r(j)} = 0$ *for all* $j > \kappa$*.*

The proofs of Lemma 2 and Theorem 2 are available in the Appendix, Sections A.1 and A.2. In summary, Theorem 2 above implies that the 2^k candidates to be the optimum $\mathbf{x}^*_{\mathcal{S}'}$ according to Lemma 2 can be narrowed down to only $2k$ vertices: k candidates satisfying the first condition and k the second condition. Moreover, evaluating Ψ_{cmh} for all k candidates satisfying the first condition (resp. the second condition) can be done in $O(k)$ time rather than $O(k^2)$. This is due to the fact that each of the k candidate vertices for each condition can be obtained by changing a single dimension with respect to the previous one. Therefore, the operation dominating the computational complexity is the sorting of the two k-vectors $(\beta^l_{\mathcal{S},1}, \dots, \beta^l_{\mathcal{S},k})$ and $(\beta^r_{\mathcal{S},1}, \dots, \beta^r_{\mathcal{S},k})$. As a consequence, the runtime required to evaluate the lower envelope $\widetilde{\Psi}_{cmh}(\mathcal{S})$, and thus our novel pruning criterion `is_prunable_cmh`, scales as $O(k \log k)$ with the number of categories of the covariate.

2.4.4.3 *Pseudocode of the pruning criterion*

Algorithm 5 `is_prunable_cmh`

Input: Current subset of features being processed $\mathcal{S}$ and corresponding k-dimensional vectors $\{\boldsymbol{x}_\mathcal{S}, \boldsymbol{n}_1, \boldsymbol{n}\}$, corrected significance threshold δ_{tar}

Output: A boolean indicating whether the feature subset $\mathcal{S}$ is pruned (`True`) or not (`False`)

```
for $j = 1, \dots, k$ do
    if $x_{\mathcal{S},j} \leq \min(n_{1,j}, n_j - n_{1,j})$ then
        Return False
    end if
end for
for $j = 1, \dots, k$ do
    $\beta^l_{\mathcal{S},j} = \frac{n_j - n_{1,j}}{n_j}(1 - \frac{x_{\mathcal{S},j}}{n_j})$
    $\beta^r_{\mathcal{S},j} = \frac{n_{1,j}}{n_j}(1 - \frac{x_{\mathcal{S},j}}{n_j})$
end for
$\pi_l(1), \dots, \pi_l(k) = \arg\mathrm{sort}([\beta^l_{\mathcal{S},1}, \dots, \beta^l_{\mathcal{S},k}])$
$\pi_r(1), \dots, \pi_r(k) = \arg\mathrm{sort}([\beta^r_{\mathcal{S},1}, \dots, \beta^r_{\mathcal{S},k}])$
$\widetilde{\Psi}_{cmh}(\mathcal{S}) \leftarrow \Psi_{cmh}(\mathcal{S})$
$\boldsymbol{x}^l_{tmp} \leftarrow [0, \dots, 0]$                ▷ Initialisation as a $k$-dimensional null vector
$\boldsymbol{x}^r_{tmp} \leftarrow [0, \dots, 0]$                ▷ Initialisation as a $k$-dimensional null vector
for $j = 1, \dots, k$ do
    $\boldsymbol{x}^l_{tmp} \leftarrow$ update($\boldsymbol{x}^l_{tmp}$)    ▷ Only one element is updated, therefore this step scales in $O(1)$. $\boldsymbol{x}^l_{tmp}$ is equal to $[x_{\mathcal{S},\pi_l(1)}, x_{\mathcal{S},\pi_l(2)}, \dots, x_{\mathcal{S},\pi_l(j)}, 0, \dots, 0]$
    if $\Psi_{cmh}(\boldsymbol{x}^l_{tmp}) \leq \widetilde{\Psi}_{cmh}(\mathcal{S})$ then
        $\widetilde{\Psi}_{cmh}(\mathcal{S}) = \Psi_{cmh}(\boldsymbol{x}^l_{tmp})$
    end if
end for
for $j = 1, \dots, k$ do
    $\boldsymbol{x}^r_{tmp} \leftarrow$ update($\boldsymbol{x}^r_{tmp}$)    ▷ Only one element is updated, therefore this step scales in $O(1)$. $\boldsymbol{x}^r_{tmp}$ is equal to $[x_{\mathcal{S},\pi_r(1)}, x_{\mathcal{S},\pi_r(2)}, \dots, x_{\mathcal{S},\pi_r(j)}, 0, \dots, 0]$
    if $\Psi_{cmh}(\boldsymbol{x}^r_{tmp}) \leq \widetilde{\Psi}_{cmh}(\mathcal{S})$ then
        $\widetilde{\Psi}_{cmh}(\mathcal{S}) = \Psi_{cmh}(\boldsymbol{x}^r_{tmp})$
    end if
end for
if $\widetilde{\Psi}_{cmh}(\mathcal{S}) > \delta_{tar}$ then
    Return False
end if
Return True
```

Algorithm 5 first verifies if the feature subset is potentially prunable, i.e. if it belongs to $\mathcal{P}_P = \left\{ \mathcal{S} \mid x_{\mathcal{S},j} \leq \min(n_{1,j}, n_j - n_{1,j}), \forall j = 1, \ldots, k \right\}$, Lines 1 to 5. Then Lines 6 to 9, the vectors $\boldsymbol{\beta}^l_{\mathcal{S}}$ and $\boldsymbol{\beta}^r_{\mathcal{S}}$, introduced in Theorem 2, are computed. Both vectors are sorted in increasing order, Lines 10 and 11, and the respective arguments are retrieved. This step is the most computationally expensive and scales in $O(k \log k)$. Lines 12 to 14 initialise the lower envelope of the minimum attainable p-value and the support vectors among which the smallest lower envelope value can be found. From Line 15 to Line 20 and from Line 21 to 25, these support vectors are updated and the corresponding attainable p-value computed, keeping the lowest one. The final condition is evaluated Line 27, by comparing the lower envelope value $\widetilde{\Psi}_{cmh}(\mathcal{S})$ to the corrected, temporary, significance threshold δ_{tar}. If the lower envelope is smaller than the corrected significance threshold, the feature subset is kept in and will be enumerated, if not the feature subset is discarded as it cannot become significant.

This last algorithm concludes the description of the algorithm FACS, which has been built in order to find **all** significantly statistically feature subsets whose representation $\mathbf{z}_{\mathcal{S}}$ would be significantly associated with the class label. In the next sections, we will focus on the experimental work in order to evaluate the performance of the algorithm.

2.4.5 *Simulation experiments*

In this section, the proposed approach, FACS, is evaluated in terms of computational runtime, statistical power and ability to correct for a confounding categorical covariate, on synthetic datasets. As the main characteristic of FACS compared to existing significant pattern mining methods is being able to account for a categorical covariate, this set of experiments has been designed to answer two major questions:

1. Is FACS actually able to account for a categorical covariate and to reduce the number of false positives due to confounding?

2. How does the ability to correct for confounding variables affect other aspects of FACS, such as statistical power or computational runtime?

2.4.5.1 *Comparison partners*

For all significant pattern mining algorithms presented in this section, we used a simplified version of the `Eclat` algorithm [113, 114], based on the implementation presented in [115], for the underlying closed pattern mining algorithm. FACS and all baselines were written in C++ and compiled with `gcc 4.8.2` with `-O3` optimisation. Each experiment was executed on a 2.5 Ghz Intel Zeon CPU with 64 GB of memory available, using a single thread.

We compared FACS with four significant discriminative pattern mining methods:

1. LAMP-χ^2 [54, 55] was the state-of-the-art in discriminative pattern mining. It uses Tarone's testability criterion but is based on Pearson's χ^2 test and thus cannot account for covariates. It is the comparison partner the closest to FACS from an algorithmic and statistical perspectives, the only difference lies in the test statistic used. Therefore comparing FACS to LAMP-χ^2 provides relevant answers to both questions above.

2. Bonf-CMH uses the CMH test, therefore is able to correct for confounders, but together with Bonferroni's correction, resulting in a considerable loss of statistical power. Comparing FACS to Bonf-CMH sheds light on the computational and statistical advantages gained by using Tarone's statistical framework together with our novel pruning criterion, when using the CMH test to account for a categorical covariate.

3. 2^k-FACS is a suboptimal version of FACS that implements the pruning criterion using the approach shown in Lemma 2, which scales as $O(2^k)$. This comparison partner differs from FACS only in the way the pruning criterion is implemented, and shows the impact of Theorem 2 compared to Lemma 2 in terms of computational efficient.

4. m^k-FACS is a suboptimal version of FACS which brute-force searches for the pruning criterion for each feature subset, scaling as $O(m^k)$. The objective of this baseline is to show how the brute-force computation of the pruning criterion compares to the use of Theorem 2 and would be prohibitive for medium to large k, highlighting the importance of our algorithmic contributions.

2.4.5.2 *Statistical metrics*

We describe the performance of FACS in terms of: (a) **statistical power**, defined as the proportion of truly associated subsets that are deemed significant, and (b) **false discovery rate** (FDR), defined as the proportion of subsets deemed significant which are false discoveries.

Both metrics, statistical power and false discovery rate, had to be adjusted in order to integrate characteristics of pattern mining algorithm outputs. First, we redefined the counts of false positives and of true positives to account for the fact that combinations of subsets or supersets of $\mathcal{S}_{true}$ might be associated with the label conditioning on the categorical covariate. Similarly, combinations of subsets or supersets of $\mathcal{S}_{conf}$ might be associated with the label and create false positives. Therefore, we decided to use the following approach. We considered a significantly associated feature combination to be a true positive if strictly more than half of its features belonged to the true feature subset, i.e. if $\mathbf{z}_{\mathcal{S}}$ is significantly associated and $|\mathcal{S} \cap \mathcal{S}_{true}| > \frac{|S|}{2}$. Analogously, such feature combination was counted as a false positive if strictly less than half of its features belonged to the true feature subset, i.e. if $\mathbf{z}_{\mathcal{S}}$ is significantly associated and $|\mathcal{S} \cap \mathcal{S}_{true}| < \frac{|S|}{2}$. If $|\mathcal{S} \cap \mathcal{S}_{true}| = \frac{|S|}{2}$, we added 0.5 to both counts of false and true positives. Using these novel definitions, we computed the false discovery rate as usual, i.e. $\mathrm{FDR} = \mathbb{E}(\frac{\mathrm{FP}}{\max(\mathrm{TP}+\mathrm{FP},1)})$. Second, we used a conservative approach of the definition of statistical power by computing the probability that $\mathcal{S}_{true}$ was deemed significantly associated, ignoring subsets of $\mathcal{S}_{true}$. However, to compare against a more challenging univariate baseline, `Bonf-CMH` was evaluated with an anti-conservative approach, such that $\mathcal{S}_{true}$ was considered to have been retrieved as long as any of its features is deemed significant.

2.4.5.3 *Data generation*

For each experiment, we generate synthetic datasets $\boldsymbol{X}$ containing n observations and p binary features, a label-vector $\boldsymbol{y}$ and a categorical covariate $\boldsymbol{c}$ with k categories.

In experiments evaluating the **runtime** of `FACS` and of its comparison partners, as a function of the number of features p or of the number of categories of the covariate variable k, correlations between features, labels and covariates play a minor role. Therefore $\boldsymbol{X}$, $\boldsymbol{y}$ and $\boldsymbol{c}$ were generated according to a fully-factorised generating distribution. Each element $X_{i,j}$ (resp. y_i) takes the value 1 or 0 according to the realisations of a Bernoulli random variable with parameter p_X (resp. p_y). The elements of the categorical covariate $\boldsymbol{c}$ are generated i.i.d. according to a categorical distribution with parameters $\boldsymbol{p}_c$.

For experiments that evaluated **statistical power** and **false discovery rate**, we had to generate realistic statistical dependencies between the truly associated feature combinations and the label, and between the confounded feature combinations, the label and the categorical covariate. To this end, a true associated feature combination $\boldsymbol{z}_{true}$ is generated as a binary vector correlated to the label-vector $\boldsymbol{y}$ but not correlated to the covariate-vector $\boldsymbol{c}$. A confounded feature combination $\boldsymbol{z}_{conf}$ is generated as a binary vector almost fully correlated to the covariate-vector $\boldsymbol{c}$. The strength of the correlations between the truly associated feature combination (resp. the covariate) and the label-vector is controlled by ρ'. As ρ' strictly controls the association between the class label and the covariate $\boldsymbol{c}$, it controls as well the association between the confounded feature subset and the class label, because their correlation almost reaches one (see the following paragraph).

The procedure used to generate a **confounded significant feature combination** is the following. Consider the correlation matrix

$$\Sigma = \begin{pmatrix} 1 & 0 & \rho' \\ 0 & 1 & \rho' \\ \rho' & \rho' & 1 \end{pmatrix}$$

We sample $\boldsymbol{o}$ from the multivariate Bernoulli distribution with mean $(p_{true}, p_c, p_y) = (0.5, 0.5, 0.5)$ and correlation matrix Σ, as many times as the number of samples n for each dataset [116] (*bindata* package in R). This results in n i.i.d three-vectors $\boldsymbol{o} = (z, c, y)$. For iteration $i \in [\![1, n]\!]$, the first component o_1 is an indicator function, which indicates if that feature interaction z_i contains the studied feature combination for sample i ($z_i = 1$), or not ($z_i = 0$). The second component o_2 is the categorical covariate c_i that we assume binary taking values in $\{0{,}1\}$, and the third component o_3 is the class label y_i. In this way, we generate the covariate vector $\boldsymbol{c}$, the label $\boldsymbol{y}$ and the true associated significant feature combination $\mathbf{z}_{true}$. For the confounded feature subsets, each feature combination $z_{conf,i}$ is obtained from the values of the binary covariate c_i by flipping its value with a low probability $p_\epsilon = 0.05$. We sample $\epsilon \sim \mathrm{B}(1, p_\epsilon)$ and then

$$z_{conf,i} = c_i \oplus \epsilon,$$

where $\oplus$ is the XOR operator. Thus, $\mathbf{z}_{conf}$ and $\boldsymbol{c}$ are strongly correlated. By looking at Σ, we see that the parameter ρ' controls the degree of association between $\boldsymbol{c}$ and $\boldsymbol{y}$. For high values of ρ', the vectors $\boldsymbol{c}$ and $\mathbf{z}_{conf}$ are highly correlated with $\boldsymbol{y}$.

Those two feature combinations ($\mathbf{z}_{true}$ and $\mathbf{z}_{conf}$) are further decomposed into two subsets of five feature vectors ($\mathcal{S}_{true}$ and $\mathcal{S}_{conf}$) such that the AND operation of those individual feature vectors gives the respective combination features. For samples for which the feature combination $\mathbf{z}_{\mathcal{S}}$ takes value 1, the corresponding elements of all five feature vectors that compose $\mathcal{S}$ have to take value 1 as well. For samples for which the feature combination $\mathbf{z}_{\mathcal{S}}$ takes value 0, the corresponding element of at least one feature among the five feature vectors that compose $\mathcal{S}$ has to take value 0. We choose this feature uniformly at random and keep the other ones with a value 1 to minimise the univariate association between individual features in $\mathcal{S}$ and the class label, making the discovery of $\mathcal{S}$ more challenging.

In addition, the $n \times p$ elements of the dataset $\boldsymbol{X}$ are generated by taking the value 1 or 0 according to the realisation of a Bernoulli random variable with parameter p_X. Different realisations are i.i.d. across observations and features. Then, each of the observations $\boldsymbol{X}_{i,.}$ was assigned a binary class label y_i and a categorical covariate value c_i that belongs to $\{0{,}1\}$. The ten feature vectors that compose $\mathbf{z}_{true}$ and $\mathbf{z}_{conf}$ replace ten distinct features chosen at random in the generated dataset, $\mathcal{S}_{conf}$ and $\mathcal{S}_{true}$.

Throughout the simulation experiments, both truly associated and confounded subsets $\mathcal{S}_{true}$ and $\mathcal{S}_{conf}$ have five interacting features each and are non-overlapping. Moreover the correlation $\rho = 2\rho'$ varies in [0,1] in the experiments and represents the strength of the signal in the data. It corresponds to the proportion of variance of the phenotype $\boldsymbol{y}$ explained by the truly associated and confounded subsets. Let us note that all methods were run in the same system and programmed in the same language making the runtime of the different approaches comparable.

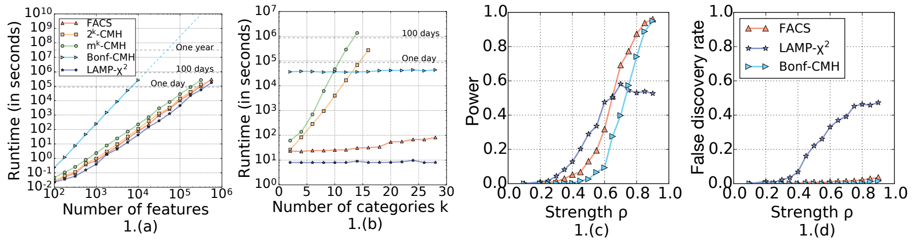


FIGURE 2.8: **Results of the simulation experiments.** (a) Runtime as a function of the number of features, p. (b) Runtime as a function of the number of categories of the covariate, k. (c) Statistical power as a function of the signal strength, ρ. (d) False discovery rate as a function of the strength of the signal ρ.

2.4.5.4 *Evaluation of runtime complexity*

We evaluated the runtime of our method while varying two fundamental parameters: the number of features p and the number of categories for the covariate k. For the runtime experiments we considered 100 datasets generated as stated above and containing $n = 500$ samples. When p was fixed, its value was $p = 5000$ and when k was fixed its value was $k = 2$ categories for the covariate. In both simulation experiments, both class labels and all categories for the covariate were equiprobable *a priori*, that is, $p_y = 0.5$ and $\boldsymbol{p}_c = \frac{1}{k}\mathbf{1}_k$. Finally, the probability p_X of any feature being positive was 0.1.

Figure 2.8(a) shows the runtime as a function of the number of features, p. This is a fundamental parameter, as datasets in the applications we target such as computational biology, are often characterized by a small sample size n and a large number of input features p. The main observation one can derive from Figure 2.8(a) is that FACS scales as the state of the art, LAMP-χ^2, when increasing the number of features p, while the Bonferroni-based method Bonf-CMH scales considerably worse. This indicates both that FACS is able to incorporate covariates with virtually no runtime overhead with respect to LAMP-χ^2 and confirms the efficacy of Tarone's testability criterion compared to Bonferroni-based method Bonf-CMH. The difference in performance gets particularly relevant for sufficiently large p where Bonf-CMH would not be feasible. The differences between significant pattern mining methods are coming from constant overheads: (i) the difference between FACS and m^k-FACS is in the order of $O(m^k)$ which is a constant in terms of p, so $O(1)$, (ii) FACS and 2^k-FACS differ by $O(2^k)$ which is also a constant in terms of p, therefore can be written $O(1)$ and (iii) the difference between LAMP-χ^2 and FACS comes from accounting covariates and scales in $O(k \log k)$, which corresponds to $O(1)$ when studying the influence of p.

As previously we fixed $k = 2$, we also wanted to study the impact of increasing k. To this end, Figure 2.8(b) shows the runtime as a function of the number of categories for the covariate, k. The runtime of FACS can be seen to scale slowly with k, as expected from the result in Theorem 2. The overhead with respect to unconditional pattern mining, represented by LAMP-χ^2, is small even for as many as $k = 26$ different categories for the covariate. In contrast, the runtime of m^k-FACS, which uses a naive-implementation of the pruning criterion, and of 2^k-FACS, which uses a suboptimal implementation based on Lemma 2, increases exponentially with k, which matches the theoretical analyses in the previous section. Additionally, we note that Theorem 2 is key as FACS is the only algorithm that scales better than Bonf-CMH for any value of k, especially from $k = 14$ to $k = 28$. In summary, this experiment demonstrates that FACS can scale to large values of k with only a minor computational overhead over LAMP and shows the importance of our efficient implementation of the pruning criterion to accomplish that result.

2.4.5.5 *Evaluation of statistical power and false discovery rate*

As announced Section 2.4.5.2, we describe the performance of FACS and its the comparison partners in terms of: (a) **statistical power**, defined as the proportion of truly associated subsets that are deemed significant, and (b) **false discovery rate** (FDR), defined as the proportion of subsets deemed significant which are false discoveries.

To compare the different approaches, we generated 300 synthetic datasets as described in Section 2.4.5.3 for different values of ρ. Choosing the same strength of association ρ between the label and the confounded feature combination, respectively the truly associated one, ensures that we do not favour the detection of true feature subsets over confounded feature subsets or vice versa. All synthetic datasets were generated using $n = 200$, $p = 5000$ and $k = 2$. The generation of the true and confounded feature subsets follows previous section 2.4.5.3. Both class labels and both categories of the covariate were equiprobable *a priori* with $p_y = 0.5$ and $p_c = 0.5$. As explained above, the probability of having the true feature combination is set to $p_{true} = 0.5$ for each sample. Finally, the probability p_X of being positive for any element of features that do not belong to $\mathcal{S}_{true}$ nor $\mathcal{S}_{conf}$ is 0.1.

Figures 2.8(c) compares FACS, LAMP-χ^2 and Bonf-CMH. Figure 2.8(c) shows that FACS has a similar statistical power as LAMP-χ^2, being slightly worse for weak signals and slightly better for stronger signals. Again, the performance of the Bonferroni-based method Bonf-CMH is drastically worse. More importantly, in Figure 2.8(d) we observe that unconditional significant discriminative pattern mining methods such as LAMP-χ^2 have an unacceptably high proportion of confounded features being deemed significant. In contrast, FACS greatly reduces the false discovery rate by conditioning on an appropriate covariate. Finally, the false discovery rate of BONF-CMH is even lower than that of FACS, a consequence of the low statistical power of methods based on Bonferroni's correction.

2.4.6 *Proof-of-concept application to GWAS datasets*

We perform a proof-of-concept experiment on GWAS datasets in order to assess the ability of FACS to correct for confounders while keeping a high statistical power.

In this proof-of-concept experiment, we look for significantly associated subsets of genetic variants in two *Arabidopsis thaliana* genome-wide association studies datasets [117], which we obtain from the easyGWAS online resource [118].

2.4.6.1 *Description of the datasets and preprocessing*

We chose two datasets from the plant model organism *A. thaliana* among datasets available in [117] that exhibit the highest amount of confounding, as measured by the genomic inflation factor λ described in [108] and that have binary labels. The datasets contain 84 and 95 samples, respectively. The labels of each dataset indicate the presence/absence of a plant defence-related phenotype: LY (yellowing leaves) and *avrB* (hypersensitive-response traits). In the two datasets, each plant sample is represented by a sequence of approximately 214,000 genetic bases. In both datasets, the SNPs were encoded as a binary vector as each plant-sample is inbred.

We consider the datasets of each plant trait separately, LY or *avrB*, and downsample the two datasets into smaller datasets: (1) according to which chromosome the genomic bases belong to because interactions between chromosomes are unlikely and (2) by downsampling evenly every 20 bases, and using different starting positions each time (i.e. 20 different offsets), to minimise the effect of the evolutionary correlations between nearby bases ($<$ 10 kilo-bases). This enables us to get rid of redundancy between bases while looking for mid-to-long range interactions. We note that each genomic base is included in one and only one dataset, and each chromosome is split in 20 subdatasets. It resulted in $5 \times 20 = 100$ complementary datasets containing between 1,423 and 2,661 features each. Our results for all methods are aggregated across all downsampled versions, per plant trait.

In both datasets we apply FACS and compare its results to two baselines already introduced above, LAMP-χ^2 and Bonf-CMH. The first one allows to find interactions of biomarkers but does not allow to correct for confounding effects, therefore potentially leading to a large number of spurious associations. The second one is able to correct for covariates, however the Bonferroni correction is more conservative than Tarone's statistical framework, which contributes to a loss of statistical power.

2.4.6.2 *Definition of the covariates*

As the datasets exhibit high genomic inflation factors, as indicated in Table 2.1, one needs to correct for the confounding effect of population structure to avoid many spurious associations. To this end, we condition on a categorical covariate that is representative of the **population structure** for the two datasets LY and *avrB*. We obtain this categorical covariate by running **k-means** on the first five principal components of the **kinship matrix** of the dataset, which represents the genetic relatedness of the plants [28, 119]. We then select the number of clusters k in a range from 2 to 8 that results in the best genomic inflation factor (the closest to 1). As a consequence, we consider $k = 3$ subpopulation clusters for *avrB* and $k = 5$ for LY.

2.4.6.3 *Results*

Table 2.1 shows the number of interactions of genetic variants reported as significant by each method, as well as the corresponding genomic inflation factor λ [108]. When compared to LAMP-χ^2, we observe a severe reduction in the number of interactions deemed significant by FACS, as well as a sharp decrease in λ. This seems to indicate that many SNPs interactions reported by LAMP-χ^2 are affected by confounding. The high λ values of LAMP-χ^2 show strong marginal associations between many feature interactions and labels, inflating the corresponding Pearson χ^2-test statistic values compared to the expected χ^2 null distribution and resulting in many spurious associations. However, since most of those feature interactions are independent of the labels given the covariates, the CMH test statistic values are much closer to the χ^2 distribution, leading to a lower λ and resulting in hits that are corrected for the covariate. Moreover, the lack of power of Bonf-CMH results in a very small number of hits, which is also the expected behaviour. We can expect that, as Bonf-CMH is also able to correct for confounding effects, the reduction in the number of hits this time corresponds to false negatives.

Datasets	FACS		LAMP-χ^2		Bonf-CMH
	hits	λ	hits	λ	hits
LY	433	1.17	100,883	3.18	19
avrB	43	1.21	546	2.38	1

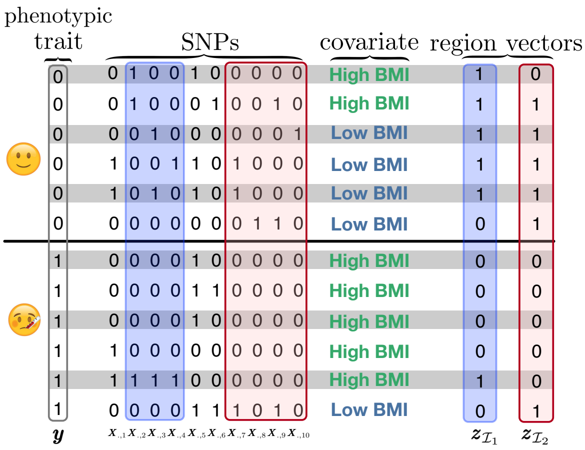
Table 2.1: **Significantly associated interactions in GWAS data.** Total number of significant interactions of SNPs found by LAMP-χ^2, FACS and Bonf-CMH and average genomic inflation factor λ. λ for Bonf-CMH is similar to FACS since both use the CMH test.

2.5 genome-wide genetic heterogeneity discovery with categorical covariates

As exposed at the beginning of this thesis, **genetic heterogeneity** is a phenomenon that cannot be easily modelled or exploited by means of classical univariate testing. This chapter introduces a novel method `FastCMH` that is able to account for confounding factors, as `FACS` does, together with exploiting genetic heterogeneity, similarly to `FAIS`. The potential of the method is illustrated in a thorough application to GWAS datasets, which goes beyond the proof-of-concept experiment presented in Section 2.4.5 and the experiments performed in [48] that could not account for confounding factors. In this chapter, we first present Section 2.5.1 a model of genetic heterogeneity and the main objectives of the algorithm `FastCMH`, which we developed to find significantly associated regions of genetic heterogeneity. Second, the `FastCMH` algorithm is described Section 2.5.2. Finally the results we obtained on simulated and real GWAS datasets are presented Sections 2.5.3 and 2.5.4, respectively.

2.5.1 *`FastCMH`: main objective*

Similarly to the notations introduced Section 2.3.1.3, we consider a dataset consisting of n samples subdivided into n_1 cases and $n - n_1$ controls according to a binary phenotype $\boldsymbol{y}$. For each individual $i \in [\![1,n]\!]$, we assume a genotypic representation in the form of an **ordered** sequence of p binary genetic variants, $\mathbf{X}_{i,.} = \left(X_{i,1}, X_{i,2} \ldots, X_{i,p}\right)$ with $X_{i,t} \in \{0,1\}$. For example, these binary variants could be the result of a dominant or recessive encoding of SNPs. Furthermore, for each individual $i \in [\![1,n]\!]$, we record a categorical covariate $\boldsymbol{c}$ with k states, i.e. $c_i \in [\![1,k]\!]$.



phenotypic trait	SNPs										covariate	region vectors	
$\boldsymbol{y}$	$\boldsymbol{X}_{.,1}$	$\boldsymbol{X}_{.,2}$	$\boldsymbol{X}_{.,3}$	$\boldsymbol{X}_{.,4}$	$\boldsymbol{X}_{.,5}$	$\boldsymbol{X}_{.,6}$	$\boldsymbol{X}_{.,7}$	$\boldsymbol{X}_{.,8}$	$\boldsymbol{X}_{.,9}$	$\boldsymbol{X}_{.,10}$		$\boldsymbol{z}_{\mathcal{I}_1}$	$\boldsymbol{z}_{\mathcal{I}_2}$
0	0	1	0	0	1	0	0	0	0	0	High BMI	1	0
0	0	1	0	0	0	1	0	0	1	0	High BMI	1	1
0	0	0	1	0	0	0	0	0	0	1	Low BMI	1	1
0	1	0	0	1	1	0	1	0	0	0	Low BMI	1	1
0	1	0	1	0	1	0	1	0	0	0	Low BMI	1	1
0	0	0	0	0	0	0	0	1	1	0	Low BMI	0	1
1	0	0	0	0	1	0	0	0	0	0	High BMI	0	0
1	0	0	0	0	1	1	0	0	0	0	High BMI	0	0
1	0	0	0	0	1	0	0	0	0	0	High BMI	0	0
1	1	0	0	0	0	0	0	0	0	0	High BMI	0	0
1	1	1	1	1	0	0	0	0	0	0	High BMI	1	0
1	0	0	0	0	1	1	1	0	1	0	Low BMI	0	1

$$\boldsymbol{z}_{\mathcal{I}_1} = \boldsymbol{X}_{.,2} \vee \boldsymbol{X}_{.,3} \vee \boldsymbol{X}_{.,4}$$

$$\boldsymbol{z}_{\mathcal{I}_2} = \boldsymbol{X}_{.,7} \vee \boldsymbol{X}_{.,8} \vee \boldsymbol{X}_{.,9} \vee \boldsymbol{X}_{.,10}$$

FIGURE 2.9: **Example of aggregation of weak signals.** Individual SNPs in each genomic region $\mathcal{I}_1$ and $\mathcal{I}_2$ do not show an association with the binary label $\boldsymbol{y}$. By contrast, their combinations into a region vector do. While both $\boldsymbol{z}_{\mathcal{I}_1}$ (blue) and $\boldsymbol{z}_{\mathcal{I}_2}$ (red) show an association with $\boldsymbol{y}$, we can notice that $\boldsymbol{z}_{\mathcal{I}_2}$ is very correlated to the binary covariate $\boldsymbol{c}$ indicating the BMI levels of the samples. Therefore, $\mathcal{I}_1$ is likely to be a truly associated region and $\mathcal{I}_2$ a spurious one. In this example, $n = 10, n_1 = n - n_1 = 5$, $p = 10$ and $k = 2$.

Under a model of genetic heterogeneity, several genetic variants in close proximity might have evolved to affect the phenotype in a similar manner. However, their individual effect sizes might be too weak to reach significance in a single-marker GWAS. Assuming that most individual variants in a genomic region $j \in \mathcal{I}$, where $\mathcal{I} = [\![s, e]\!]$, have the same direction of effect motivates aggregating them into a new feature $z_{\mathcal{I},i} = \max\left(X_{i,s}, X_{i,s+1}, \ldots, X_{i,e}\right)$ for the entire region. This is equivalent to defining $z_{\mathcal{I},i} = 1$ if the genomic region $[\![s,e]\!]$ for individual i contains any genetic variant encoded as 1 (typically minor alleles or risk alleles under the model of choice), and $z_{\mathcal{I},i} = 0$ if it only contains genetic variants encoded as 0. More generally, the region can be encoded with a logical OR combination $z_{\mathcal{I},i} = X_{i,s} \vee X_{i,s+1} \vee \ldots \vee X_{i,e}$. For genomic regions for which these assumptions apply, the region vector $\boldsymbol{z}_{\mathcal{I}}$ will exhibit a stronger signal than any of the individual variants, allowing the discovery of novel genome-wide significant multivariate associations. This situation is illustrated in Figure 2.9, where the variants contained in regions $\mathcal{I}_1$ (in blue) and $\mathcal{I}_2$ (in red) are all weakly associated with the phenotype $\boldsymbol{y}$. In contrast, their respective vectors $\boldsymbol{z}_{\mathcal{I}_1}$ and $\boldsymbol{z}_{\mathcal{I}_2}$ exhibit a much stronger association.

Nevertheless, as stated above, significant associations in a GWAS often originate merely as the result of confounding by external covariates such as gender, age, population structure or environmental factors. It is therefore essential to account for these covariates in any method that tries to assess the association between genotype and phenotype. This is also represented in Figure 2.9. The association with the phenotype $\boldsymbol{y}$ of the region vector $\boldsymbol{z}_{\mathcal{I}_2}$ is a spurious association exclusively mediated by the covariate $\boldsymbol{c}$ (BMI level of the sample), while the region vector $\boldsymbol{z}_{\mathcal{I}_1}$ remains associated after correcting for the effect of the covariate.

The main objective of the FastCMH algorithm can be summarised as:
Objective: Given a dataset $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{c})$, the goal of FastCMH is to:

1. Compute Tarone's corrected significance threshold δ_{tar}.

2. Retrieve all contiguous genomic regions $\mathcal{I} \subset \{[\![s,e]\!] | 1 \leq s \leq e \leq p\}$ such that the p-value $p_{\mathcal{I}}$ is below δ_{tar} when testing the null hypothesis H_0: $Z_{\mathcal{I}} \perp\!\!\!\perp Y | C$ of conditional independence given the covariates –rather than normal independence.

To achieve its goal, FastCMH combines the scheme proposed in FAIS [48] to explore the search space consisting of all possible genomic regions with the novel approach FACS presented Section 2.4 of this thesis, to correct for categorical covariates in significant pattern mining. A full description of FastCMH is provided in the Section 2.5.2. The test statistic of choice is the CMH test, similarly to the algorithm FACS, which allows to account for a confounding categorical covariate. To the extent of our knowledge, FastCMH is the first method that exploits genetic heterogeneity to retrieve associated variants in GWAS while accounting for a categorical covariate.

2.5.2 *FastCMH: description and pseudocode*

In this section, we will first give a very high-level description of the algorithm of FastCMH (Section 2.5.2.1), then describe in more detail two key elements of the algorithm: the get_testable_regions algorithm that allows to efficiently implement Tarone's procedure in order to calculate the corrected significance threshold (Section 2.5.2.2), and the filtering procedure's filter_overlapping_regions that groups together overlapping significant genomic regions in order to correct for potential redundancy in the results (Section 2.5.2.3).

2.5.2.1 *FastCMH: high-level pseudo-code*

The high-level pseudocode of FastCMH is shown in Algorithm 6. Conceptually, our method follows a similar sketch as FACS and involves five main steps.

Algorithm 6 `FastCMH`

Input: Dataset $\mathcal{D} = \big\{(\mathbf{X}_{i,\cdot}, y_i, c_i)\big\}_{i=1}^{n}$, desired FWER α
Output: Set of non-overlapping conditionally associated genomic regions
$\mathcal{P}_{sig,filt} = \{\mathcal{I} \mid \mathcal{I} = [\![s,e]\!], 1 \leq s \leq e \leq p, p_{\mathcal{I}} \leq \delta_{tar}\}$

```
1: Initialise global variables $\delta_{tar} = 1$ and $\mathcal{P}_{tar}(\delta_{tar}) = \emptyset$
2: get_testable_regions($\mathcal{D}, \alpha$)
3: $\mathcal{P}_{sig,raw} \leftarrow \{\mathcal{I} \in \mathcal{P}_{tar}(\delta_{tar}) \mid p_{\mathcal{I}} \leq \delta_{tar}\}$
4: $\mathcal{P}_{sig,filt} \leftarrow$ filter_overlapping_regions($\mathcal{P}_{sig,raw}$)
5: Return $\mathcal{P}_{sig,filt}$
```

First, in Line 1, the global variables δ_{tar} and $\mathcal{P}_{tar}(\delta_{tar})$ are initialised. Second, in Line 2, we invoke the routine `get_testable_regions` to compute Tarone's adjusted significance threshold δ_{tar} and retrieve the corresponding set of testable genomic regions $\mathcal{P}_{tar}(\delta_{tar})$ under the CMH test. The enormous number of candidate genomic regions, often in the order of hundreds of billions, or even trillions, makes the routine `get_testable_regions`, described in detail in Algorithm 7 below, the most challenging and crucial part of `FastCMH`.

Third, in Line 3, p-values $p_{\mathcal{I}}$ obtained from CMH tests are evaluated for all testable genomic regions $\mathcal{I} \in \mathcal{P}_{tar}(\delta_{tar})$. Since a large proportion of all candidate genomic regions are not testable, and thus can never be significant, Tarone's trick allows us to greatly reduce the computational burden of this step without incurring any additional false negatives. Those testable regions $\mathcal{I} \in \mathcal{P}_{tar}(\delta_{tar})$ whose p-values $p_{\mathcal{I}}$ are below Tarone's adjusted significance threshold δ_{tar} are deemed significant and stored in $\mathcal{P}_{sig,raw}$.

Fourth, while all genomic regions in $\mathcal{P}_{sig,raw}$ are significantly associated with the phenotype –given the effect of the covariate– both, the exhaustive nature of the search and linkage disequilibrium, tend to generate disjoint clusters of significant genomic regions that have a high overlap with each other. To eliminate this redundancy which might otherwise complicate the analysis of the results, we invoke the routine `filter_overlapping_regions` in Line 4. This procedure groups all significant genomic regions in $\mathcal{P}_{sig,raw}$ into disjoint clusters of overlapping regions, generating a new set $\mathcal{P}_{sig,filt}$ containing only the most significant genomic region for each cluster and the cluster boundaries, discarding the other significant regions. Finally, the set $\mathcal{P}_{sig,filt}$ is returned as `FastCMH`'s output, Line 5.

2.5.2.2 *`FastCMH`: getting testable regions*

As mentioned before, efficiently finding Tarone's adjusted significance threshold δ_{tar} and the set of testable genomic regions $\mathcal{P}_{tar}(\delta_{tar})$ is the key algorithmic step in `FastCMH`. A naive enumeration approach, which would require computing the minimum attainable p-value $\Psi_{\mathcal{I}}$ for all $\frac{p(p-1)}{2} = O(p^2)$ candidate regions, would not scale to the number of genetic variants p in typical GWAS datasets. For this reason, the routine `get_testable_regions` of `FastCMH` combines the branch-and-bound approach used by its predecessor `FAIS` with the a modification of novel search space pruning criterion developed for the CMH test in Section 2.4.4.

Algorithm 7 `get_testable_regions`

Input: Dataset $\mathcal{D} = \left\{(\mathbf{X}_{i,\cdot}, y_i, c_i)\right\}_{i=1}^{n}$, desired FWER α
Output: Tarone's adjusted significance threshold δ_{tar} and set of testable genomic regions $\mathcal{P}_{tar}(\delta_{tar})$

```
1:  $\mathcal{R}_{queue} \leftarrow \{\mathcal{I} \,|\, \mathcal{I} = [\![e,e]\!], 1 \leq e \leq p\}$
2:  while $\mathcal{R}_{queue} \neq \emptyset$ do    ▷ Regions in $\mathcal{R}_{queue}$ enumerated firstly in increasing order of length and then starting position
3:      $\mathcal{I} \leftarrow$ dequeue($\mathcal{R}_{queue}$)    ▷ $\mathcal{I} = [\![s,e]\!]$
4:      if is_testable_cmh($\mathcal{I}, \delta_{tar}$) then
5:          $\mathcal{P}_{tar}(\delta_{tar}) \leftarrow \mathcal{P}_{tar}(\delta_{tar}) \cup \mathcal{I}$
6:          while $\widehat{FWER}_{tar}(\delta_{tar}) > \alpha$ do
7:              Decrease $\delta_{tar}$
8:              $\mathcal{P}_{tar}(\delta_{tar}) \leftarrow \{\mathcal{I} \in \mathcal{P}_{tar}(\delta_{tar})$ : is_testable_cmh($\mathcal{I}, \delta_{tar}$)$\}$
9:              $\widehat{FWER}_{tar}(\delta_{tar}) \leftarrow \delta_{tar}|\mathcal{P}_{tar}(\delta_{tar})|$
10:         end while
11:     end if
12:     if non_prunability_condition($[\![s-1,e]\!]$) then    ▷ See below for a detailed description of non_prunability_condition
13:         $\mathcal{R}_{queue} \leftarrow \mathcal{R}_{queue} \cup [\![s-1,e]\!]$
14:     end if
15: end while
```

The routine `get_testable_regions` of Algorithm 7 first initialises the search space of genomic regions $\mathcal{R}_{queue}$ in Line 1 to contain candidate genomic regions of length 1.

After initialisation, in Line 2, the algorithm starts enumerating the genomic regions in $\mathcal{R}_{queue}$ in the same order as the `FAIS` algorithm in [48]: enumerating first in increasing order of region length, i.e. smaller regions first, and then, among all regions having the same length, in increasing order of starting position. For each genomic region $\mathcal{I}$ being processed, we perform the steps described below.

First, in Line 4, we compute the minimum attainable p-value for the CMH test, $\Psi_{\mathcal{I}}$, using the closed-form expression Equation 2.2. If implemented naively, this would have complexity $O(n(e-s+1))$. However, since candidate genomic regions are numerated in increasing order of length, it is possible to obtain the genomic region vector $\mathbf{z}_{[\![s,e]\!]}$ from either the genomic region vector $\mathbf{z}_{[\![s,e-1]\!]}$ or by $\mathbf{z}_{[\![s,e+1]\!]}$. Therefore, the complexity can be reduced to $O(n)$. Moreover, since $\mathbf{z}_{i,[\![s,e-1]\!]} = 1$ or $\mathbf{z}_{i,[\![s,e+1]\!]} = 1$ implies that $\mathbf{z}_{i,[\![s,e]\!]} = 1$, the complexity can be further reduced to $O(n - x_{prev})$, where x_{prev} is the number of individuals that have the genomic region vector equal to 1 in either region $[\![s,e-1]\!]$ or region $[\![s,e+1]\!]$. Implementing the computation of the genomic region vector in this manner, in Line 4 of Algorithm 7, greatly increases the efficiency of the algorithm with only a moderate increase in memory usage, equivalent to storing a second copy of the original dataset in memory. Then, we check if the region vector is testable at the current significance threshold δ_{tar}, i.e. if $\Psi_{\mathcal{I}} \leq \delta_{tar}$. If it is, the region is added to the set of testable regions $\mathcal{P}_{tar}(\delta_{tar})$ in Line 5 and Tarone's condition $\widehat{FWER}_{tar}(\delta_{tar}) = \delta_{tar}/|\mathcal{P}_{tar}(\delta_{tar})| > \alpha$ is checked in the following line. If the condition is found to be violated, it means that the current significance threshold δ_{tar} is too large and must be decreased (Line 7). In practice, we implement this step as $\delta \leftarrow 10^{-\Delta}\delta$, where Δ is an implementation-dependent hyperparameter. This is equivalent to performing grid-search on δ, with logarithmically-spaced candidate values with step-size Δ in the log scale. Provided that Δ is not too large, we found this hyperparameter to have a negligible effect on the result. We fixed $\Delta = 0.06$ throughout our experiments, corresponding to considering 500 values of δ in a logarithmic grid between $\delta = 1$ and $\delta = 10^{-30}$. By decreasing δ_{tar}, some already processed genomic regions which were found to be testable, i.e. $\Psi_{\mathcal{I}} \leq \delta_{tar}$ for a larger value of δ_{tar}, might now become untestable. Those genomic regions are retrieved and removed from $\mathcal{P}_{tar}(\delta_{tar})$ in Lines 8. Based on the scheme to decrease δ_{tar} described in the paragraph above, the set of genomic regions to be removed from $\mathcal{P}_{tar}(\delta_{tar})$ is composed of those genomic regions $\mathcal{I}$ currently in $\mathcal{P}_{tar}(\delta_{tar})$ that do not satisfy $\delta_{tar} < \Psi_{\mathcal{I}} \leq 10^{\Delta}\delta_{tar}$. This property shows that it is possible to implement Line 8 with O(1) complexity by using a data-structure for $\mathcal{P}_{tar}(\delta_{tar})$ that stores genomic regions in different bins according to their minimum attainable p-value. More precisely, we assign a genomic region $\mathcal{I}$ to the i-th bin if $10^{-i\Delta} < \Psi_{\mathcal{I}} \leq 10^{-(i-1)\Delta}$. With this data-structure, each execution of Line 8 corresponds to removing exactly one bin from $\mathcal{P}_{tar}(\delta_{tar})$, an operation that requires no search. $\widehat{FWER}_{tar}(\delta_{tar})$ is finally updated Line 9.

The last step in processing a candidate genomic region $\mathcal{I}$ is also the most relevant for computational efficiency, the **prunability** step in Line 12 of Algorithm 7. As the significant region mining algorithm follows a breadth-first enumeration strategy, enumerating regions by increasing order of length and starting position, the objective is to evaluate whether the region $[\![s-1, e]\!]$ is prunable. Choosing this particular region allows to save runtime complexity, as the prunability step mainly requires to compute minimum attainable p-values, leading to only using information that is either readily available, i.e. that concerns its parents in the enumerating tree, or that scales in $O(1)$. In this enumeration scheme, the currently processed region $\mathcal{I} = [\![s, e]\!]$ has in general two children supersets that are only one element longer, $[\![s-1, e]\!]$ and $[\![s, e+1]\!]$. The only exceptions occur when $s = 1$ or when $e = p$, in which case one superset that is only one element longer exists (or zero for $s = 1$ and $e = p$). If $s = 1$, $[\![s-1, e]\!]$ does not exist. If $s \neq 1$, we evaluate the prunability of $[\![s-1, e]\!]$, the direct child of $\mathcal{I}$ in the enumeration tree for which the second direct parent $[\![s-1, e-1]\!]$ has already been processed or pruned, as the regions are enumerated in increasing order of length and starting position. In this situation, if $\mathcal{I}$ is prunable, all its descendants can be pruned as well, which holds in particular for $[\![s-1, e]\!]$. If $\mathcal{I}$ is not prunable, then $[\![s-1, e]\!]$ is either prunable or not. Notably, if its second direct parent $[\![s-1, e-1]\!]$ was evaluated as prunable or has already been pruned, then $[\![s-1, e]\!]$ is prunable. As a conclusion the non-prunability condition non_prunability_condition can be summarised as:

$$\begin{aligned} \texttt{non\_prunability\_condition} = & (s \neq 1 \wedge [\![s-1, e-1]\!] \text{ has not been pruned} \\ & \wedge\ \text{not}(\texttt{is\_prunable\_cmh}([\![s-1, e-1]\!]) \\ & \vee \texttt{is\_prunable\_cmh}([\![s, e]\!]))) \end{aligned}$$

If this **non-prunability condition**, non_prunability_condition, evaluates to True, the region $[\![s-1, e]\!]$ is added to the queue of regions to be processed $\mathcal{R}_{queue}$. In the non-prunability condition expression, is_prunable_cmh is the **pruning condition** and evaluates to True if the minimum attainable p-value of the region being evaluated is smaller than the current corrected significant threshold and if the condition on the support is respected, and to False if not. The pseudocode of the algorithm used to evaluate the pruning condition as well as the mathematical details describing its derivation in the context of mining significantly associated genomic regions can be found in Section 2.4.4. However, the features are combined differently in FACS and FastCMH, using a logical AND or logical OR, respectively. Therefore, the pruning condition developed for FACS needs to be adapted to FastCMH. The required changes are two-fold. First, the condition Line 2 of Algorithm 5 needs to change from $x_{\mathcal{S},j} \leq \min(n_{1,j}, n_j - n_{1,j})$ to $x_{\mathcal{S},j} \geq \max(n_{1,j}, n_j - n_{1,j})$. The reason is that, as we progress down the tree, the number of active elements ($\boldsymbol{z}_{\mathcal{I}} = 1$) of each genomic region vector increases due to the OR operation, while the support is decreasing when the logical AND operation is used instead. Therefore, the non-testability relationship –between parents and children– that can be exploited is the one that implies that if the support of a feature is sufficiently large, rendering the feature subset vector non-testable, its children in the enumerating tree, with an even larger support, won't be testable either. Second, the change from the AND to the OR combination leads to a change in the definition of the vectors $\boldsymbol{\beta}^l_{\mathcal{I}}$ and $\boldsymbol{\beta}^r_{\mathcal{I}}$. In FastCMH, Lines 8 and 9 would change to $\beta^l_{\mathcal{S},j} = \frac{n_j - n_{1,j}}{n_j}(\frac{x_{\mathcal{S},j}}{n_j})$ and $\beta^l_{\mathcal{S},j} = \frac{n_j - n_{1,j}}{n_j}(\frac{x_{\mathcal{S},j}}{n_j})$, respectively.

The routine get_testable_regions naturally terminates when all candidate regions in $\mathcal{R}_{queue}$ have either been pruned or processed. At that point, the algorithm has converged and the final values of δ_{tar} and $\mathcal{P}_{tar}(\delta_{tar})$ are available.

2.5.2.3 *FastCMH: filtering procedure*

Algorithm 8 describes the filter_overlapping_regions method used in Step 4 in Algorithm 6 above. This filtering procedure was first used in [48].

```
Algorithm 8 filter_overlapping_regions
Input: Set of regions with the associated p-values, $\mathcal{P}_{sig,raw} = \{(\mathcal{I}, p_\mathcal{I}) \,|\, p_\mathcal{I} \leq \delta_{tar}\}$
Output: Set of significantly associated, non-overlapping regions which are most significant in each cluster and boundaries of each cluster $\mathcal{P}_{sig,filt}$

1: Determine disjoint clusters $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_\ell$, where each cluster is the union of a subset of $\mathcal{P}_{sig,raw}$
2: Assign each interval $\mathcal{I} \in \mathcal{P}_{sig,raw}$ the label $v$ if it belongs to cluster $\mathcal{C}_v$
3: For all clusters $\mathcal{C}_v \in \{\mathcal{C}_1, \ldots, \mathcal{C}_\ell\}$ , find the region $\mathcal{I}_v \in \mathcal{C}_v$ that has the smallest p-value $p_{\mathcal{I}_v}$ amongst all regions in $\mathcal{C}_v$
4: For all clusters $\mathcal{C}_v \in \{\mathcal{C}_1, \ldots, \mathcal{C}_\ell\}$, find the boundaries $[\![s_v, e_v]\!]$ of $\mathcal{C}_v$
5: Return $\mathcal{P}_{sig,filt} = \{(\mathcal{I}_v, [\![s_v, e_v]\!]) \,|\, v \in [\![1, \ell]\!]\}$
```

In simple terms, the regions in $\mathcal{P}_{sig,raw}$ are first grouped into **clusters**; if one considers the union of all regions in $\mathcal{P}_{sig,raw}$, then there would be several groups of overlapping regions which each form larger contiguous regions. We call each of these larger disjoint contiguous regions a cluster. Note that the clusters do not overlap, but the regions within each cluster do overlap. Figure 2.10 shows an example of Algorithm 8 applied to a cluster containing four regions. The clusters can be determined in Line 1 by following two rules: (i) every regions must belong to one cluster, and (ii) if two regions overlap, they belong to the same cluster. After the clusters $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_\ell$ have been determined, another pass is made through the regions and each region is given the label $v \in [\![1, \ell]\!]$ of the cluster to which it belongs (Line 2). Next, we chose two types of summarisation of the clusters. First, we find the region in each cluster which has the smallest p-value. In the case of ties (two or more regions with the same minimum p-value), the region that has the longest length is returned. If the lengths are the same, then the region with the smallest starting point is returned. This view point allows to access the core of the region and eliminate potential noisy variants at the end points (Line 3). Second, we retrieve the boundaries of the clusters, which are determined as the boundaries of the union of the regions in the cluster. This second approach allows to reduce the risk of missing potentially important variants (Line 4).

Finally, we construct $\mathcal{P}_{sig,filt}$ in Line 5 as the collection of tuples that contain for each cluster the region that is the most significant and the boundaries of the cluster. Figure 2.10 illustrates the procedure for a single cluster; it shows how four overlapping regions (red) form a single cluster (magenta), and how the filtering process identifies the region with the smallest p-value (green, p-value $= 10^{-9}$). Additionally the total number of associated regions is returned.

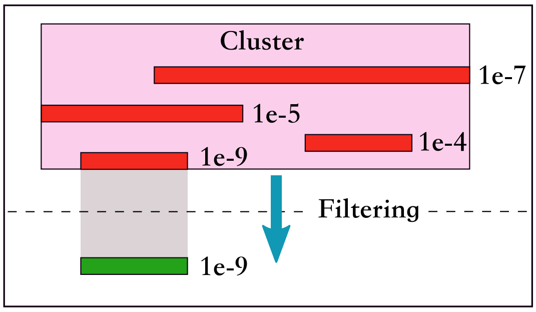


FIGURE 2.10: **An illustrative example of `filter_overlapping_regions`**. There are four overlapping significant regions (in red), which together form a single cluster (magenta). The result of the filtering in this cluster is the green region since it has the smallest p-value, p=1e-9, and the boundaries of the cluster are indicated by the magenta left and right end points.

2.5.3 *Simulation experiments*

The objective of the simulation experiments is to answer the following questions in a setting where the ground truth is known:

1. Is `FastCMH` able to correct for confounders without affecting the statistical power and how does it impact runtime?

2. Does searching among **all** intervals allow to discover novel associations that would not be found with state-of-the-art methods such as burden tests?

2.5.3.1 *Statistical metrics*

In the following sections, we will use the metrics introduced hereafter:

1. The **power**, calculated as the number of non-confounded, truly associated regions that are retrieved, divided by the number of regions deemed significant. This ratio is then average over 200 iterations. This metric shows the ability of the different algorithms to find truly associated regions.

2. The **confounded positive rate** (CPR), defined as the number of confounded regions that are deemed significantly associated, divided by the total number of confounded regions. This proportion is then average over 200 iterations. It is a variant of the false positive rate (FPR), which only accounts for confounded regions when evaluating the number of false positives and true negatives, instead of any non-associated region. This metric helps to understand the ability of `FastCMH` and of the different baselines to account for confounding when selecting regions.

3. The **FWER**, which stands for the family-wise error rate. In our experiments, FWER only accounts for the non-confounded regions in the calculation of the number of false positives. This metric shows how the algorithms behave with regard to false positives that do not emerge because of confounding, while, complementarily, CPR examines and summarises the number of false positives originating from confounding factors.

2.5.3.2 *Evaluation of the computational and statistical impacts of correcting for confounders*

Comparison partners: We compare `FastCMH`, our proposed method, with two alternative approaches: (1) `FAIS`-χ^2, a version of the method in [48] employing Pearson's χ^2 test, which uses Tarone's approach but cannot account for confounding and (2) `Bonf-CMH`, which does not use Tarone's statistical framework, but does use the CMH test.

Data generation: Datasets are generated, containing p features and n samples, where p and n can vary depending on the experiment. Each element is sampled i.i.d. from a Bernoulli distribution with probability p_X of being 1 equal to 0.3. When needed, associated genomic regions are inserted in the generated datasets. When this is the case, it consists of exactly one truly significantly associated genomic region vector and one confounded genomic region vector, that is, a region vector that is highly correlated with the (confounding) covariate $\boldsymbol{c}$, $\boldsymbol{c}$ being itself correlated with the phenotype $\boldsymbol{y}$. In our experiments, both regions contain 5 variants each. The parameter $\rho \in [0,1]$ controls the strength of the signal of the truly significant region and of the confounding covariate, and thus of the confounded region. The generation process is similar to the one presented Section 2.4.5.3, with the main differences that the five variants in both regions are restricted to be contiguous and that each region vector is the OR combination of the variants that compose it, rather than the AND combination. In turn, the latter change requires adapting the way the individual variants of interest are generated so that their OR combination matches the derived vectors $\mathbf{z}_{true}$ and $\mathbf{z}_{conf}$. In this case, when $z_{true,i} = 0$ (resp. $z_{conf,i} = 0$), all variants in $\mathcal{S}_{true}$ (resp. $\mathcal{S}_{conf}$) must take value 0 for the i-th sample, whereas when $z_{true,i} = 1$ (resp. $z_{conf,i} = 1$) a single variant in $\mathcal{S}_{true}$ (resp. $\mathcal{S}_{conf}$) is sampled uniformly at random and set to 1, and all others are set to 0, once again minimising the univariate association of the single variants of the associated regions with the phenotype.

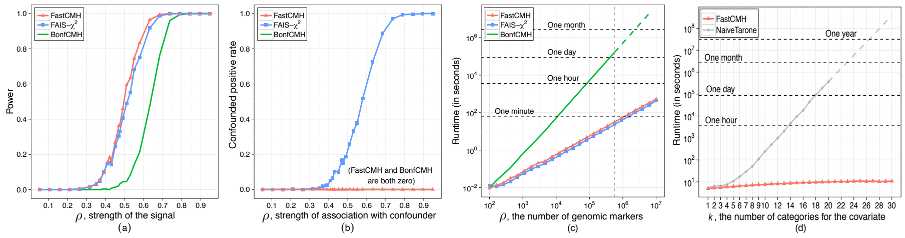


Figure 2.11: **Results of the simulation experiments.** (a) A comparison of the power of FastCMH, FAIS-χ^2 and Bonf-CMH for detecting true significant regions, as ρ varies. The parameters are chosen as: $n = 500$, $p = 10^6$, $k = 2$ and $\rho \in [0.05, 0.95]$. (b) The proportion of confounded significant regions falsely detected by each of those three algorithms (CPR). The parameters have the same values as for (a). (c) A comparison of the runtimes for the three methods as a function of the number of features p, where the dashed section for Bonf-CMH represents extrapolated values. Both axes are plotted on the log-scale. The set of parameters is as follows: $n = 500$, $k = 4$, $p \in [10^2, 10^7]$. (d) The difference in runtime between FastCMH and a naive implementation of a procedure combining Tarone's trick and the CMH test, as a function of the number of categories of the covariate. The dashed section of the naive method represents extrapolated values. We chose: $n = 500$, $p = 10^5$, $k \in [\![1,30]\!]$.

Power, confounded positive rate and FWER: There are two complementary situations where FastCMH has improved performance. First, it has improved detection performance of truly significant regions, due its use of Tarone's testability criterion, when compared to Bonf-CMH. In Figure 2.11(a), both FastCMH and FAIS-χ^2 have higher power than Bonf-CMH for $\rho \in [0.3, 0.8]$. Second, it will often (correctly) omit regions which appear to be significant, but are actually highly correlated with the covariate rather than the phenotype. Figure 2.11(b) shows that FastCMH and Bonf-CMH do not detect these confounded genomic regions as opposed to FAIS-χ^2. We consider the detection of these regions to be false positives. Furthermore, Figure 2.12 shows that both FastCMH and its comparison partners satisfy FWER control. As explained in Section 2.5.3, only false positive regions that were non-confounded were taken into account, to disentangle the effect of confounding with respect to the evaluation of the FWER control. In Figure 2.13, a variation of these experiments is performed in which we show that the power, confounded positive rate and the FWER of FastCMH are mostly unaffected by the number of categories, provided the resulting contingency tables have enough observations.

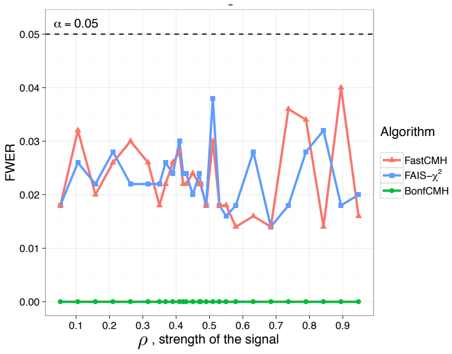


Figure 2.12: **Comparison of family-wise error rates.** Family-wise error rate of FastCMH, FAIS-χ^2 and Bonf-CMH, when each algorithm specifies the target FWER to be $\alpha = 0.05$. In this experiment, the FWER is also controlled with the use of FAIS-χ^2 as the false positives that are accounted for are only the non-confounded ones.

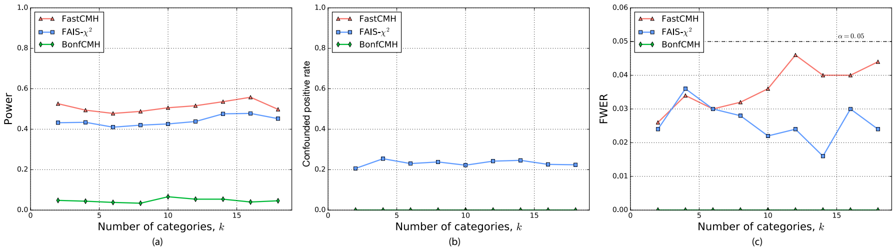


Figure 2.13: **Impact of the number of categories of the covariate on statistical metrics.** A comparison of (a) statistical power, (b) proportion of confounded regions falsely detected and (c) FWER for FastCMH, FAIS-χ^2 and Bonf-CMH as the number of categories of the covariate, k, varies.

Speed: Figure 2.11(c) shows that FastCMH is also dramatically faster than Bonf-CMH for large p. In these two experiments, we did not include associated regions as preliminary experiments showed that it did not have influence on the runtime. For example, Bonf-CMH would take over 24 hours to process a dataset with $p \approx 5 \times 10^5$ (vertical grey dashed line), whereas FastCMH would take less than a minute. Moreover, FastCMH is virtually as fast as FAIS-χ^2, showing that our method can correct for confounders with negligible runtime overhead. Figure 2.14 also contains experiments which show that the runtime of FastCMH scales linearly with the number of samples n. In addition to the methods described above, we show in Figure 2.11(d) that our implementation of FastCMH is several orders of magnitude faster that a naive implementation of Tarone's trick applied to CMH, as described Section 2.4.4, Lemma 2 and Theorem 2. In fact, the computation time of this naive method increases exponentially as k increases, while FastCMH increases only almost linearly, in $O(k \log k)$. This empirically confirms the theoretical result described Section 2.4.4 regarding the scalability of their search space pruning condition for the CMH test, also in the context of genetic heterogeneity discovery.

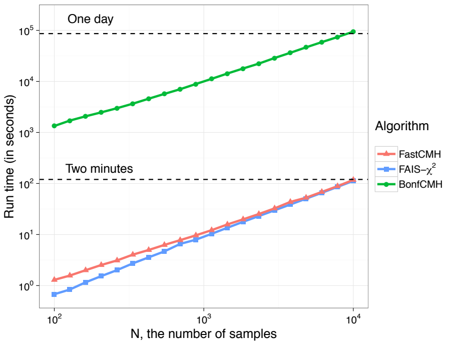


FIGURE 2.14: **Impact of the number of samples on runtime.** Runtime of FastCMH, FAIS-χ^2 and Bonf-CMH, as the number of samples, n, varies.

2.5.3.3 *Evaluation of the ability of FastCMH to find regions of genetic heterogeneity*

This section is devoted to describing an exhaustive comparison between FastCMH and multiple variations of burden tests. Importantly, we will consider two types of encodings to summarise SNPs in a genomic region into a single genomic score (OR and sum), as well as two ways to select the set of candidate regions to be tested (window-based and gene-based), as detailed next.

Encoding in burden tests: Two alternative encodings have been used to collapse the SNPs in each candidate region into a region vector:

(I) an indicator of the presence of any number of minor allele in the region, equivalent to the encoding used by FastCMH.

(II) the count of minor alleles in the region.

Moreover, burden tests were evaluated under different simulation setups. Therefore, in addition to the presentation of the results we will also describe the data generation process of the different simulation experiments in the paragraphs below.

Window-based burden tests:

Data generation: To represent the biological diversity of causal regions each dataset included seven truly associated genomic regions of different length ℓ, with $\ell \in [2, 4, 6, 8, 10, 12, 14]$. The same layout was applied to the confounded genomic regions. When simulating the data, we ensured that the confounded regions were far apart from each other in order to have distinct signals for all associated regions. The strength of the associations between the truly associated and confounded genomic regions and the phenotypic trait are controlled by $\rho \in [0, 1]$.

Burden test: We used two types of windows in the burden tests, namely **non-overlapping** and **sliding** windows. For both of them, w was the size of the window that was tested and varied across the burden tests. For the burden tests with sliding windows, $inc = 1$ was the number of variants (or stride) between the starting positions of two consecutive windows. This is illustrated in Figure 2.15. In the literature [120, 121], strides of length $inc \in [\![1, 2]\!]$ are considered, as well as a stride of length $inc = w/2$, however the most common choice is $inc = 1$. For the burden tests with non-overlapping windows, the alignment of the window boundaries with the boundaries of the truly associated genomic regions has a strong influence on the power of the burden test. In the analysis described below, our goal is to compare `FastCMH` against the most favourable scenario for burden tests. Therefore, the starting position of all truly associated regions is chosen to coincide with the starting position of one of the windows tested by the non-overlapping burden test baseline. The tested windows can be smaller or bigger than the truly associated region, but since there will always be a window that starts in the same location as the region, this puts the burden tests in a more favourable position by minimising the impact of the fragmentation of the genome in windows of size w. Note that, additionally, this unrealistic advantage we concede the non-overlapping window burden tests in our simulation setup implies they will dominate the sliding-window baselines.

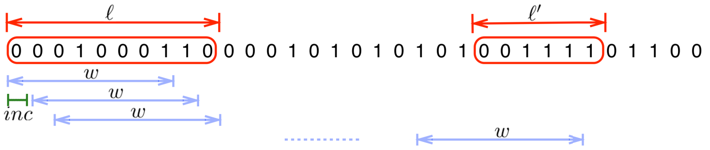


FIGURE 2.15: **Illustration of the sliding windows mechanism.** Sliding windows (in blue) containing $w = 8$ genomic variants. The stride of one variant ($inc = 1$) between two consecutive windows is indicated in green. All of the windows are tested. Two truly associated genomic regions are represented in red with lengths $\ell = 10$ and $\ell' = 6$.

Gene-based burden tests: As opposed to window-based burden tests, gene-based burden tests do not take into account all genomic variants but only predefined regions of interest, normally based on prior biological knowledge.

Data generation: In the simulations, two associated regions of length 8 are generated, one truly associated with the phenotype and one confounded with the phenotype.

Burden test: In the gene-based burden tests, the genomic regions that are tested –also referred as windows in this analysis– are all of length w and each pair of consecutive tested windows are separated by exactly w variants. The idea behind this setting is to simulate the existence of variants that are never tested by gene-burden tests. In order to define the tested windows, two parameters w and f were used. One of the windows overlaps with the truly associated region and one window overlaps with the confounded region. The parameter f measures the overlap of a region (either the truly associated or the confounded) to a tested window. More precisely, f is equal to the proportion of the w variants of the tested window that are contained in a region (again, truly associated or confounded). Intuitively, f allows controlling the amount of misspecification between the prior knowledge and the ground-truth, allowing to study the robustness of gene-based burden tests.

The gene-based burden tests were performed under seven combinations of (w, f) as shown in Table 2.2. Figure 2.16 illustrates the interplay of the parameters w, f and ℓ in the simulated data.

Case	Parameters	
	w	f
(a)	8	0
(b)	8	$\frac{1}{4}$
(c)	8	$\frac{1}{2}$
(d)	8	$\frac{3}{4}$
(e)	10	$\frac{4}{5}$
(f)	4	1
(g)	8	1

TABLE 2.2: **Combinations of parameters w and f used in simulation experiments for burden tests.**

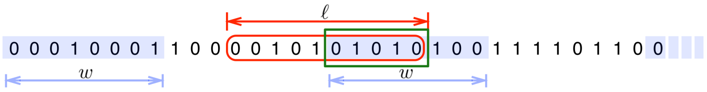


FIGURE 2.16: **Illustration of the gene-based burden test mechanism.** The burden tests will only be conducted on the regions of w variants marked in blue. For simplicity, we assume that these regions correspond to genes and that all genes have the same number of variants ($w = 8$ in the figure). The region highlighted in red with length $\ell = 8$ is the truly associated genomic region. The overlap between the truly associated region and a gene is shown in green. The value of f is equal to the proportion of the w variants in the gene that are also contained in the truly associated genomic region. In this example, $f = \frac{5}{8}$.

Experimental setup: Unless stated otherwise, for all methods, the target FWER is set to $\alpha = 0.05$. All experiments are performed with $n = 500$ samples and

l = 100,000 variants. All results are averaged over 200 iterations. We compare FastCMH to FAIS-χ^2 and Bonf-CMH. For each fixed ρ, we calculate the power, the confounded positive rate and the FWER by averaging the results over all associated genomic regions and by correcting with a Bonferroni correction factor equal to the total number of tests performed. In this study we did not take into account the dependence between the tests. The encoding of the burden tests is detailed in each simulation separately.

Confounded positive rate and FWER in window-based burden tests: In Figures 2.17(a) and 2.18(a), all conditional tests succeed in not discovering the confounded genomic regions while FAIS-χ^2, which does not condition on confounders, reports them. Figures 2.17(b) and 2.18(b) show that FastCMH and the burden tests ensure a good control of the FWER; in particular FastCMH has a FWER below the threshold $\alpha = 0.05$.

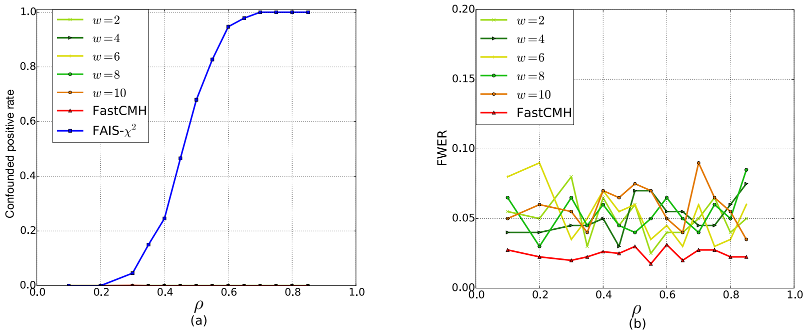


FIGURE 2.17: **Confounded positive rate and FWER in window-based burden tests, with non-overlapping windows.** Burden tests, FAIS-χ^2 and FastCMH are compared. The length of the windows w in the burden tests varies between 2 and 10. (a) Proportion of confounded regions falsely detected (confounded positive rate). (b) FWER for FastCMH and for the burden tests. Burden tests use Encoding (I) in this figure, however the results are similar with Encoding (II).

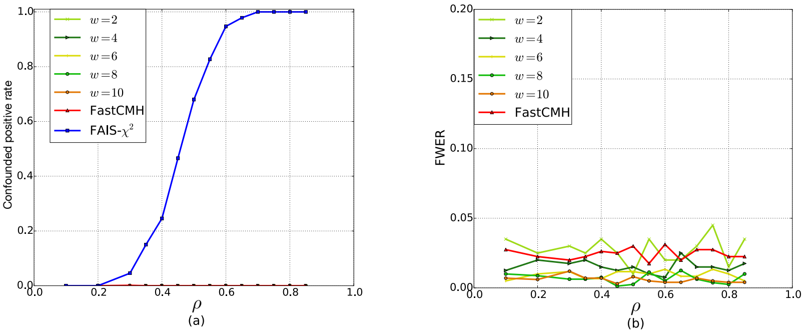


Figure 2.18: **Confounded positive rate and FWER in window-based burden tests, with sliding windows.** Burden tests, FAIS-χ^2 and FastCMH and compared. The length of the windows w in the burden tests varies between 2 and 10. (a) Proportion of confounded regions falsely detected (confounded positive rate). (b) FWER for FastCMH and for the burden tests. Burden tests use Encoding (I) in this figure, however the results are similar with Encoding (II).

Statistical power in window-based burden tests: Figure 2.19 shows the power of the burden tests with Encoding (II), and the power of FastCMH as a function of the strength of the association ρ between the associated genomic regions and the phenotype. The figure illustrates the results for both non-overlapping windows (Figure 2.19(a)) and for sliding windows (Figure 2.19(b)). In both cases, we observe that FastCMH achieves better power than both window-based tests, regardless of the size of the tested windows. This is mainly due to the flexibility of our method FastCMH, which is able to simultaneously detect associated regions of different lengths, combined with an efficient correction for multiple hypothesis testing. In contrast, the window-based tests exhibit low power for all window sizes. This is due to the fact that the associated genomic regions are split over several tested windows, which are in general weakly correlated with the phenotype as they combine part of the associated variants with non-associated ones. Moreover, as soon as the correlation between the signal and the phenotype is large enough, i.e. larger than $\rho = 0.6$, FastCMH's statistical power remains very close to 1.

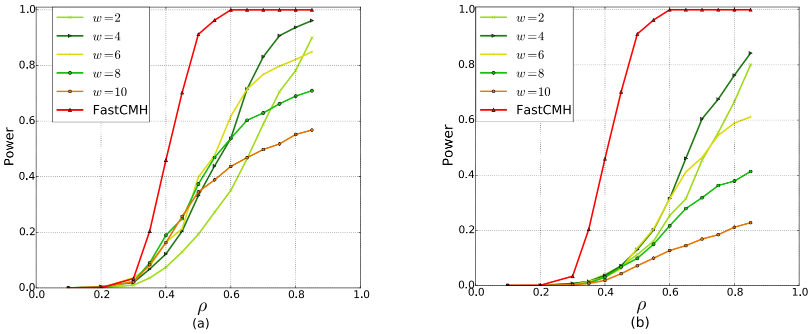


FIGURE 2.19: **Statistical power in window-based burden tests.** Burden tests and `FastCMH` are compared. The length of the windows w in the burden tests varies between 2 and 10. (a) `FastCMH` is compared to non-overlapping window-based burden tests. (b) `FastCMH` is compared to sliding window-based burden tests. The results are obtained with Encoding (II), as it is slightly more favorable than Encoding (I) in the case of window-based burden tests with non-overlapping windows.

Statistical power with varying length of the associated regions in window-based burden tests: We perform an additional set of experiments to evaluate the impact of the length ℓ of the associated genomic regions on the statistical power of the burden tests with non-overlapping and sliding windows.

Figure 2.20 shows how ℓ strongly impacts the power of the burden tests with non-overlapping windows of size w. We observe that, except in some rare configurations of parameters ($w = \ell$), `FastCMH`'s statistical power is higher than the power of the burden tests in all settings, independently of the lengths of the window being tested w and of the associated region ℓ. When $w = \ell$, the non-overlapping windows burden test achieves a similar power to that of `FastCMH`. This setting is particularly beneficial to the burden tests using non-overlapping windows, in particular in the setup we chose where each associated genomic region is perfectly aligned with one tested window. In other words, this special setting corresponds to assuming that the prior knowledge is perfectly aligned with the ground truth. However, the statistical power of the burden tests drops rapidly when $|\ell - w|$ increases (length mismatch or misalignment). In practice, neither the location of the truly associated genomic regions nor their length are known a priori. Thus, different tests with non-overlapping windows of different lengths have to be performed, leading to a loss of power as the Bonferroni correction becomes larger, or alternatively, power might be lost due to misspecification, as shown in the experiments.

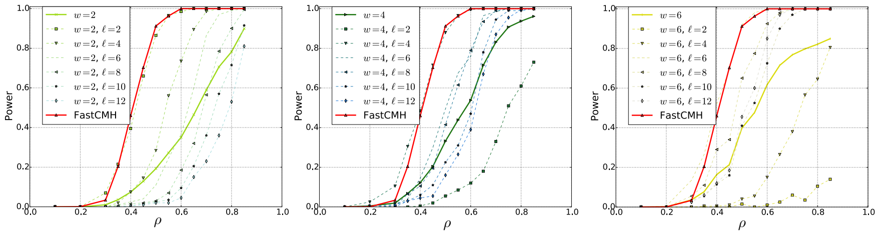


Figure 2.20: **Comparison of statistical power, between `FastCMH` and burden tests, with non-overlapping windows.** The lengths w of the windows and of the truly associated genomic region ℓ vary, $w \in [2,4,6]$ and $\ell \in [2,4,6,8,10,12,14]$. The thick red curves describe the statistical power of `FastCMH`. The other thick curves indicate the average of the statistical power across all burden tests for all genomic regions lengths ℓ. The thin dashed curves represent the statistical power of the burden tests for each length of the associated region separately. Encoding (I) is used, however the results obtained with encoding (II) lead to the same conclusions.

Figure 2.21 shows how the length ℓ of the associated genomic regions has an impact in the power of the burden tests with sliding windows of size w. `FastCMH` clearly outperforms the burden tests in all settings, independently of w and ℓ. For a fixed window-size w, the statistical power of the burden tests varies with the length of the associated region ℓ. Indeed, for each length of the associated region, it partially or fully overlaps with several windows. The distribution of the partial or full overlaps of the tested windows with respect to the associated region, depends on: a) the stride (*inc*) between two consecutive windows, b) the length of the associated region ℓ and c) the size of the windows w. These three factors strongly influence the power of the window-based tests. For example, if the window is large compared to the size of the associated region $w > \ell$ (cases $w = 6$ with $\ell = 2$ and $\ell = 4$), the power of the tests has a dramatic drop as the window includes many, relative to ℓ, irrelevant variants that contaminate the signal with noise. If $w \ll \ell$ (cases $w = 2$ with $\ell = 10$ and $\ell = 12$ and case $w = 4$ with $\ell = 12$), the windows do not contain enough of the truly associated variants to be significantly associated with the phenotype and the burden tests also perform poorly in these cases. The power of the burden tests with sliding windows increases when the overlapping windows are both small enough to only include associated variants and large enough to include a large fraction of the signal, so that the region can be detected (case $w = 2$ with $\ell = 4$ and case $w = 4$ with $\ell = 6$).

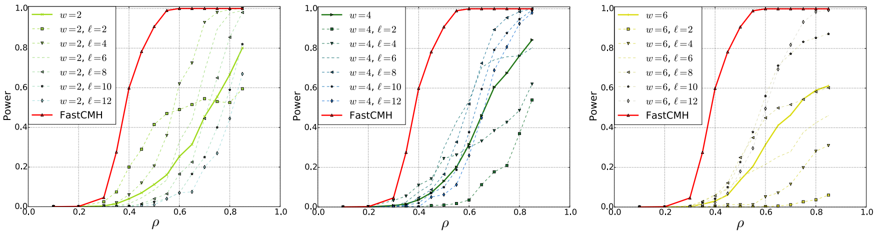


FIGURE 2.21: **Comparison of statistical power, between FastCMH and burden tests, with sliding windows.** The lengths w of the windows and of the truly associated genomic region ℓ vary, $w \in [2,4,6]$ and $\ell \in [2,4,6,8,10,12,14]$. The thick red curves describes the statistical power of FastCMH. The other thick curves indicates the average of the statistical power across all burden tests for all genomic regions lengths ℓ. The thin dashed curves represent the statistical power of the burden tests for each length of the associated region separately. Encoding (I) is used, however the results obtained with encoding (II) lead to the same conclusions.

To conclude, we want to stress the fact that burden tests with window-based approaches are very sensitive to inaccuracies due to incomplete or erroneous coverage of the associated regions by the tested windows. Contrary to this, FastCMH successfully circumvents the problem by testing all possible lengths/starting positions of (testable) genomic regions. We believe this to be a better fit to the reality of genome-wide association studies, which in practice are often exploratory and little is known about the size and the position of associated genomic regions *a priori*. The experiments presented above confirm the effectiveness of our method FastCMH when compared to window-based burden tests in settings when little is known about the ground truth.

Statistical power, confounded positive rate and FWER in gene-based burden tests: Here we present simulations to compare FastCMH with burden tests that only test predefined regions of interest. For our simulations, we consider both encodings, i.e. Encoding (I) and Encoding (II). The generated data contains one truly associated region and a confounded one, both of length 8. On the y-axis of Figures 2.22, 2.23 and 2.24 we show the resulting values of three statistical metrics: 1) the power, 2) the proportion of confounded regions falsely detected (CPR) and 3) the FWER, respectively. The x-axis represents the strength of the association ρ between each of the two associated regions and the phenotype. Burden tests give different results in all seven settings, because they ignore variants outside genomic windows; this is not the case for the two Tarone-based algorithms nor for Bonf-CMH. In Figure 2.22, we observe that the burden tests have a higher power in case (g) than FastCMH and FAIS-χ^2 because exactly all the variants of the associated genomic region are combined in the tested window, while FastCMH performs better in all the other settings, despite the much larger number of tests. We observe that Encoding (II) is slightly more favorable to the burden tests as it sums the single signals, instead of taking the maximum as in Encoding (I), making the combination more robust to noise.

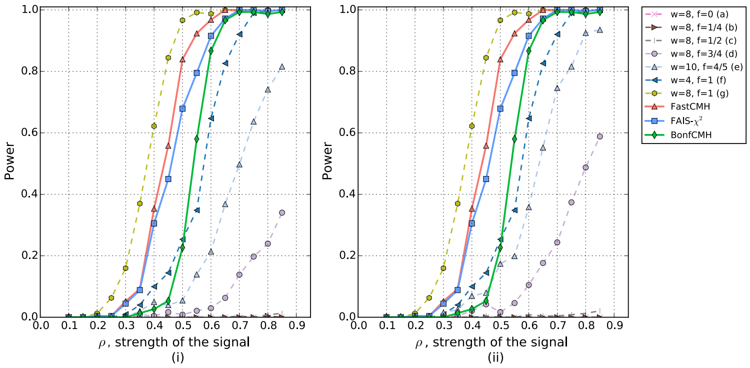


Figure 2.22: **Statistical power as a function of the strength of the signal ρ.** Comparison between FastCMH, FAIS-χ^2, Bonf-CMH and the gene-based burden tests. (i) and (ii) refer to the encoding, Encoding (I) or Encoding (II) respectively, that is used for the gene-based burden tests. The labels (a) to (g) refer to the seven parameter settings for the burden test baselines, which describe different windows sizes and levels of overlap between the tested windows and the associated region. The power of gene-based burden tests in cases (a), (b) and (c) is close to 0.

Regarding the probability of detecting the confounded genomic region, shown in Figure 2.23, all tests, except for FAIS-χ^2 that does not condition on confounders, succeed in ignoring the confounded region. Finally, in Figure 2.24, we observe that all the Tarone-based methods (FastCMH and FAIS-χ^2) ensure a slightly better control of the FWER than the burden tests do, the probability to have at least one non-confounded false positive is marginally smaller when using FastCMH and FAIS-χ^2 than with its comparison partners.

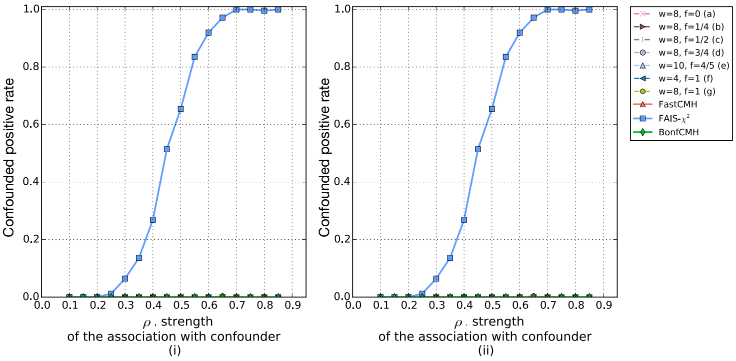


FIGURE 2.23: **Confounded positive rate as a function of the strength of the signal ρ.** Comparison between FastCMH, FAIS-χ^2, Bonf-CMH and the gene-based burden tests. (i) and (ii) refer to the encoding, Encoding (I) or Encoding (II) respectively, that is used for the gene-based burden tests. The labels (a) to (g) refer to the seven parameter settings for the burden test baselines, which describe different windows sizes and levels of overlap between the tested windows and the associated region. None of the methods, except for FAIS-χ^2, retrieve the confounded genomic region.

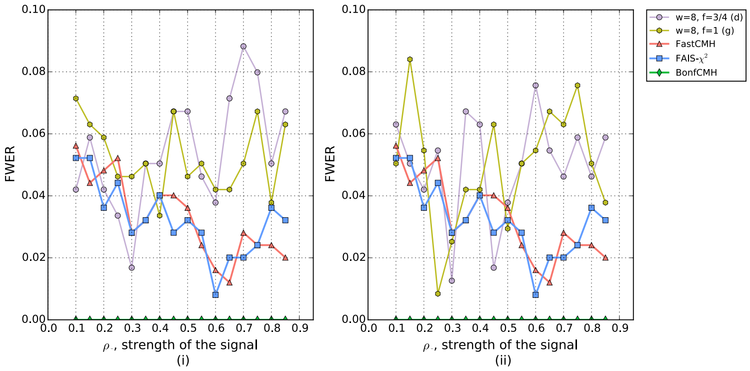


Figure 2.24: **FWER as a function of the strength of the signal ρ.** Comparison between `FastCMH`, `FAIS-`χ^2, `Bonf-CMH` and the gene-based burden tests. (i) and (ii) refer to the encoding, Encoding (I) or Encoding (II) respectively, that is used for the gene-based burden tests. The labels (d) and (g) refer to two of the seven parameter settings for the burden test baselines, which describe different windows sizes and levels of overlap between the tested windows and the associated region. For the sake of clarity, only two burden test cases are shown, cases (d) and (g). However, the FWER variations for the other gene-based burden tests were similar. As explained above, the FWER measures the probability that at least one non-confounded false positive region is deemed significantly associated.

In summary, gene-based burden tests exhibit low statistical power and appear to be inefficient at finding genomic regions that are not almost identical to predefined regions of interest. In contrast, `FastCMH` retrieves associated genomic regions with high power, without the need for predefined biological knowledge to guide the search, while also correcting for confounders.

Combining the simulation results presented for window-based burden tests with the results on gene-based burden tests suggests that `FastCMH` has superior performance in exploratory genome-wide association studies, due to being robust to misspecification of the set of genomic regions to be tested.

2.5.4 *Application to GWAS datasets*

In this section, we present the real-world GWAS datasets that we use to evaluate `FastCMH`: i) a case/control study of association with chronic obstructive pulmonary disease (COPD) in humans and ii) five plant datasets of the model organism *A. thaliana* involving different binary phenotypic traits.

2.5.4.1 *Description of the datasets and preprocessing*

Human data: We analyzed samples from the COPDGene study [122] whose goal is to identify genetic risk factors for COPD. Participants of the study belong to two different ethnic groups: African Americans and non-Hispanic whites. The samples of the two populations were combined and 615,906 SNPs found in the intersection were kept. The combined dataset contains 7,993 samples of which 3,633 are cases and 4,360 are controls. Finally, each SNP was binarised according to a dominant encoding. In this way, significantly associated genomic regions can be interpreted as regions for which the presence/absence of at least one minor allele in the entire region of interest is associated with disease risk for COPD. We use the dominant encoding to binarise the SNPs, i.e. each pair of SNPs containing at least one minor allele is encoded as a 1 and pairs of SNPs containing major alleles only are encoded as 0. We chose this encoding as we were preliminary interested in regions where the presence of at least one minor allele would be associated to the phenotype of interest. However, other encodings could alternatively be tested.
Plant data: We analyzed a widely used *A. thaliana* GWAS dataset by [117] from the easyGWAS online resource [118]. This dataset contains a large collection of 107 phenotypes, 21 of which are dichotomous. We kept five phenotypes: LY and LES (lesioning or yellowing leaves traits) and *avrB*, *avrPphB* and *avrBpm1* (hypersensitive-response traits) with large genomic inflation factors (see Table 2.3 in Section 2.5.4.3). Each of the five *A. thaliana* datasets contains between 84 and 95 inbred samples and approximately 214,050 homozygous SNPs. For each *A. thaliana* dataset, the SNPs were encoded as binary vectors as each plant-sample is inbred.

2.5.4.2 *Definition of the covariates*

The ability of `FastCMH` to handle categorical covariates can be used to correct for confounding variables. In the COPD study, defining the covariate is straightforward: we define the categorical covariate $\boldsymbol{c}$ as the (known) genetic ancestry of the individuals, namely African Americans or non-Hispanic whites. To illustrate both the ability of `FastCMH` to cope with several covariates simultaneously and to handle a large number of categories k for each covariate, we also consider "height" [123] as an additional covariate, discretised into decile bins. For each of the *A. thaliana* datasets, the categorical covariate $\boldsymbol{c}$ we condition on to correct for population structure was defined using k-means clustering on the three principal components of the empirical kinship matrix [28], with k optimized to minimize genomic inflation (see Table 2.3).

2.5.4.3 *Results*

Here we discuss the results we obtained when analysing the human and plant data. We first present our findings with respect to the correction of confounding factors, followed by a description of the significant genomic regions that our method discovered. Finally, we provide a comparison with burden tests [61].

Dataset and phenotype	Samples n	Cases %	k	FAIS-χ^2 λ	FAIS-χ^2 Hits	FastCMH λ	FastCMH Hits
COPDGene							
▷ COPD	7,993	45.4	20	16.70	88,403	1.05	3
A. thaliana							
▷ *avrB*	87	63.2	3	1.66	14	1.17	11
▷ *avrRpm1*	84	66.7	3	1.53	15	1.13	13
▷ *avrPphB*	90	51.1	4	1.70	6	1.22	5
▷ LES	95	22.1	3	2.05	20	1.21	3
▷ LY	95	30.5	5	2.51	26	1.30	1

TABLE 2.3: **FastCMH and FAIS-χ^2 results on the GWAS datasets.** FastCMH is compared to the previous state-of-the-art algorithm (FAIS-χ^2), which cannot correct for covariates. For each method, the columns λ and "Hits" refer to the genomic inflation factor and the resulting number of non-overlapping genomic regions deemed significant, respectively. The value of λ is computed based on the test statistics of all testable regions.

Population structure correction: In Table 2.3 we show that the results of FAIS-χ^2 for all five *A. thaliana* datasets exhibit a moderate-to-severe degree of genomic inflation [108], with λ ranging between 1.53 and 2.51. In all instances, FastCMH successfully manages to correct for population structure, reducing the inflation to a range of 1.13 to 1.30.

The ability of FastCMH to correct for population structure becomes even more evident with the results from COPDGene. Firstly, there are marked genetic differences between individuals of African American and non-Hispanic white ancestry, as illustrated Figure 2.25. This coupled with the shift in the ratio of cases/controls across populations (30.81% for African Americans versus 52.81% for non-Hispanic whites) causes an extreme level of inflation that FAIS-χ^2 is unable to correct for. With a genomic inflation factor of $\lambda = 16.70$, any hit reported by FAIS-χ^2 becomes unreliable. In contrast, FastCMH eliminates the inflation almost entirely by reducing it to $\lambda = 1.05$. To further illustrate the effects of population structure correction when using FastCMH versus FAIS-χ^2, in Figure 2.26 we show QQ-plots of p-values for all testable genomic regions in three selected datasets: two for *A. thaliana* (LES and LY) and the COPDGene study. Based on Figure 2.26, we can conclude that FastCMH can successfully correct severe levels of genomic inflation. Additionally, we investigated the possibility of further correcting for population structure in the COPDGene study by defining the categorical covariate using k-means on the top three principal components of the empirical kinship matrix, as we did when analyzing the *A. thaliana* datasets. This leads to a further decrease in genomic inflation, as shown in Figure 2.25(b), without affecting the set of genomic regions deemed significantly associated by the FastCMH algorithm.

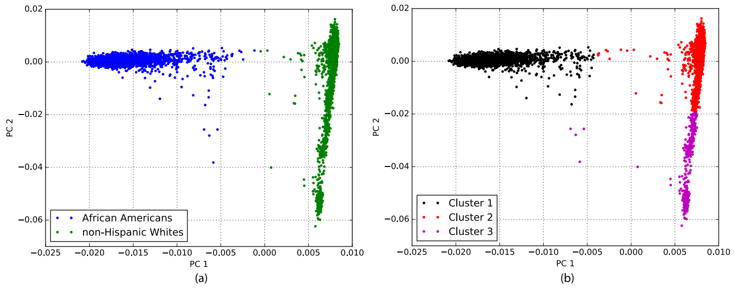


Figure 2.25: **Two-dimensional representation of all** 7,993 **samples in the COPDGene study.** The embedding of the samples in the COPDGene study is done according to the two principal components of the kinship matrix: (a) Individuals coloured according to ethnicity ($\lambda = 1.048$). (b) Individuals coloured according to the category assigned by k-means clustering ($k = 3$) on the three principal components of the kinship matrix ($\lambda = 1.016$).

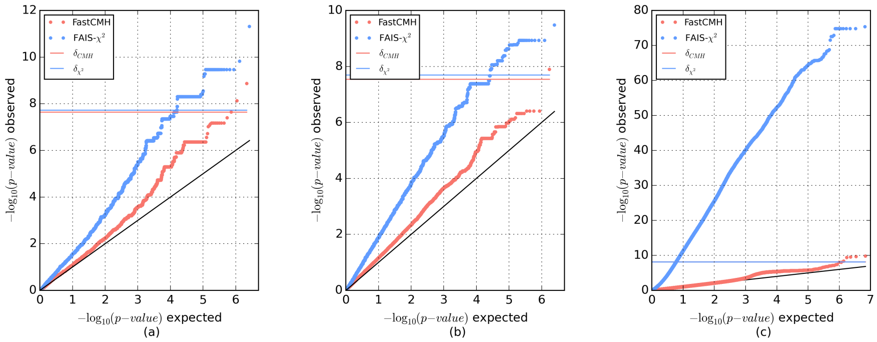


Figure 2.26: Comparison of the QQ-plots for the p-values of all testable genomic regions obtained with FastCMH (red) and the previous state-of-the-art algorithm FAIS-χ^2 (blue) for three datasets: (a) *A. thaliana* LES (b) *A. thaliana* LY (c) COPDGene. Horizontal lines correspond to the adjusted significance thresholds.

Finally, Figure 2.27, we study the impact of k, the number of categories of the covariate, on the runtime of FastCMH. We ran the analysis on the COPDGene data with different levels of discretisation of the covariate "height". Our results are consistent with the trend observed in Figure 2.11(d): the runtime of FastCMH scales smoothly with k, while approaches based on naive evaluations of the pruning criterion scale exponentially with k. This severely limits their applicability, being only feasible for

$k < 16$, a limitation not present for `FastCMH`.

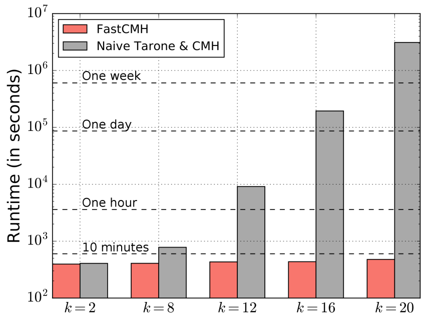


FIGURE 2.27: **Runtime of `FastCMH` and a naive implementation of the lower envelope for the CMH test in the COPDGene study.** The x-axis represents different values of the number of categories for the covariate.

Significantly associated genomic regions: In Table 2.3, we also show the number of non-overlapping genomic regions deemed statistically significant (hits) by our method, `FastCMH`, and our comparison partner, `FAIS`-χ^2. Both algorithms were run with a target FWER of $\alpha = 0.05$.

Across all five *A. thaliana* datasets, we observe that `FastCMH` retrieves systematically less genomic regions (33 in total) than `FAIS`-χ^2 (81 in total). Moreover, the decrease in the number of hits is larger for those datasets with stronger genomic inflation. For instance, in LY ($\lambda = 2.51$ for `FAIS`-χ^2), our method retrieves a single genomic region, while `FAIS`-χ^2 retrieves 26. Similarly, in LES ($\lambda = 2.05$ for `FAIS`-χ^2), our method has 3 hits while `FAIS`-χ^2 reports 20. Based on the results presented in Section 2.5.3, and the correlation in the decrease of number of hits with genomic inflation, it is plausible to conclude that the results of `FAIS`-χ^2 can be inflated by population structure, while `FastCMH` successfully corrects for such inflation. Finally, it is worth noting that out of the 33 significantly associated genomic regions retrieved by `FastCMH` in the *A. thaliana* datasets, 17 of them did not contain any SNPs which were deemed significant by a single-SNP association study, illustrating how mining genomic regions can lead to the discovery of novel associations. The most significant genomic regions and their respective p-values are shown in Appendix Section A.3.

Our results for the COPDGene study also clearly demonstrate the need to correct for population structure while mining significant genomic regions. `FAIS`-χ^2 reports a very large number of hits (88,403 hits), mainly due to the extreme genomic inflation ($\lambda = 16.70$). In contrast, `FastCMH` reports only 3 significantly associated genomic regions. Each of the three regions respectively overlaps with a gene in the gene cluster known as the (CHRNA5-CHRNA3-CHRNB4) nicotinic acetylcholine receptor (nAChR), located on chromosome 15q25.1. Independent studies have reported individual as well as joint association of some of these genes to COPD [123, 124]. What is remarkable about our results is that the 3 regions detected by FastCMH are formed by SNPs, each of which do not seem to have an association to COPD individually, but it is their joint effect across genetically different populations that is strongly associated to the disease. Details about the SNPs involved, their locations, and individual as well as region-based p-values are shown in Appendix Section A.4. When analysing both populations independently with FAIS-χ^2, no significant region was found in the African American cohort whereas only one region was reported for the non-Hispanic whites. With this, we conclude that the main advantage of our method relies on attaining statistical power, not only through an efficient mechanism that avoids testing untestable regions, but also by allowing the analysis of larger datasets with samples of mixed populations thanks to a reliable and computationally efficient correction of confounding factors.

Comparison with burden tests: While FastCMH aims to test all genomic regions, burden tests are often applied to genes exclusively, or other predefined genomic regions, because their runtime and/or statistical power degenerates when testing all possible genomic regions. To illustrate the usefulness of exploring all genomic regions we ran different kinds of burden tests for all five *A. thaliana* datasets as well as for the COPDGene study as additional baselines. We ran (a) gene-based burden tests on genomic regions defined by all genes extended by 10kb on both sides, resulting in 24,426 regions for *A. thaliana* and 17,817 regions for COPDGene, and (b) window-based burden tests on contiguous non-overlapping windows of either 500 kilobases or 1 megabase. We used Encodings (I) and (II). All types of burden tests were performed using both the likelihood ratio test under a logistic regression model with the categorical covariate encoded using k dummy indicator variables as well as using the CMH tests in the same way FastCMH does.

For *A. thaliana*, 45% of all the SNPs discovered by FastCMH are not inside genes and, as a result, were not discovered by the gene-based burden tests. FastCMH also leads to results complementary to those of the gene-based burden tests at the gene level: 21% of the genes reported by any of the burden tests are also found by FastCMH, including the most significant ones. However, it is important to note that the different variations of the gene-based burden-tests in this study show a substantial variability in the set of genomic regions deemed significant. This, together with the fact that, as shown in Figure 2.28 and Table 2.4, many of the gene-based burden tests appear to suffer from confounding more strongly than FastCMH, as quantified by the genomic inflation factor λ, suggest that FastCMH retrieving only 27% of the hits of all burden tests is neither surprising nor indicative of low power. Window-based burden tests found no hit in any of the *A. thaliana* datasets for variants of Encoding (I) and three for those using Encoding (II). At last, out of all genes deemed significant by at least one method, 40% were only retrieved by FastCMH. Details of the SNPs found are reported in Appendix Section A.5.

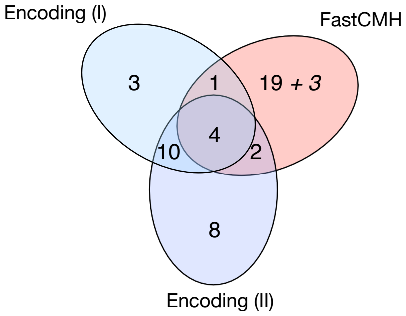


Figure 2.28: **Venn diagram for the genes found by the gene-based burden tests and/or FastCMH.** Encoding (I) represents the union of all the hits of gene-based burden tests that use encoding (I). Likely, Encoding (II) represents the union of all the hits of the gene-based burden tests that use Encoding (II). FastCMH considers both intervals inside genes or intersecting genes (normal font) and intervals fully outside genes (italic font) that are only retrieved by FastCMH. We note that the comparison of FastCMH to gene-based burden tests in the Venn diagram is conservative in the sense that burden test hits are "pooled" across multiple variations of the method, corresponding to different choices for the statistical tests, without an additional correction for the multiple hypothesis testing problem.

Phenotype	FastCMH	Burden tests				
		dummy - (I)	*dummy - (II)*	*PCs - (I)*	*PCs - (II)*	*CMH*
avrB	1.17	1.12	1.22	1.07	1.17	1.05
avrRpm1	1.13	1.13	1.24	1.07	1.15	1.03
avrPphB	1.22	1.41	1.62	1.15	1.27	1.12
LES	1.21	1.43	1.68	1.23	1.43	1.16
LY	1.30	1.44	1.63	1.44	1.63	1.20

Table 2.4: **Genomic inflation factors for all gene-based burden tests and for FastCMH.** Note that for FastCMH, the genomic inflation factor is calculated using p-values for testable genomic regions only, leading to an inflated genomic inflation factor. *dummy* indicates that the covariates are coded as k dummy indicator variables, *PCs* means that we chose the three first principal components of the kinship matrix as covariates, *(I)* and *(II)* correspond to the encodings described above. Finally, *CMH* corresponds to the gene-based burden test using the CMH test applied to encoding *(I)* for each gene.

Concerning the COPD study, none of the three genes (CHRNA5-CHRNA3-CHRNB4) found by FastCMH were significant using any of the gene-based or window-based burden tests. Taking the smallest p-value across all burden tests performed, only CHRNB4 was close to significance (p-value $5.72 \cdot 10^{-6}$) while CHRNA5 and CHRNA3 had p-values 0.24 and 0.41, respectively. While each of the three significantly associated genomic regions found by `FastCMH` overlaps with one gene in the cluster (CHRNA5-CHRNA3-CHRNB4), the significant regions do not span the entire gene, suggesting that the ability of `FastCMH` to efficiently test regions of any size and starting position might be instrumental in successfully retrieving there association. Complementary details of the results of burden tests on the COPD dataset are available in Appendix Section A.6.

In summary, `FastCMH` is not to be understood as a substitute for burden tests, but as a complementary approach that allows testing a much broader range of hypotheses, allowing the discovery of novel associations which would otherwise be missed by burden tests.

2.6 software package for the detection of statistically significant snp combinations

In order to give access to methods inspired from the field of significant pattern mining to a broader audience, we built an open source, efficient, user-friendly software package called CASMAP. Although the software package allows for general applications of significant pattern mining, it was developed with a strong focus towards GWAS. In particular, the software package leverages FAIS-χ^2, FastCMH and FACS in order to be able to account for higher-order feature interactions and genetic heterogeneity in GWAS studies. Additionally, both methods FastCMH and FACS leave the possibility to account for a categorical confounding covariate, such as age or gender, making it particularly suitable for GWAS.

Compared to MP-LAMP [125], the state-of-the-art significant pattern mining-based software package for GWAS at the time of publication, the contributions of CASMAP are twofold: i) it allows for the **correction of covariates**, such as age or population structure, which could lead to the detection of spurious associations if not taken into account and ii) it provides **methods to carry out region-based association study** accounting for all starting positions and lengths of the regions, in addition to **conducting higher-order epistasis search**.

The CASMAP toolbox is easy to install and easy to use. Implemented in C++, it is available both in Python and R and is compatible with tab-delimited text files. The **input** files consist of the sample data, the phenotype and an optional covariate file. After running the analysis, the output of the tools are text files whose contents will depend on which analysis was conducted. The **statistical tests** used are either the CMH test if the user provides a covariate file or the χ^2 statistical test. For **region-based association studies**, the main **output** consists of significantly associated genomic regions, marked by a start and end positions (SNPs), with their respective p-value. To avoid reporting numerous overlapping regions, a clustering post-processing step is performed and the final output contains the results of this step. In **higher-order epistasis analyses**, the main **output** reports the sets of SNPs whose association to the phenotype was found to be statistically significant. In addition, the tool creates output files that contain detailed profiling results and a summary of statistical results.

As proof-of-concept example of the software package, we used the algorithm FastCMH present in CASMAP on the COPD genome-wide association study dataset [122] that was already presented in the experiment section of FastCMH, Section 2.5.4. As a recall, the dataset comprises 7,993 individuals and a total of 615,906 SNPs. Each SNP was binarised according to a dominant encoding. Without stratifying the data, the genomic inflation factor is equal to $\lambda = 16.70$, indicating strong population structure. Therefore, to correct for confounding, a categorical covariate was obtained by clustering in four clusters the six principal components obtained with the method from [28], which led to a reduced genomic inflation factor $\lambda = 1.02$. This example had for objective to demonstrate the computational efficiency of one of our method `FastCMH` and we measured that it ran in approximately 7 minutes.

3

DEEP-LEARNING ENABLES ACCURATE PREDICTIONS OF RIBOSOME BINDING SITE ACTIVITY

3.1 STATE OF THE ART IN RIBOSOME BINDING SITE ACTIVITY PREDICTION

Quantitatively mapping a function to a regulatory sequence is key in multiple biological domains, in particular in synthetic biology where one aims at controlling the circuitry of a cell or component. To this end, models have been built for diverse regulatory regions [30, 31, 126, 127], with the objective to be able to predict accurately and quantitatively the corresponding protein level. Among these regulatory regions, the ribosome binding site (RBS) has been particularly studied [128, 129]. RBSs are part of the 5′-untranslated region (5′-UTR) of mRNAs and controls the rate-limiting initiation of translation. As a few mutations in the RBS sequence can lead to several order-of-magnitude differences between the expression of the regulated gene, RBSs are often used in synthetic biology to optimise protein level [129]. Therefore, in this part of the thesis we will focus on ribosome binding sites.

Several state-of-the-art prediction models have been developed in order to be able to predict quantitatively the function of the regulatory sequence of interest, here the ribosome binding site. Existing models can be classified in several categories, a) models that use a mechanistic hypothesis, such as the RBS calculator [128], b) classical machine learning models such as random forests [130] or c) deep learning models, including convolutional neural networks and fully-connected networks [131, 132]. While using a mechanistic model has the advantage to inject prior information into the model, it is mostly beneficiary to cases where the labelled data available to fit the model has a small sample size (a few hundred samples). As soon as the dataset size increases, it is possible to use classical machine learning models as presented in [130]. However, [130]'s limitations are first that the sequence space observed is rather short (six nucleotides) and second that the sample size is only a few thousands, rendering the usage of more complex, potentially highly accurate models, such as deep learning ones not as valuable as it could be. Two recent publications [131, 132] have shown the importance of generating datasets that contain hundred of thousands of samples. Both studies build a deep learning model that allows to accurately predict RBS function, reaching unprecedented performance above $R^2 = 0.9$ in biology, characterising human and yeast 5′-UTRs, respectively.

With the emergence of this pioneering large sample size work, several questions remain. From the biological side, one central question is the generalisation of the data generation method to other regulatory regions. From the machine learning side, one can ask whether these extremely accurate performance are reliable or not, in other words, how certain is the model about the predicted values in order to be able to use the prediction in downstream tasks.

In summary, the objective of the machine learning part of this project was to support experiment design and to build accurate deep learning models in order to go beyond state-of-the-art performance in function prediction from the sequence without prior mechanistic knowledge. The organisation of this chapter is the following:

- Section 3.2 presents the data generation method designed by the biologists in order to obtain large datasets characterising hundreds of thousands of ribosome binding site sequences.
- Section 3.3 describes the preprocessing step from the raw next-generation-sequencing (NGS) data to the identification of the diversifier sequences, and the steps used to optimise the throughput of the biological experiment from a first proof-of-concept dataset.
- Section 3.4 presents the criteria by which the labels used for training the machine learning model are defined and how well these labels correlate to a standard measure of gene expression in biology.
- Section 3.5 describes the model architecture of SAPIENs (**S**equence-**A**ctivity **P**rediction **In** **E**nsemble of **N**etwork**s**), which is the proposed deep learning model that allows to predict function from sequence in our project. This section also explains how SAPIENs obtains well-calibrated uncertainty estimates of the individual predictions.
- Section 3.6 describes the experimental design of the machine learning experiments, and shows how SAPIENs compares to off-the-shelf machine learning models.
- Section 3.7 describes the insight obtained about the influence of positions and bases of the RBS sequences on their activities.

The manuscript related to this project, entitled *Large-scale DNA-based phenotypic recording and deep learning enable highly accurate sequence-function mapping*, has been accepted in Nature Communications [133]. Further details of the machine learning approach that is described in this thesis are available in the Machine Learning Annex of the paper and the code is available in github.com/LaetitiaPapaxanthos/SAPIENs. This chapter of the thesis is highly inspired from the sections describing the data analysis and machine learning methods of the paper published in Nature Communications. As a reminder, the contributions of the authors to *Large-scale DNA-based phenotypic recording and deep learning enable highly accurate sequence-function mapping* are, as written in the paper: M.J. and Y.B. conceived the project. M.J., Y.B., and K.B.

coordinated the study. K.B. conceived and supervised machine learning analyses. M.J. supervised experimental work. S.H., K.F., and M.J. performed experiments. S.H., K.F., L.P., A.C.G., and M.J. analyzed data. A.C.G., L.P., and M.J. developed measures to increase throughput. S.H. and L.P. developed the algorithm for processing of NGS data. L.P. conceived, developed, and analyzed machine learning models. C.B. advised design of DNA adapters and NGS. M.J., L.P., and Y.B. wrote the manuscript with input from all authors.

3.2 introduction to uaspire for the generation of large sequence-function datasets

Linking variants of genetic regulatory elements to their activity at large scale is a difficult task in biology, and most of the current approaches either yield datasets containing up to 10^3 samples [134], are case-specific and technically challenging [132, 135] or are error- and bias-prone [31, 136]. However, sequence-function mapping is at the center of many biological domains, in particular in synthetic biology. To this end, collaborators in the department of biosystems science and engineering (DBSSE) at ETH Zürich have developed a method in order to yield large-scale datasets, mapping hundred of thousands of sequences of a genetic regulatory element of interest to their respective function, with the potential to map millions.

However, while we are at the verge of being able to generate millions of sequence-function pairs, the sequence space to explore remains too large to hope to be able to cover it entirely. For example, the total number of sequences of length $l = 17$, with an alphabet of four letters (A, C, G, T/U), reaches $N = 4^{17} \approx 10^{10}$ distinct sequences. Therefore, beyond the generation of large datasets, being able to predict accurately the function of any genetic regulatory element sequence becomes crucial for biological research. To this end, deep learning maximises the benefit of large data collection owing to its ability to capture complex, non-linear dependencies and to its computational scalability [32]. Deep learning models can exploit patterns and dependencies in the sequence data in order to predict accurately the function of any input sequence, mapping sequence to function without any prior mechanistic knowledge, and have had a few great successes in applications such as genomics or proteomics [137–142].

In order to obtain a large dataset, the biologists in the collaboration built a three-component DNA construct that contains, on the same DNA molecule, a **diversifier** that is the regulatory element sequence, a **modifier** that is the gene coding for a recombinase whose expression is controlled by the diversifier and a **discriminator** which sequence is modified by the recombinase protein. The state of the discriminator, indicated by its sequence, is binary, modified or non-modified, and is a proxy for recombinase expression, which is regulated by the diversifier sequence (all other regulating factors kept constant). Additionally, we assume that the recombinase is itself a proxy for the expression of any gene that would be regulated by the diversifier. Therefore, the discriminator state is a proxy for the ability of the diversifier sequence to upregulate or downregulate any gene, which we denote the diversifier activity. Figure 3.1 shows the interaction between the three components of the DNA construct. The link between the regulatory sequence (diversifier) and its activity (state of the discriminator) is done with next-generation-sequencing (NGS), as on the same paired-read (forward and backward) both the regulatory sequence and the sequence of the discriminator (modified or unmodified) are present. However, if we only had one discriminator per regulatory element sequence, the resolution of the diversifier activity by a binary state discriminator would be insufficient. Therefore, several copies of the DNA construct containing the same regulatory element sequence are sampled, both at the same time point and at different time points, pooled and fed to NGS together. Therefore the resolution of the activity at a specific time point is increased from a binary state to a ratio of modified discriminators, and this ratio can be obtained at multiple chosen time points. Logically, the more copies per diversifier sequence, the better the resolution. For example, let us assume two regulatory element sequences, one that leads to an increase in recombinase expression and the other one to a non-expressed gene, at a given time point. The more copies, the closer the ratio of modified discriminators will be to 1 in the first case, and to 0 in the second case. In the end, a kinetic profile is obtained for each diversifier sequence, that gives a temporal indication of the evolution of the corresponding gene expression. Each of these kinetic profiles is later aggregated into a single scalar (see Section 3.4.2), which is used as a proxy for the diversifier sequence's activity and therefore of the corresponding gene expression (all other factors controlled).

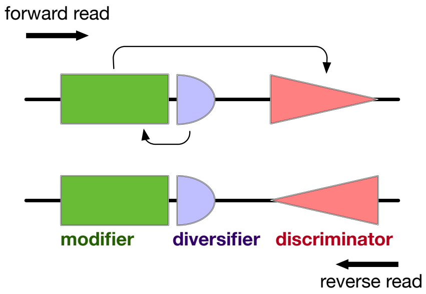


FIGURE 3.1: **Representation of the three-component DNA construct.** The diversifier (green), for example a regulatory sequence, influences the expression of the modifier (purple). If the modifier is expressed, it will change the state of the discriminator (red). By sequencing a single DNA sequence, both the discriminator state and the diversifier will be on the same read, therefore allowing to obtain the sequence and function pair simultaneously.

The idea behind the three-component DNA construct is to be able to trade-off accuracy in the measure of gene expression, substituting GFP measurements against a summary statistic of the kinetic profile, by sample-size, as the new approach would be able to label hundred of thousands of different diversifier variants as its capacity is uniquely limited by the NGS machine capacity.

In practice, the biological experiments are performed in *E. coli*. The diversifier used is a ribosome binding site (RBS) in *E. coli*, the modifier is a coding sequence of the protein recombinase Bxb1 and the discriminator is a short DNA sequence. As said above, RBSs are of high interest in synthetic biology as a few mutations can lead to orders of magnitude of differences between the expression of the regulated gene. The recombinase Bxb1 is a protein that leads to an irreversible DNA sequence inversion (referred as flipping later on). The location of sequence that is inverted induced by the recombinase is highly specific as the sequence that is flipped (the discriminator) is characterised by two attachment sites *attB* and *attP*, which are long (50 pb and 53 pb), therefore avoiding off-target effects. The discriminator is the DNA sequence flanked by the two attachment sites *attB* and *attP*, such that the recombinase being expressed leads to its irreversible flipping.

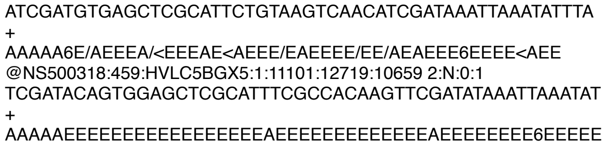
3.3 data preprocessing and optimisation of the experimental throughput

3.3.1 *Data preprocessing of the output of NGS*

The data preprocessing is composed of two steps, a first step presented Section 3.3.1.1 that describes how we preprocessed the fastq files obtained from the NGS machine (Illumina NextSeq, ∼400 millions reads) to pairs of RBS sequence-discriminator sequence, and a second step Section 3.3.1.2 that consists of aggregating the discriminator sequences per RBS sequence and time point.

3.3.1.1 *From raw data to RBS-discriminator pairs*

The output of NGS is composed of fastq files as shown Figure 3.2. Each fastq file is composed of groups of four lines annotating one read and is coupled to another fastq file. One of the four lines corresponds to the actual read of interest. The reads are paired into a forward and a reverse read, extracted from two coupled fastq files. In our experiment, the reverse read contains the RBS sequence of interest together with the 5′-end of the protein Bxb1 and the forward read contains the discriminator sequence, flipped or non-flipped. A first filtering step consists of removing all the paired-reads that contain more than six consecutive unidentified nucleotides (denoted as N). A second filtering step consists of keeping only all paired-reads that contain the 10-bp constant sequence GAGCTCGCAT (5′-end extract of the *bxb1* coding sequence), while allowing for 3 mismatches. Among the remaining reverse reads, the positions of the constant sequences are recorded and the 17-bp RBS sequences are localised directly to the right of the constant sequence. The discriminator sequences are extracted from the forward reads, searching for parts of the part of *attP* and *attR* sites, i.e. for the sequences GGGTTTGTACCGTACA and GCCCGGATGATCCTGAC, allowing for three mistmatches. As a last step, reads that contained more than 8% of errors in the coding sequence were removed to exclude some off-target mutations. In the end of this first preprocessing step, the dataset contained pairs of "RBS sequence-discriminator sequence", grouped by the time point at which the DNA samples were collected.



```
@NS500318:459:HVLC5BGX5:1:11101:21748:10659 2:N:0:1
ATCGATGTGAGCTCGCATTCTGTAAGTCAACATCGATAAATTAAATATTTA
+
AAAAA6E/AEEEA/<EEEAE<AEEE/EAEEEE/EE/AEAEEE6EEEE<AEE
@NS500318:459:HVLC5BGX5:1:11101:12719:10659 2:N:0:1
TCGATACAGTGGAGCTCGCATTTCGCCACAAGTTCGATATAAATTAAATAT
+
AAAAAEEEEEEEEEEEEEEEEEAEEEEEEEEEEEEEAEEEEEEEE6EEEEE
```

Figure 3.2: **Example of two reads in a fastq file.**

3.3.1.2 *From RBS-discriminator pairs to kinetic profiles*

The datasets generated with the method uASPIre (**u**ltradeep **A**cquisition of **S**equence-**P**henotype **I**nter**re**lations) are composed of RNA sequence-flipping profile pairs. The RNA sequences are 17 bp long and located directly upstream of the start codon (Figure 3.3(a)). The flipping profiles are built according to the following steps. The preprocessing pipeline described above allows to record together millions of pairs of forward and reverse reads, which contain respectively the RBS sequence and the discriminator sequence, the latter being in a flipped state or a non-flipped state. At each chosen time point, it allows to collect such pairs for several RBS sequences and multiple times per RBS sequence. For each RBS, at each time point, it is therefore possible to register the counts of flipped and non-flipped discriminator reads (Figure 3.3(b)). For each RBS, the ratio of flipped discriminator reads at each time point enables to estimate the translation kinetics on a predefined time period (Figure 3.3(c)). As a consequence, the datasets that are generated with uASPIre are composed of pairs of 17-bp long RBS sequences and their kinetic profiles (Figure 3.3(d)).

In total, two datasets were generated. A *small* one, composed of $\sim$10,000 sequence-profile pairs. It is used as a proof-of-concept dataset for the biological experiments to optimise the generation of the larger one. A *large* dataset was created afterwards, composed of $\sim$300,000 sequence-profile pairs. This last dataset is used to train off-the-shelf machine learning models, to develop a tailored deep learning model `SAPIENs` (**S**equence-**A**ctivity **P**rediction **I**n **E**nsemble of **N**etwork**s**), interpret the later and do subsequent data analysis.

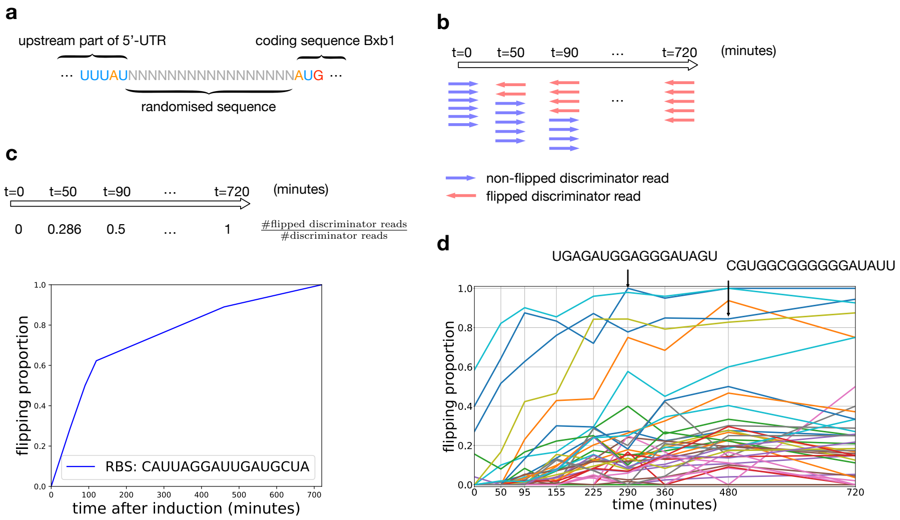


Figure 3.3: **From the read counts to the flipping profiles.** (a) Localisation of the randomised RBS sequence. (b) Flipped and non-flipped discriminator read counts across time, for a single RBS sequence. (c) Flipping ratio across time for a single RBS sequence. (d) RBS sequences and their flipping profiles for several RBS sequences taken at random.

3.3.2 *Simulation and library design to optimise the throughput of NGS*

The *small* dataset contained approximately 10,000 RBS variants sampled uniformly at random (Section 3.3.2.1). This dataset was used to improve the generation of the second dataset, first by running simulations to have an estimate of the number of RBS variants that are retrieved after NGS as a function of the input (Section 3.3.2.2) and by enriching the library in RBSs exhibiting a high activity (Section 3.3.2.3).

3.3.2.1 *Description of the small, proof-of-concept dataset*

The *small* dataset was composed of 10,427 RBS variants sampled uniformly at random. For this first dataset, a provisory label was built by aggregating each flipping profile. The aggregation was done by taking the integral of the flipping profile using the trapezoidal rule and dividing the latter by the interval length on which the integral was estimated. This label is referred to as IFP$_{trz}$ and is contained in $[0,1]$. The IFP$_{trz}$ scalar value is therefore a normalised estimate of the RBS activity and is used as proxy for the protein level regulated by the RBS sequence. A first analysis of the labels of this dataset shows that the distribution of RBS activities is strongly skewed towards weak RBSs, as shown Figure 3.4. This proof-of-concept dataset is then used to optimise the generation of the *large* dataset as explained below.

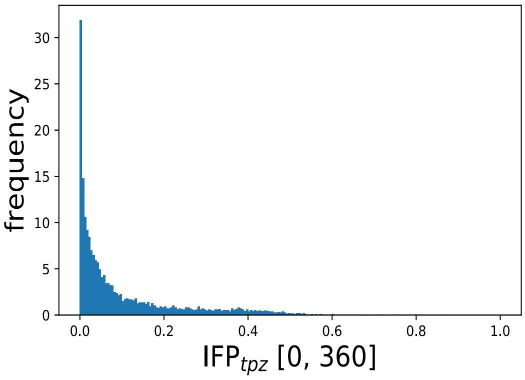


FIGURE 3.4: **Label distribution of the proof-of-concept dataset.** The proof-of-concept RBS library with 17 consecutive, fully randomised bases (N17) upstream of the GFP start codon shows a strong skew towards weak RBSs as represented by the integral of the flipping profile between 0 and 360 minutes after induction approximated using the trapezoidal rule (IFP$_{tpz}[0 - 360]$ min).

3.3.2.2 *Optimisation of NGS loading*

To increase the throughput of uASPIre, we analysed the data from the proof-of-concept (poc) experiment, which contained kinetic data of approximately 10,000 RBS variants. We sought to estimate an optimal number of variants to be loaded into NGS in order to retrieve a maximised number of variants with high quality data (i.e. above different thresholds θ for the minimal read count of discriminators per RBS sequence per time point). For this simulation, we assumed that the limiting factor is the NGS throughput and that the maximal number of valid reads (i.e. reads that pass the preprocessing pipeline quality constraints) retrieved by NGS is constant across experiments under the same experimental conditions. This simulation is based on the idea that increasing the number of RBS variants reduces the coverage and vice versa, as the maximum number of valid reads is constant. For the distribution of read counts we assumed that it follows a log-normal distribution and that its variance is independent of the coverage. The proof-of-concept dataset is composed of approximately 2×10^8 valid reads, which are spread among $n_t = 18$ time points and $n_{poc} = 10{,}427$ variants with an average coverage of $cov \sim 1000$ reads per variant per time point. If the coverage of the small dataset is reduced by a factor of $r_c > 1$, and the number of time points by a factor of $r_t > 1$, the total number of variants that could be loaded into NGS without loss would be $n_{input}(r_c, r_t) = n_{poc} \times r_c \times r_t$, by conservation of the maximal number of valid reads. However, out of these $n_{input}(r_c, r_t)$ variants, only $n_{output}(\theta, r_c, r_t) < n_{input}(r_c, r_t)$ would pass the quality control as enforced by the minimal read threshold θ. To simulate the effect of the minimal read threshold, we downsampled the read counts of the proof-of-concept dataset by a factor r_c and applied to it the minimal read threshold θ resulting in a number of variants above threshold $n_{simul}(\theta, r_c) < n_{poc}$. The estimated final number of variants is therefore $n_{output}(\theta, r_c, r_t) = n_{simul}(\theta, r_c) \times r_c \times r_t$. Figure 3.5 shows the estimation of the number of variants $n_{output}(\theta, r_c, r_t)$ as a function of the minimal read count threshold θ and of the total library size $n_{input}(r_c, r_t)$, as well as the experimental library size values.

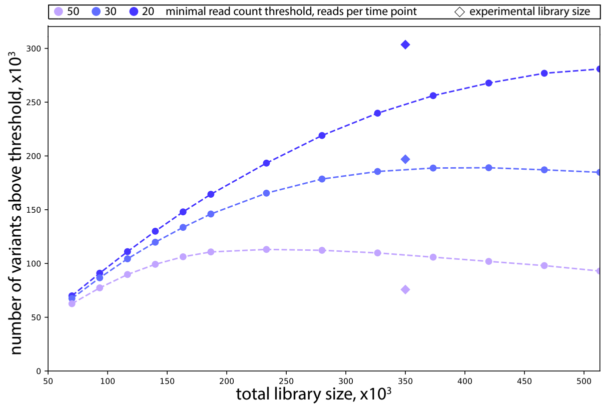


Figure 3.5: **Optimisation by adjustment of NGS loading.** The effect of the total library size on throughput (i.e. number of variants above read-count threshold) of uASPIre for the optimised sampling schedule is shown for different read-count thresholds. The experimental library size values are shown with diamond shape.

3.3.2.3 *RBS library design*

Initial efforts for training a convolutional neural network (CNN [143], described below) based on the proof-of-concept dataset resulted in a systematic underestimation of RBS strength, in particular for strong RBSs (see Figure 3.6). This is likely due to the library being skewed towards weak sequences as a result of the full randomisation of the 17 bases upstream of the Bxb1 start codon Figure 3.4. To overcome this, three libraries (High1 − 3) presumably enriched in moderate-to-strong RBSs are designed *in silico* based on the proof-of-concept dataset and added to a fully randomised library (N17). The three libraries are downsampled from three degenerate RBSs, which are constrained to the IUPAC notation. As a reminder, the IUPAC notation is a standard DNA (or RNA) representation where each character encodes a set of nucleotides. The single nucleotides {A, C, G, T, U} are represented as such, {W, S, M, K, R, Y} represent pairs of nucleotides, {B, D, H, V} triplets of nucleotides and N any nucleotide. Libraries High1 and High2 are designed using position probability matrices (PPMs), 2D matrices in which each element represents the proportion of times a nucleotide occurs at a given position in the sequence. To this end, RBSs from the proof-of-concept dataset were grouped into 10 linearly distributed bins according to a proxy for the normalised integral of their flipping profile (IFP_{trz}), for each of which a PPM was computed. Degenerate RBS sequences for High1 and High2 were designed with the goal to obtain IUPAC PPMs that most closely resemble (minimal mean-squared error) the PPMs of the highest and second highest bins, respectively. Finally, the creation of the library High3 involved the coupling of a prediction model and a genetic algorithm. To this end, we first investigated whether a deep learning-based model trained on the proof-of-principle dataset could achieve non-trivial predictive performance, despite the relatively small sample size, in order to meaningfully guide library design. For this purpose, a convolutional neural network (CNN) composed of one convolutional layer and two fully-connected layers, the last one's output being a scalar value, was used. The predictive performance of such a model was explored by means of 5-fold cross-validation on the proof-of-concept dataset, such that for each fold, 70% of the data is used for model fitting, 10% for model selection and the remainder 20% a held-out test set. The hyperparameters were selected with grid search on each validation set. As shown in 3.6, despite being outperformed by `SAPIENs` when trained on the final dataset, this smaller model achieves sufficient predictive power ($R^2 = 0.644$, MAE$= 0.055$) to guide library design. Therefore, we selected a final model by an additional 5-fold cross-validation, in this case using 80% of the data in each fold for model fitting and the remainder 20% for model selection. The best set of hyperparameters found in this manner was composed of a filter size of 5 and a number of filters of 128 for the first convolutional layer, 16 output elements for the first fully-connected layer, a weight decay of 0.005, the learning rate equal to 0.0001 and the batch size set to 512. Once this final model was selected and fitted, the RBS sequences from the three highest bins were randomly mutated for 200 iterations, with 1 or 2 mutations, keeping only the mutations that led to an increase in IFP_{trz} as measured by the predicted values of the CNN. The PPM of the pool of mutated sequences was calculated, a subsample of 20,000 sequences was randomly generated from this PPM and the predicted IFP_{trz} was computed for each generated sequence of the subsample using the trained CNN. Starting with a random degenerate RBS sequence, we mutated it iteratively one position at a time (random ordering of the positions), and kept the IUPAC nucleotide at the corresponding position that led to the smallest Kolmogorov-Smirnov (KS) distance between the predicted IFP_{trz} distribution of the subsample and the predicted IFP_{trz} distribution of the 1000 sequences generated from the new degenerate RBS sequence. This iterative process was continued until the relative decrease in KS distance was less than $\epsilon = 10^{-3}$ for three consecutive iterations. Figure 3.7 shows that the sequence design experiments were successful as the enriched libraries, indicated by High$1-3$, contain a larger proportion of medium-to-strong sequences compared to the fully-randomised library N17. We can also notice that High3 contains a larger percentage of strong RBSs than all other libraries, which corresponds to the original design goal.

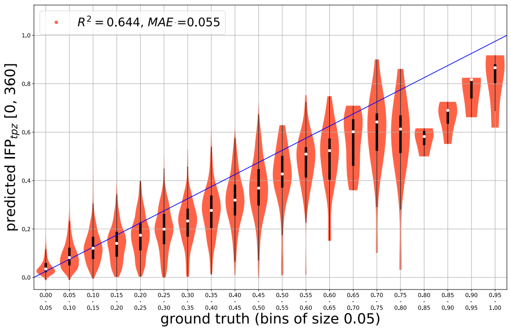


FIGURE 3.6: **Preliminary CNN predictions on the proof-of-concept dataset.** Initial predictions obtained with a convolutional neural network model (5-fold cross-validation) trained on the proof-of-concept RBS library. Coefficient of determination (R^2) and mean absolute error (MAE) are obtained based on predictions on held-out data.

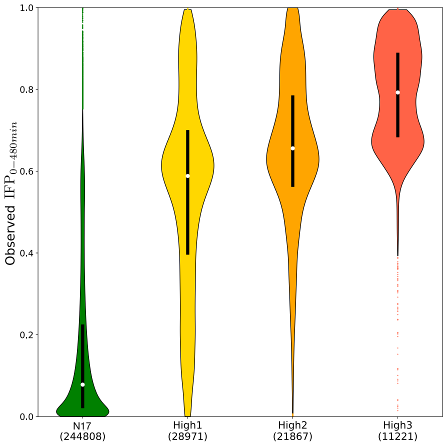


FIGURE 3.7: **Composition of the fully-randomised library and of the designed ones.**

3.4 definition of the label used by the machine learning models, in the *large* dataset

As briefly mentioned above, the flipping profiles were summarised into a scalar value that is representative of the RBS activity. In order to do so, it was necessary to find a summary statistic that is well correlated to the standard measurements of gene expression. To this end, 31 **internal-standard RBSs** were defined to span the full range of gene expression as measured by the preliminary proxy IFP$_{trz}$. The preliminary label IFP$_{trz}$ was compared to the GFP measurements using different curve fits to choose the best one for the machine learning model, as presented Section 3.4.1, and the chosen label was further refined to reduce the influence of noise in the measurements, Section 3.4.2. We also showed that labels that come from different biological replicate datasets can be successfully mapped to each other, as explained Section 3.4.3, indicating that training on one replicate is enough to predict on other potential replicates, conditioning on a pre-normalisation step.

3.4.1 *The label can be used as proxy for RBS activity*

In order to convert Bxb1-catalysed discriminator flipping into cellular Bxb1 concentrations, we compared the recorded cellular fluorescence profiles for the 31 internal-standard RBSs with their corresponding flipping profiles as collected by NGS (see Figure 3.8). To this end, we sought to i) establish a combination of summary statistics that exhibit a high degree of correlation between the two measured quantities across the entire range of RBS strengths, ii) identify the best (potentially non-linear) fit between the two summary statistics, and iii) ensure that a high degree of diversity is maintained for the representation of the discriminator flipping across the entire set of sequences in the dataset.

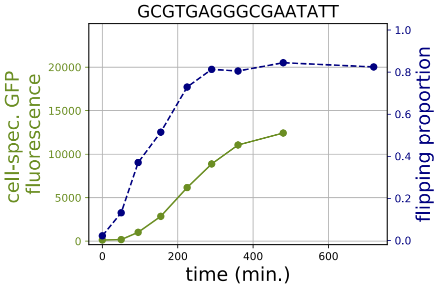


Figure 3.8: **Comparison of the cell-specific GFP fluorescence kinetic profile (green) with the flipping kinetic profile (blue) for one internal-strandard RBS sequence.**

We used integral-based (i.e. area under the curve) summary statistics for the flipping profiles and slope-based representations (i.e. slope of the linear fit) for the fluorescence profiles (Figure 3.9(a) and (b)). For the flipping statistics we also quantified the diversity of each representation by estimating the differential entropy [144] of its probability density (Figure 3.9(c)). In order to obtain robust GFP measurements, the fluorescence profiles of each of the 31 standard-inner RBSs were measured three times. Before calculating summary statistics of the fluorescence profiles, we preprocessed the fluorescence profiles as follows. The preprocessing step consisted of imputing the fluorescence values a) for time points of profiles that are missing or b) at time point 0 for profiles that showed a fluorescence at 0 higher than the one at the following time point (50 minutes). To do so we fitted each of the biological replicate curves with a generalised logistic function ($x \to \frac{K}{(1+Q\exp(-Bx))^{1/v}}$) using the values available and imputed the missing values or the inflated values (at 0) with the value of the fitted function at this time point. For each type of summary statistic, we additionally treated the time ranges over which both the fluorescence and flipping summaries are computed as additional hyperparameters to be optimised. To compute the summary statistics, the intervals of interest for the flipping profiles are $[0, 360]$, $[0, 480]$ and $[0, 720]$ (minutes). Similarly, the fluorescence profiles intervals of interest are $[0, 225]$, $[0, 290]$, $[0, 360]$ and $[0, 480]$ (minutes). Integral-based summary statistics for the flipping profiles are estimated using the trapezoidal rule. Slope-based representations for the fluorescence profiles are calculated by fitting a linear regression to the datapoints within the interval of interest (boundaries included) of the three biological replicates together. The slope of the fit serves as slope-based representation and the standard deviation of the slope is used to estimate the deviation around the estimated slope. Once the representations for both profiles are estimated, fits were evaluated from representatives of the flipping profiles to those of the fluorescence profiles. To do so linear ($x \to Ax + B$), log-linear ($x \to A \log x + B$) and general logistic fits ($x \to A + \frac{K-A}{1+Q\exp(-Bx)}$) were used. We quantified the quality of each pair of summary statistics using the resulting coefficient of determination R^2 of the fit as evaluated using leave-one-out cross-validation on the pool of 31 internal standard RBSs in order to compensate for potential effects of overfitting in the analysis. The leave-one-out cross-validation consisted of learning the free parameters of the fitted functions on all but one internal-standard RBSs (here $30 = 31 - 1$ datapoints) and predicting the output for the last datapoint that was not used for fitting. This step was performed 31 times (one time per inner-standard RBSs). The coefficient of determination was then calculated between the 31 inner-standard RBS representations of the fluorescence profiles and the 31 predictions. Moreover, the standard deviation of each summary statistic for fluorescence was computed for all internal standard RBSs relying on the three biological replicates.

As shown Figure 3.9, we observe that the pairs of summary statistics correlate strongly, indicating that the normalised integral of the flipping profile is strongly indicative of the prevailing cellular GFP concentration. We observe that the coefficients of determination between pairs of summary statistics after the leave-one-out cross-validation are better with a logistic fit (Figure 3.9(a)). As a compromise between a high correlation with GFP concentration with a logistic fit and a high diversity of the label (Figure 3.9(c)), we chose integral of the flipping profile between 0 and 480 minutes after induction as label for the further data analysis and machine learning steps. The curve representing the mapping between the two summary statistics, the slope of the GFP profile between 0 and 290 minutes and the integral of the flipping profiles between 0 and 480 is shown Figure 3.9(b) in the center.

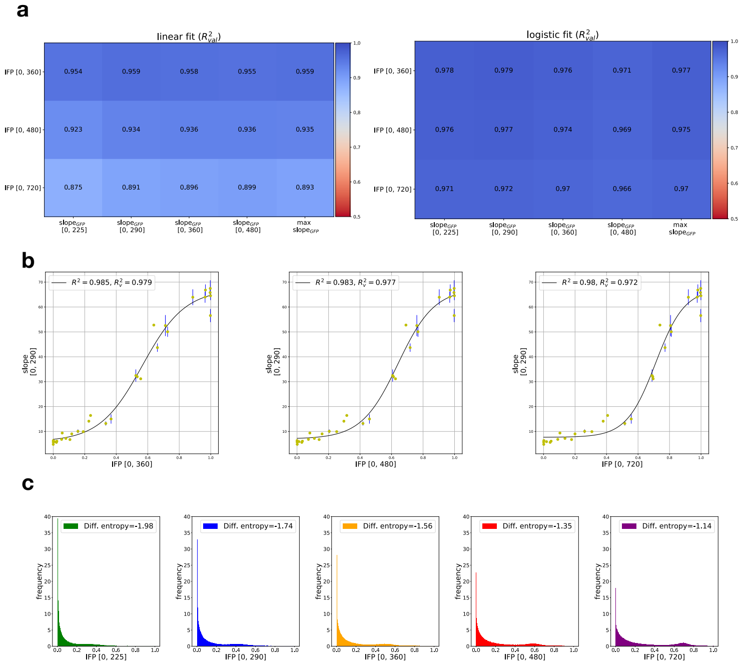


FIGURE 3.9: **Identification of optimal parameters to correlate Bxb1-mediated recombination with cellular GFP levels.** (a) Coefficient of determination after leave-one-out cross validation (loo-CV) (R^2_{val}) between different slope- and integral-based summary statistics for cell-specific fluorescence and the flipping profiles of the 31 internal-standard RBSs using linear and logistic fits. Note that slope-based summary statistics for the flipping profiles failed to deliver robust fits (R^2_{val} consistently below 0.5) and were therefore not included in this figure. (b) Selected logistic fits involving the integral of the flipping profile (IFP) for different time spans and the slope of the cell-specific fluorescence curve between 0 and 290 min after induction. The standard deviation of three biological replicates for the fluorescence profiles is indicated by vertical error bars and coefficients of determination without (R^2) and with loo-CV (R^2_v) are displayed. (c) IFP distribution across the entire larger RBS library for different integration intervals. The differential entropies of the respective IFP probability densities are indicated. IFP $[0, t]$: normalised integral of the flipping profile between 0 and t minutes after induction; slope$_{GFP}$ $[0, t]$: slope of the cell-specific fluorescence curve between 0 and t minutes after induction; max slope$_{GFP}$: maximum slope (minimum three timepoints) of the cell-specific fluorescence curve.

3.4.2 *Creation of the label, as normalised integral of the flipping profile*

A majority of flipping profiles present moderate irregularities, such as measurements at later time points might yield lower values than earlier ones. As the flipping profiles represent cumulative distributions of the proportion of flipped discriminator reads, this type of irregularities should ideally not be present. However, it is hardly possible to perfectly control all factors of the experimental protocol and this creates irreducible random noise. In order to smooth out the noise, it is possible to fit the profiles using a generalised logistic function. We used the following generalised logistic fit as it is a non-decreasing function and shows the same S-shape as the one observed on many of the profiles (see Figure 3.3(d)):

$$f(t; A,D,E,t_0,v) = \frac{A}{\left(E + e^{-D(t-t_0)}\right)^{\frac{1}{v}}} \tag{3.1}$$

The fit summarises the relationship between the time and the proportion of flipped discriminators per RBS. As the profiles are numerous and diverse, $> 10^5$ observations, it is not possible to fit automatically all the profiles with the default parameters as it leads to several errors and unfitted profiles. To this end, we implemented a preprocessing algorithm to find automatically better initialisation parameters, fit all the profiles in parallel and evaluate the normalised fitted flipping profile integrals. The algorithm can be described as follows:

Algorithm 9 Pseudocode to fit the profiles and compute the normalised integral of the flipping profiles $\text{IFP}_{0-480min}$.

Input: $\boldsymbol{Y}$ profiles array, $t_{start} = 0$ minutes beginning of integration interval, $t_{end} = 480$ minutes end of integration interval, $\boldsymbol{R}$ total reads array (same sample ordering as $\boldsymbol{Y}$), $\theta = 20$ minimal read count threshold (see Section 3.6.1), $b_{\mathcal{C}} = 10$ number of bins, $n_{\mathcal{K}} = 50$ number of clusters.
Output: $\boldsymbol{I}$ normalised integrals array.

```
Crop the flipping profiles $\boldsymbol{Y}$ outside of the integration interval $[t_{start}, t_{end}]$.
Select all the flipping profiles that verify the minimal read count threshold in
  the integration interval $\theta$. Let $\mathcal{S}_\theta$ be the set of profiles to be fitted.
Remove from $\mathcal{S}_\theta$ the flipping profiles that are constant and equal to 0 or to 1.
Evaluate the integral values of every profile in $\mathcal{S}_\theta$ according to the trapezoidal
  rule.
Rank the samples according to their integral values and cluster them by bins $\mathcal{B}_i$
  of size $1/b_{\mathcal{C}}$, the bins boundaries are $[1/b_{\mathcal{C}} \times (i-1), 1/b_{\mathcal{C}} \times i]$, for $i \in [\![1, b_{\mathcal{C}}]\!]$.
for $b$ from $1$ to $b_{\mathcal{C}}$ do:
    Cluster the profiles in bin $\mathcal{B}_b$ with $k$-means, among $n_{\mathcal{K}}$ clusters.
    Compute the centroid profile $\mathbf{C}_{.,b,k}$ in for each cluster $\mathcal{C}_{b,k}$, for $k \in [\![1, n_{\mathcal{K}}]\!]$.
    for $k$ from $1$ to $n_{\mathcal{K}}$ do:
        Fit the centroid profile $\mathbf{C}_{.,b,k}$. Save the parameters of the fit $\mathbf{P}^{init}_{.,b,k}$.
    end for
    for $k$ from $1$ to $n_{\mathcal{K}}$ do:                ▷ Step done in parallel on 32 CPU cores.
        Initialise the fitting functions for the cluster $\mathcal{C}_{b,k}$ with $\mathbf{P}^{init}_{.,b,k}$.
        Fit all profiles in $\mathcal{C}_{b,k}$ and save the new parameters $\mathbf{P}_{:,b,k}$.
    end for
end for
Fit manually the profiles that are constant and equal to 0 or 1 (trivial fit).
Compute the normalised integrals of all the fitted flipping profiles $\boldsymbol{I}$.      ▷ Step
  done in parallel on 32 CPU cores.
Return $\boldsymbol{I}$
```

An example of such fit is provided Figure 3.10. This algorithm allowed us to obtain the final labels $\text{IFP}_{0-480min}$ for each RBS sequence in the *large* dataset.

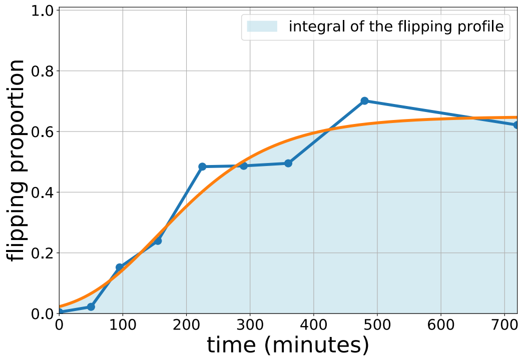


FIGURE 3.10: **Example of logistic fit (yellow) of one flipping profile (blue) and of the area that is used to calculate the label (light blue)**

3.4.3 *Normalisation of the label between biological replicates*

In order to facilitate the comparison of biological replicates we capitalised on the 31 internal-standard RBSs. These serve as internal references spanning a large range of RBS activities and allow to compensate for potential batch effects and other systematic biases between replicates. Formally, for each of the 31 internal-standard RBSs, we denote by x and y the measured normalised integral of the flipping profile (IFP) for the biological replicate to be normalised and the reference replicate, respectively. We fit either a polynomial function of degree two, $f : [0, 1] \to \mathbb{R}$ with $f(x) = I + Ax + Bx^2$, or its inverse $f(x) = g^{-1}(x)$ with $g(z) = I + Az + Bz^2$, such that the mean squared error between $f(x)$ and y is minimised across the 31 measurement pairs. Moreover, we impose the following constraints on the parameters of f: first, RBSs that show no activity in one replicate should remain inactive in the other replicates ($f(0) = 0$). Second, RBSs whose discriminators are entirely flipped before induction in one replicate should exhibit that behaviour in the other replicates ($f(1) = 1$). Third, the ranking of RBSs according to their strength should be preserved across replicates (f is monotonically non-decreasing in $[0, 1]$). It should be noted that, empirically, these assumptions appear to hold across the three biological replicates in this study. Imposing the first two constraints above reduces the number of free parameters of the polynomial function from three to one, resulting in the following family of functions, parametrised by A:

$$f(x) = Ax + (1 - A)x^2 \tag{3.2}$$

Moreover, the third constraint translates into the following bounds on the set of allowed values for the free parameter A: $0 \leq A \leq 2$. This procedure was carried out for each pair of biological replicates. The quality of the resulting fits was then evaluated on the full datasets, excluding the 31 internal-standard RBSs that were used to optimise A.

Figure 3.11 shows pairwise comparisons between the replicate datasets, before (a) and after (b) the normalisation. We observe that the fit learned on the 31 internal-standard RBSs allows to correct for batch effects almost perfectly in all cases for the remaining sequences.

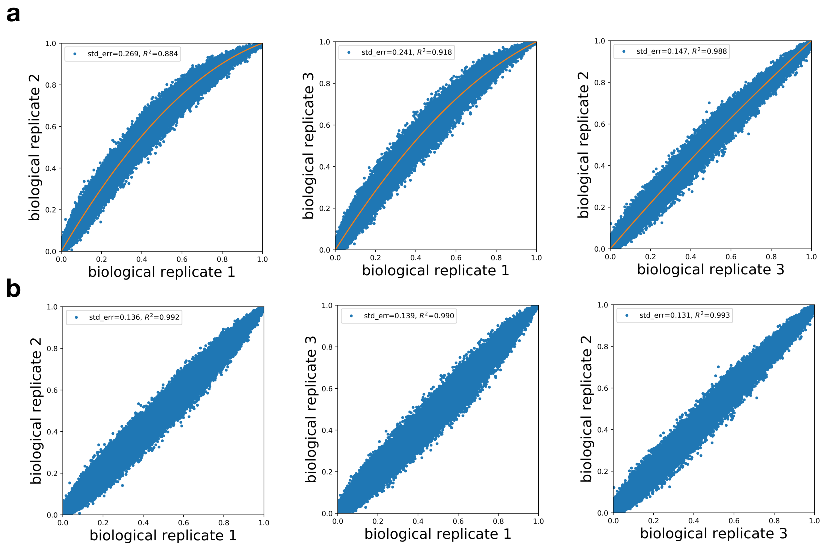


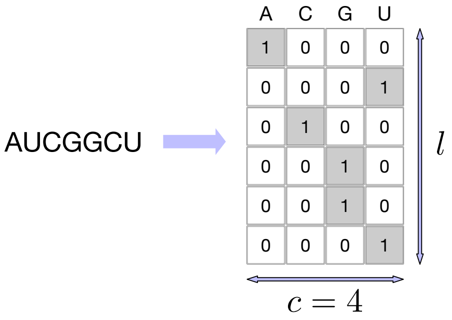
FIGURE 3.11: **Biological replicates.** A total of three independent biological replicates (i.e. individual shake flask cultivations) of the *large* dataset were subjected to the uASPIre workflow. Relying on the spiked-in 31 internal-standard RBSs, a normalisation curve was constructed and used to normalise the $\text{IFP}_{0-480min}$ values between replicates. (a) Comparison between IFP of independent biological replicates before normalisation. The yellow curve shows the polynomial fit on the 31 internal-standard RBS. (b) Comparison between IFP of independent biological replicates after normalisation. Coefficient of determination R^2 and standard error std_err are calculated on the full dataset but the 31 reference sequences.

3.5 DEVELOPMENT OF THE NEURAL NETWORK SAPIENs AS RBS PREDICTOR

Once the label has been defined, we build a deep learning model to accurately predict the RBS strength and to have a measure of uncertainty of the prediction for future usage of the predictions of the model. The model developed is an ensemble of residual convolutional neural networks whose architecture is described Section 3.5.1. From this model, well-calibrated estimates are obtained as explained Section 3.5.2 by parametrising the label distribution as a mixture of beta distributions.

3.5.1 *Description of the machine learning model SAPIENs*

The input of the model is an RNA sequence $\boldsymbol{s} = (s_1, ..., s_l)$ for length l, where each element $s_i, i \in [\![1, l]\!]$, represents a base $\in \{A,C,G,U\}$ [138]. As the neural network requires a numerical input, the sequences are one-hot encoded into numerical arrays, with length l and $c = 4$ channels, one for each base: the A-channel, the C-channel, the G-channel and the U-channel, as illustrated Figure 3.12. We fit the flipping profile of each RBS with a generalised logistic function (Section 3.4.2), integrating the fitted kinetic curves between the time points at 0 and 480 minutes and normalising the integral value by dividing by 480 (minutes). The resulting normalised integral value (range between 0 and 1; IFP$_{0-480min}$) is used as a descriptor of RBS behaviour and is selected as an exemplary target for prediction since it exhibits high correlation with cellular GFP levels and a high diversity across the RBS libraries (Figure 3.9).



A	C	G	U
1	0	0	0
0	0	0	1
0	1	0	0
0	0	1	0
0	0	1	0
0	0	0	1

FIGURE 3.12: **One-hot encoding.** The genomic sequence AUCGGCU is one-hot encoded. The resulting array is a 2D matrix that has as many rows as the genomic sequence is long (l) and that has 4 channels (c). **A**s are represented by a one in the first column and zeros in the other columns, **C**s are represented by a one in the second column and a zero in the other three, **G**s are represented by a one in the third column and a zero elsewhere and **U**s are represented by a one in the fourth column and a zero in the other columns.

Initially, we define a set of preliminary candidate deep-learning architectures for a predictive model according to standard practices [138, 145, 146]. These include convolutional neural networks (CNNs) with and without residual blocks, as well as multilayer perceptrons. These architectures are assessed as part of the hyperparameter selection process, which indicate superior performance of the CNN with residual blocks (ResNet) [147, 148] for this particular application, resulting in a model with three residual blocks of two convolutional layers and two sets of two fully-connected layers. We apply three main variations to the ResNet model in order to improve predictive accuracy and additionally provide a measure for predictive uncertainty. First, we choose the negative log-likelihood, which is a proper scoring rule, as the training criterion to achieve better uncertainty estimates [149]. The predicted $\text{IFP}_{0-480min}$ is modelled using a beta distribution, as it provides a flexible distribution with support in the interval $[0,1]$. Second, the last two fully-connected layers in the network are modified to output two values instead of one, thereby allowing to independently parametrise the two shape parameters of the predictive beta distribution for each input sequence. Equivalently, as the first two moments of the beta distribution are functions of the shape parameters, we were able to retrieve the mean and the standard deviation of the predictive distribution for each input sequence. Third, we use an ensemble of $N = 2 \times 5$ ResNet models [149], each trained separately with a different random initialisation of network parameters, a random order of training sequences during stochastic gradient-based optimisation and different architecture and optimiser hyperparameters. This third variation helps increase predictive accuracy and capture epistemic uncertainty.

The final model, **S**equence-**A**ctivity **P**rediction **I**n **E**nsemble of **N**etwork**s** (SAPIENs), is an ensemble composed of five ResNet models with three residual blocks of two convolutional layers, composed of 64 filters of sizes 9 and 1 respectively, followed by two sets of two fully connected layers with 64 units each (weight decay parameter: 10^{-6}, learning rate: 0.01) and five ResNet models with three residual blocks of two convolutional layers, composed of 512 filters of sizes 10 and 1 respectively, followed by two sets of two fully connected layers with 64 units each (weight decay parameter: 10^{-6}, learning rate: 0.001). The architecture of an element of the ensemble is shown Figure 3.13. In all cases, we keep a held-out test set and split the remaining dataset into a training and a validation set while keeping the same proportion of strong RBSs as defined by the 15th percentile of the $\text{IFP}_{0-480min}$ distribution. We use batch-normalisation [150] followed by LeakyReLU activation functions [151] between each layer. For optimisation, we used the Adam optimiser [152]. The model is implemented in Keras with the Tensorflow [153] backend. All hyperparameters (number of filters and layers, filters sizes, number of units of the fully-connected layers, weight decay, learning rate, batch size) were selected with random search [154] on the basis of their performance on the validation set.

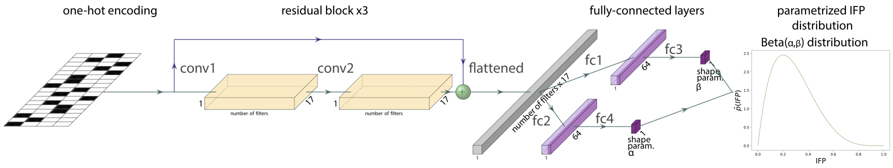


Figure 3.13: **Schematic architecture of a single ResNet.** One-hot encoded 17-bp RBS sequences are fed into three residual blocks, composed of two convolutional layers (conv1, conv2), and two sets of two fully-connected layers (fc1/fc3, fc2/fc4). Yellow and purple boxes represent the output of the convolutional and fully-connected layers, respectively. The grey box represents the output of the flattening operation. The model yields a probability distribution of the normalised $\text{IFP}_{0-480min}$ for each sequence from which the corresponding predicted $\text{IFP}_{0-480min}$ value (mean μ) and an uncertainty estimate (standard deviation σ) can be calculated. SAPIENs overall architecture is a combination of ten individually parametrised ResNets.

3.5.2 *Uncertainty estimation for the predicted $IFP_{0-480min}$ value*

3.5.2.1 *Modelling of the experimental measured values $IFP_{0-480min}$*

In our study, the ground truth values to be predicted by the model are the experimentally measured normalised integrals of the flipping profiles ($\text{IFP}_{0-480min}$) that quantify RBS strength. As described above, the $\text{IFP}_{0-480min}$ values are continuous and defined on the interval $[0, 1]$. Several approaches can be used to model such an output. In the context of neural networks, the most straight-forward way would be to use an appropriate activation function for the single output of the last fully-connected layer. Such activation function can be a **sigmoid function** ($f : \mathbb{R} \to [0, 1]; x \to \frac{1}{1+\exp(-x)}$) or a **soft-clipping function** ($f : \mathbb{R} \times \mathbb{R} \to [0, 1]; (x, v) \to \frac{1}{v} \log(\frac{1+\exp(vx)}{1+\exp(v(x-1))})$ with v a hyperparameter) and the model output would then be defined on $[0, 1]$, as the ground truth value. However, as explained above, we aim to estimate the uncertainty of the prediction together with the $\text{IFP}_{0-480min}$ values, which is generally not feasible with a single output value, as it is not sufficient in itself to estimate the uncertainty of each output value. Therefore, we choose to model each $\text{IFP}_{0-480min}$ value as a random variable whose distribution allows to get an estimate for the uncertainty of the prediction as well.

Towards this aim, we look for a distribution that satisfies the following desiderata: i) be supported on the interval $[0, 1]$, ii) be differentiable with respect to its parameters, iii) be flexible enough to model the mean and the standard deviation independently, iv) be flexible enough to take a large number of shapes, such as asymmetric, non-monotonic or skewed distributions and v) be parsimonious, i.e. depend on a small number of parameters. Commonly-used distributions that match these requirements are the **beta distribution**, the **logit-normal distribution** and the **truncated normal distribution**. We discard the logit-normal distribution as i) its mean and variance do not have a closed-form expression and have to be approximated differently, for example with a quasi Monte Carlo estimator and ii) preliminary experiments show that the model had difficulties in parametrising output distributions whose predictive mean ought to be close to the extremes of its support, i.e. near 0 or near 1. We also observe this difficulty in modelling very weak or very strong RBS activities when using the truncated normal distribution, albeit to a lesser extent. By contrast, the beta distribution does not suffer from any of those problems, and performs consistently well throughout the entire $[0, 1]$ range.

3.5.2.2 *Introduction to the beta distribution*

The beta probability density function (pdf) is defined as follows:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} \tag{3.3}$$

where α and β are the two shape parameters of the beta pdf and $B(\alpha, \beta) = \int_0^1 u^{\alpha-1}(1-u)^{\beta-1}du$ is the beta function, which is a normalisation constant that ensures that the total probability is 1. Roughly, the beta distribution can be interpreted as a generalisation of the Bernoulli distribution and the shape parameters can be seen as expected numbers of draws of the two classes when they are larger than one. For example, in our context, the normalised integral can be seen as representing a proportion of flipped discriminator reads if we were looking at only one time point. If the total number of flipped discriminator reads would be α and the total number of non-flipped discriminator reads would be β, then we could estimate the proportion of flipped reads to be equal $\frac{\alpha}{\alpha+\beta}$, which is exactly the mean of $x \to f(x; \alpha, \beta)$ for $x \in [0, 1]$, as $\int_0^1 \frac{u^{\alpha-1}(1-u)^{\beta-1}}{B(\alpha,\beta)} u du = \frac{\alpha}{\alpha+\beta}$.

3.5.2.3 *Negative log-likelihood as a proper scoring rule*

The first step towards getting valid uncertainty estimates is to choose a proper scoring rule as training criterion [149]. A scoring rule is a function that assigns a numerical score to a predictive distribution $p_{\mathbf{W}^*}(y|\boldsymbol{X})$ (here the weights $\mathbf{W}^*$ are the optimised ones) and rewards better calibrated predictive distributions, i.e. that are closer to the ground-truth conditional distribution of the targets given the samples $q(y|\boldsymbol{X})$. Let $S(p_{\mathbf{W}^*}, (y, \boldsymbol{X}))$ be a scoring rule that evaluates how well the predictive distribution matches the ground-truth conditional distribution $q(y|\boldsymbol{X})$ relative to the event $y|\boldsymbol{X}$. The expected scoring rule can be written $S(p_{\mathbf{W}^*}, q) = \int p_{\mathbf{W}^*}(y, \boldsymbol{X}) q(y, \boldsymbol{X}) dy d\boldsymbol{X}$. Most importantly, $S(p_{\mathbf{W}^*}, q)$ is a **proper scoring rule** if for all $p_{\mathbf{W}^*}$, $S(p_{\mathbf{W}^*}, q) \leq S(q, q)$, with equality if and only if $p_{\mathbf{W}^*}(y|\boldsymbol{X}) = q(y|\boldsymbol{X})$ for all sample-target pairs $(y, \boldsymbol{X})$ in the dataset D.

We choose the log-likelihood as proper scoring rule, or equivalently the negative log-likelihood as loss function. Let $\mathcal{B}$ be a batch with $n_{\mathcal{B}}$ samples, let $i \in [\![1, n_{\mathcal{B}}]\!]$

be a sample number in the batch $\mathcal{B}$, $\boldsymbol{X}_{:,i}$ a one-hot encoded genetic sequence, y_i its ground truth target value and $\mathbf{W}$ the weights of the model (we ignore the biases for simplicity of notation). The likelihood of having the ground truth target value y_i given $\boldsymbol{X}_{:,i}$ as input of the model is $p_{\mathbf{W}}(y_i|\boldsymbol{X}_{:,i})$, which depend on the parameters $\mathbf{W}$. We do not write the dependence on the hyperparameters as we can assume that they have been already selected on the validation set. The cost function can therefore be written as follows:

$$F(\mathbf{X}, y; \mathbf{W}) = -\frac{1}{n_{\mathcal{B}}} \sum_{i=1}^{n_{\mathcal{B}}} \log(p_{\mathbf{W}}(y_i, \boldsymbol{X}_{:,i})) + \lambda ||\mathbf{W}||_2^2 \tag{3.4}$$

where the left part of the right term is equal to the averaged negative log-likelihood over the samples in batch $\mathcal{B}$, and the rightmost term is the regularization term, which is the sum of the square of each weight. It is multiplied by the weight decay hyperparameter λ.

Minimizing the cost function amounts to: i) maximising the averaged log-likelihood and to ii) shrinking the weights in order to prevent the model from overfitting to the training data and to increase the ability of the model to generalise on an unseen test data. The objective is to find the optimal parameters that would ensure that for all elements in the batch, y_i is very likely to be the output of the model when $\boldsymbol{X}_{:,i}$ is the input.

As we choose to model the target variable with a beta probability density function, the likelihood can be written as $p_{\mathbf{W}}(y_i|\mathbf{X}_{:,i}) = \frac{y_i^{\alpha_{\mathbf{W}}(\mathbf{X}_{:,i})-1}(1-y_i)^{\beta_{\mathbf{W}}(\mathbf{X}_{:,i})-1}}{B(\alpha_{\mathbf{W}}(\mathbf{X}_{:,i}),\beta_{\mathbf{W}}(\mathbf{X}_{:,i}))}$.The functions $\alpha_{\mathbf{W}}$ and $\beta_{\mathbf{W}}$ in the likelihood are both neural network functions, whose weights $\mathbf{W}$ are learned during training. Furthermore, $\boldsymbol{X} \to (\alpha_{\mathbf{W}}(\boldsymbol{X}), \beta_{\mathbf{W}}(\boldsymbol{X}))$ defines a space of beta pdfs and establishes a correspondence between an input sample $\mathbf{X}_{:,i}$ and a beta distribution. As the space of beta pdfs is constrained by the parameters $\mathbf{W}$, it is in principle not possible for the model to find the beta distributions that would maximize $p_{\mathbf{W}}(y_i|\mathbf{X}_{:,i})$ for each sample $\mathbf{X}_{:,i}$. However, we can hope that the model finds a space of beta distributions that allows to make the output y_i very probable given $\mathbf{X}_{:,i}$ for all i.

3.5.2.4 *Mean and variance of the beta distribution*

The mean and the variance of the beta distribution are functions of the shape parameters. Let $i \in [\![1, n_{\mathcal{B}}]\!]$ be a sample number in a batch $\mathcal{B}$, $\mathbf{X}_{:,i}$ a one-hot encoded genetic sequence, y_i its ground truth target and $\mathbf{W}^*$ the weights of the model estimated after minimisation of the cost function. After optimisation, the mean of the beta pdf for sample $\mathbf{X}_{:,i}$ can be expressed as follows:

$$\mu_{\mathbf{W}^*}(\mathbf{X}_{:,i}) = \frac{\alpha_{\mathbf{W}^*}(\mathbf{X}_{:,i})}{\alpha_{\mathbf{W}^*}(\mathbf{X}_{:,i}) + \beta_{\mathbf{W}^*}(\mathbf{X}_{:,i})} \tag{3.5}$$

The corresponding variance can be written as:

$$\sigma^2_{\mathbf{W}^*}(\mathbf{X}_{:,i}) = \frac{\alpha_{\mathbf{W}^*}(\mathbf{X}_{:,i})\beta_{\mathbf{W}^*}(\mathbf{X}_{:,i})}{(\alpha_{\mathbf{W}^*}(\mathbf{X}_{:,i}) + \beta_{\mathbf{W}^*}(\mathbf{X}_{:,i}))^2(\alpha_{\mathbf{W}^*}(\mathbf{X}_{:,i}) + \beta_{\mathbf{W}^*}(\mathbf{X}_{:,i}) + 1)} \tag{3.6}$$

For each sample, it is therefore possible to access the predicted mean and variance of its beta distribution once we learn the shape parameters. The predicted mean $\mu_{\mathbf{W}^*}(\mathbf{X}_{:,i})$ of a sample i is the predicted target value and is an estimation for the ground truth target y_i. The predicted variance $\sigma^2_{\mathbf{W}^*}(\mathbf{X}_{:,i})$ contributes to the predictive uncertainty.

3.5.2.5 *Ensembling leads to a mixture of beta distributions*

The measured $\text{IFP}_{0-480min}$ for each RBS is modelled as a draw from a beta distribution. The mean and variance of this distribution estimated by the ResNet model (see above) correspond to the predicted $\text{IFP}_{0-480min}$ value and an indication of the aleatoric uncertainty of prediction (see Section 3.5.2.6), respectively. To complement this aleatoric estimate with an estimate of epistemic uncertainty (see Section 3.5.2.7), we first used an ensemble of $N = 5$ ResNet models with identical architecture and optimizer hyperparameters but different random parameter initialisation and ordering of the input sequences. The uncertainty estimate is therefore given by the standard deviation of the mixture of $N = 5$ beta distributions (Figure 3.14(a)-(b)). Furthermore, we extended this ensemble strategy at a later stage by also including M different configurations for the higher level hyperparameters, such as architecture and optimizer hyperparameters, with five ResNet models per configuration, resulting in a total of $N = M \times 5$ ResNet models in the ensemble. Finally, a number of configurations $M = 2$ (Figure 3.14(c)), therefore an ensemble 10 ResNets, was fixed as a trade-off between predictive performance and computational complexity. Figure 3.14(d) shows how the mixture of beta distributions for a held-out sequence approximates in average better the ground truth than the individual beta distribution.

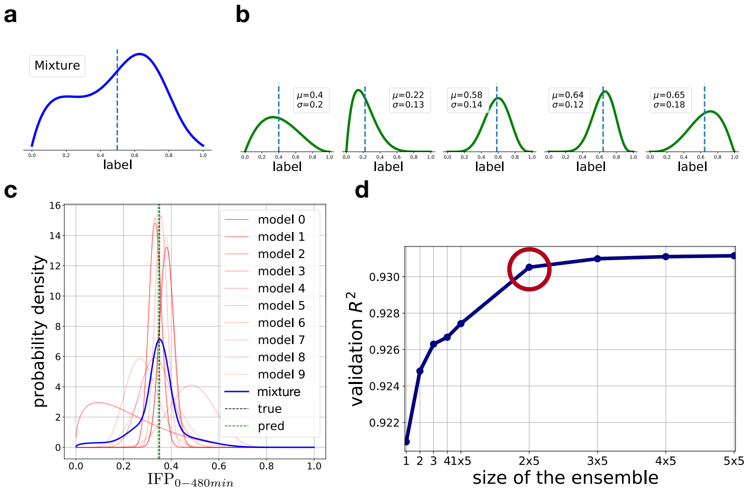


Figure 3.14: **Uncertainty estimation.** (a) Each single ResNet models the label IFP$_{0-480min}$ for each RBS sequence as a beta distribution whose mean (μ) and standard deviation (σ) can be computed. While the mean corresponds to the predicted IFP$_{0-480min}$ value, the standard deviation represents a measure of the uncertainty of the prediction. (b) Combining multiple ResNet models into an ensemble allows modeling the label IFP$_{0-480min}$ as a mixture of beta distributions, for each RBS sequence. (c) Example of the predicted mixture of beta distributions (in blue) of an ensemble of ten models for a given RBS sequence. The ensemble achieves a better prediction than the individual models as can be appreciated from its mean (green dashed line) which is in close proximity to the experimentally determined ground truth IFP$_{0-480min}$ value (black dashed line). (d) The validation coefficient of determination R^2 increases with the number of ensemble members. An ensemble size of 2×5 (red circle), corresponding to two hyperparameter configurations with five independently trained ResNets per configuration, was selected as a trade-off between accuracy and computational demand.

Let $m \in [\![1, n_{\mathcal{M}}]\!]$ be a model number, $i \in [\![1, n_{\mathcal{B}}]\!]$ be a sample number in a batch $\mathcal{B}$, $\mathbf{X}_{:,i}$ a one-hot encoded genetic sequence, y_i its ground truth target, $\mathbf{W}^*_m$ the optimised weights of model m, $\mu_{\mathbf{W}^*_m}(\mathbf{X}_{:,i})$ the mean of the beta pdf obtained with model m for input $\mathbf{X}_{:,i}$ and $\sigma^2_{\mathbf{W}^*_m}(\mathbf{X}_{:,i})$ the variance of the beta pdf obtained with model m for input $\mathbf{X}_{:,i}$. The output of the ensemble for each sample is a uniformly-weighted mixture of beta pdfs whose mixture density is as below:

$$p_{mixt}(y_i|\mathbf{X}_{:,i}) = \frac{1}{n_{\mathcal{M}}} \sum_{m=1}^{n_{\mathcal{M}}} p_{\mathbf{W}^*_m}(y_i|\mathbf{X}_{:,i}) \tag{3.7}$$

$$= \frac{1}{n_{\mathcal{M}}} \sum_{m=1}^{n_{\mathcal{M}}} \frac{y_i^{\alpha_{\mathbf{W}^*_m}(\mathbf{X}_{:,i})-1}(1-y_i)^{\beta_{\mathbf{W}^*_m}(\mathbf{X}_{:,i})-1}}{B(\alpha_{\mathbf{W}^*_m}(\mathbf{X}_{:,i}), \beta_{\mathbf{W}^*_m}(\mathbf{X}_{:,i}))} \tag{3.8}$$

The predicted mean and variance of the mixture would then write as:

$$\mu_{mixt}(\mathbf{X}_{:,i}) = \frac{1}{n_{\mathcal{M}}} \sum_{m=1}^{n_{\mathcal{M}}} \mu_{\mathbf{W}^*_m}(\mathbf{X}_{:,i}) \tag{3.9}$$

$$\sigma^2_{mixt}(\mathbf{X}_{:,i}) = \frac{1}{n_{\mathcal{M}}} \sum_{m=1}^{n_{\mathcal{M}}} (\sigma^2_{\mathbf{W}^*_m}(\mathbf{X}_{:,i}) + \mu^2_{\mathbf{W}^*}(\mathbf{X}_{:,i})) - \mu_{mixt}(\mathbf{X}_{:,i})^2 \tag{3.10}$$

The mixture mean and variance are functions of the means and variances of each model in the ensemble, and by definition, of the respective shape parameters. When using an ensemble, we make the hypothesis that the normalised integral of the flipping profile ($\text{IFP}_{0-480min}$) is a random variable that follows a mixture of beta distributions, and not a beta distribution as it would be the case for a single model. The mixture mean corresponds to the predicted target and the mixture variance contributes to the predictive uncertainty. It is possible to decompose this uncertainty in an aleatoric uncertainty and an epistemic uncertainty representing two sources of uncertainty of the prediction, as explained below.

3.5.2.6 *Estimation of the aleatoric uncertainty*

The aleatoric uncertainty refers to the intrinsic uncertainty of the data, which would remain if we repeated the same experiment in the same conditions several times. It can for example be directly linked to a noisy observation process that cannot be reduced or captured with more datapoints under the same experimental conditions. In order for a single model to express a tailored aleatoric uncertainty, it is possible to model the target as a random variable and predict the variance $\sigma^2_{\mathbf{W}^*}$ of the distribution of the target for every sequence input, in addition to the mean. In practice, the variance of the predictive distribution $p_{\mathbf{W}^*}(y|\mathbf{X})$ is a measure of the aleatoric uncertainty at each datapoint $\mathbf{X}$, assuming the weights $\mathbf{W}^*$ correspond to the true (unknown) weights.

3.5.2.7 *Estimation of the epistemic uncertainty*

The epistemic uncertainty or model uncertainty refers to sources of uncertainty that would be reduced if additional information were given, for example a larger sample size. An example of such uncertainty is that the mathematical model could neglect certain measurable effects. In practice, this uncertainty can be accounted for by taking into consideration the uncertainty of the optimised parameters $\mathbf{W}^*$ of the single model $p_{\mathbf{W}^*}(y|\mathbf{X})$. As a matter of fact, the parameters $\mathbf{W}^*$ of the predictive distribution $p_{\mathbf{W}^*}(y|\mathbf{X})$ are estimated and do not necessarily correspond to the unknown ground truth parameters. In order to capture the uncertainty of these parameters, it is possible to use an ensemble of models, i.e. to average the predictive distributions over several models that are either initialised differently, use different random batches or have different hyperparameters. If we were to consider one model to predict the target, it would be equivalent to assuming that the parameters $\mathbf{W}^*$ are equal to the ground truth ones and therefore we would not account for the model uncertainty in the variance of the predictive distribution. By contrast, if we could learn all possible models, we could marginalise (i.e. integrate over) the parameters $\mathbf{W}^*$ to estimate the ground truth conditional distribution, $q(y|\boldsymbol{X})$, as the average over all the predictive distributions corresponding to the infinitely many optimised models: $q(y|\boldsymbol{X}) = \int p_{\mathbf{W}^*}(y|\boldsymbol{X})r(\mathbf{W}^*|D)d\mathbf{W}^*$ where D is the dataset of interest and $r(\mathbf{W}^*|D)$ is the posterior probability of the parameters given the dataset. However, in practice this is unfeasible, and the ensembles are a mean to capture some uncertainty, by estimating the posterior probability of the parameters $\mathbf{W}^*$, $r(\mathbf{W}^*|D)$, as a sum of Dirac delta functions centered on the optimised parameters of the $n_{\mathcal{M}}$ models of the ensembles, such that the predictive distribution can be estimated by $p_{mixt}(y|\boldsymbol{X}) = \sum_{m=1}^{n_{\mathcal{M}}} \int p_{\mathbf{W}^*}(y|\boldsymbol{X})\delta(\mathbf{W}^* - \mathbf{W}_m^*)d\mathbf{W}^*$. Therefore, calculating the variance σ_{mixt}^2 of the predictive distribution $p_{mixt}(y|\boldsymbol{X})$ allows to capture some epistemic uncertainty, together with the aleatoric uncertainty.

3.6 EXPERIMENTAL DESIGN, EVALUATION AND BENCHMARKING OF SAPIENs

A set of experiments is performed to assess the performance of the model in different conditions. Section 3.6.1 explains why and how a minimum read count of 20 has been chosen to filter the dataset before the machine learning experiments. Sections 3.6.2 and 3.6.3 compare the performance of SAPIENs to various off-the-shelf machine learning models. Section 3.6.4 shows how the model behaves when the training sets and and test sets do not necessarily follow the same distribution. Section 3.6.5 shows the ability of the model to yield well-calibrated uncertainty estimates. Section 3.6.6 discusses the performance of the model on the biological replicates. Section 3.6.7 shows that the predicted $\text{IFP}_{0-480min}$ values correlate with the standard measures of gene expression, which conclude that the predictions can be used as proxy for RBS activity, therefore alleviating the need to sequence the entire space of the 4^{17} RBS sequences.

3.6.1 *Minimal read number threshold*

A minimal threshold for the number of NGS reads per RBS is determined as a quality control criterion for both training and test sets. Increasing this threshold is expected to trade off two opposite effects since it increases the average quality of the data leading to a decrease in the underlying aleatoric uncertainty but at the same time reduces the dataset size available for training, which generally lowers predictive performance. To this end, we first define six filtered datasets obtained by keeping only RBS sequences with at least 10, 15, 20, 30, 40 or 50 reads per sampling time point. Then, we randomly split each filtered dataset into training, validation and test sets as described above and made sure that for each split the high-quality training, validation and test sets were contained in the lower quality training, validation and test sets, respectively. Moreover, a test set is held out for the following prediction experiments. In order to identify an optimal lower read count threshold, we train a single ResNet model for 150 epochs. We randomise the search for hyperparameters [154] (see Section 3.5.1) used the same 150 sets of hyperparameters for each filtered training dataset and calculated the coefficient of determination on the validation set. Hence, the minimal threshold is effectively treated as a hyperparameter. This analysis indicates that a minimal read count threshold of 20 reads per time point is optimal for predictive performance, which saturates for lower thresholds despite the increase in overall dataset size (Figure 3.15(a)). We keep this training/validation/test split ("split0") for the following prediction experiments. Finally, we confirm that these conclusions are not an artifact of the random split of the original dataset by repeating this analysis using five different training, validation and test set splits (Figure 3.15(b)).

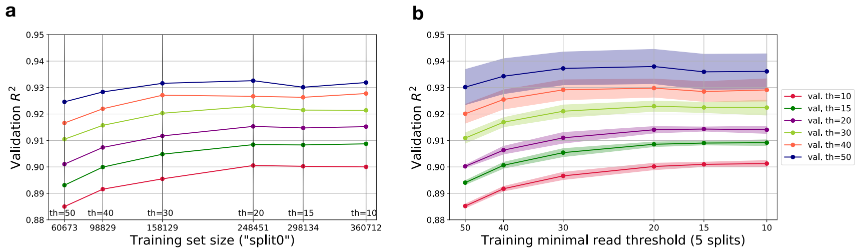


FIGURE 3.15: **Effect of the minimal read threshold on the predictive performance of one ResNet model.** (a) ResNet models were trained on different subsets of the training data corresponding to different minimal read count thresholds, and were evaluated in terms of the coefficient of determination R^2 in different subsets of the validation set also corresponding to different minimal read count thresholds. The data training/validation/test split ("split0") corresponds to the one we use to obtain the results in Figure 3.16 The experiment shown in (a) is repeated four times for different random splits to assess robustness. Data points in (b) represent the average of five random splits with two-standard-deviation intervals shown as shaded areas.

3.6.2 *Performance of SAPIENs on the entire dataset*

Using "Split0", we evaluate our model in more detail. Importantly, this implies that the test set had not been used in previous experiments in order to avoid overfitting. First, we used random search for selecting the best combination among 150 sets of hyperparameters on the validation set (see Section 3.5.1), let SAPIENs run for 300 epochs and used an early stopping criterion on the validation set to avoid overfitting by selecting the epoch with the best validation R^2. Figure 3.16 shows a comparison of $IFP_{0-480min}$ values as predicted by SAPIENs with the corresponding experimental values measured by uASPIre, for which we reach a coefficient of determination of $R^2 = 0.927$ and a mean absolute error (MAE) of 0.039. Moreover, the systematic inaccuracy in predicting strong RBSs is eliminated as a result of the addition of the three designed sub-libraries $High1 - 3$.

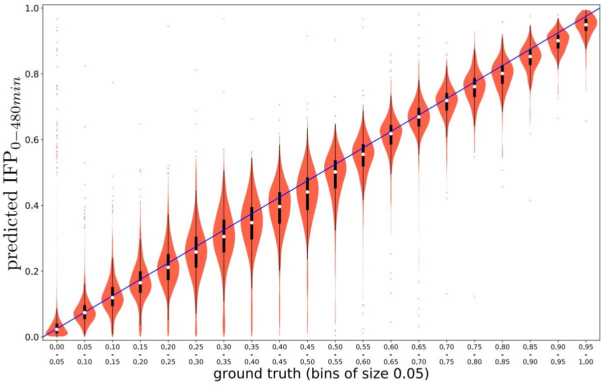


FIGURE 3.16: **Prediction performance of SAPIENs in the *large* dataset.** The x-axis represents experimentally measured IFP$_{0-480min}$ values in the test set and the y-axis represents the corresponding predicted values by SAPIENs. Sequences in the test set were binned (bin size: 0.05) according to measured IFP$_{0-480min}$. Violins comprise percentiles 0.5 to 99.5 of predicted values per bin with median and outliers represented as white circles and blue dots, respectively. Black bars contain the 25th to 75th percentiles.

3.6.3 *Performance of SAPIENs compared to off-the-shelf machine learning models as a function of the training set size*

We train SAPIENs and several classical linear and non-linear machine learning models on the same 248,451 RBS sequences chosen at random from the larger uASPIre dataset, issued from "split0". Hyperparameters are optimised exclusively on a validation set ($\sim$ 30,000 sequences) and afterwards all models are evaluated on a held-out test set ($\sim$ 30,000 sequences). The single ResNet and SAPIENs models are trained for a maximum of 150 epochs, using early stopping. A total of 100 randomly generated models with $1-3$ residual blocks are considered. Hyperparameters tuned for the other models are regularisation strength for ridge regression [23], number of neighbours K for k-nearest neighbours [155], number of trees for random forests [156], and maximum depth and learning rate for gradient tree boosting [157], the later also benefits from early stopping in the validation set. The impact of the training set size on predictive performance, Figure 3.17, is evaluated by training the different models on different smaller datasets, while ensuring that the training and validation sets are contained in the training and validation sets of higher sample size experiments (i.e. nested training and validation sets). Hyperparameters for all models are optimised independently for each training set size on the corresponding validation set.

As illustrated Figure 3.17, in the largest training set, the linear model ridge regression ($R^2 = 0.678$) is clearly outperformed by non-linear models k-nearest neighbours (k-NN, $R^2 = 0.738$), random forest ($R^2 = 0.835$) and gradient tree boosting (GTB, $R^2 = 0.893$), which highlights the importance of interactions between nucleotides in the RBS. Notably, SAPIENs outperforms all other approaches reaching an R^2 of 0.927 and MAE of 0.039. Importantly, except for the overall weakest-performing Ridge Regression, prediction accuracy increases with training set size for all models as reflected by rising prediction performance (R^2 and % within 2-fold error). While a general trend towards saturation is observed, no plateau is reached even for the largest training set of 248,451 sequences.

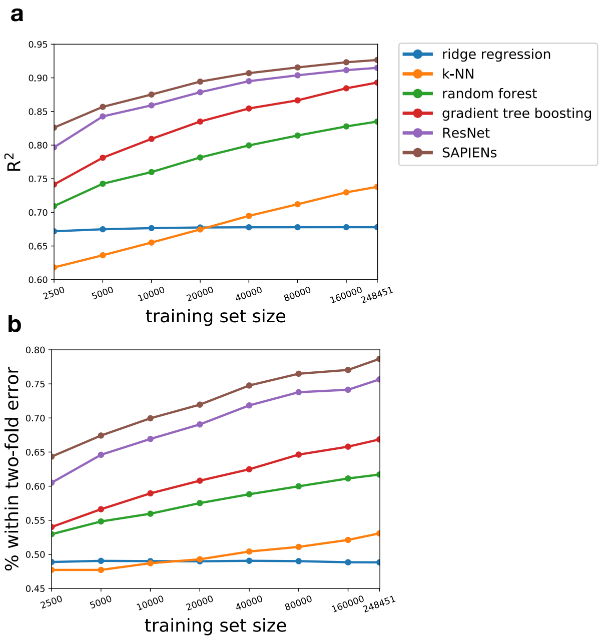


FIGURE 3.17: **Comparison between SAPIENs and state-of-the-art machine learning models.** SAPIENs, a ResNet model, gradient tree boosting, random forest, nearest neighbours and ridge regression are compared for different training set sizes, from 2500 to 248,451 samples. (a) The performance metrics used in the comparison is the coefficient of determination R^2. (b) The performance metrics used in the comparison is the percentage of predicted values within two-fold of the ground-truth values. The x-axis is represented in log scale.

3.6.4 *Effect of the designed sub-libraries on performance*

The effect of adding designed sub-libraries to increase the fraction of stronger RBSs in the bulk library is further analysed to evaluate a potential gain in predictive performance for the intermediate and strong sequences. To this end, we perform cross-analyses with the fully degenerate sub-library (N17) and the bulk library (N17+High1 − 3). We train on N17 and predict on unseen subsets of N17 and N17+High1 − 3, and train N17+High1 − 3 and predict on unseen subsets of N17 and N17+High1 − 3 (Figure 3.18(a)). In another set of analyses, we omit each of the enriched sub-libraries while training by moving them to the test sets and evaluate the corresponding effect (Figure 3.18(b)). In each case, we train a single ResNet model for 300 epochs for computational considerations and we use early stopping in the validation set. The hyperparameters are tuned independently for each dataset and selected from 150 random configurations in the corresponding validation set. All analyses are done with the same training and validation set sizes. Comparative analyses were performed with the same test set.

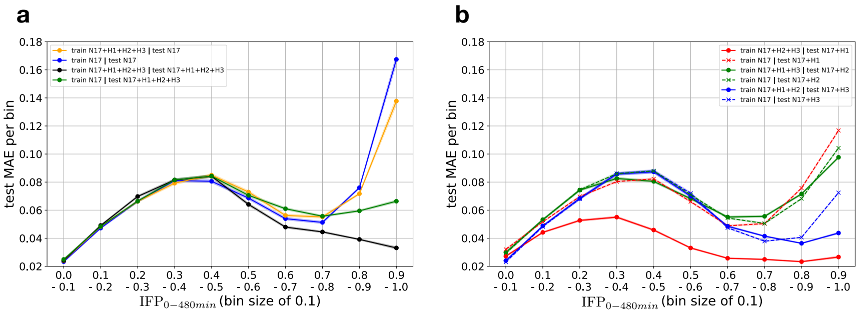


Figure 3.18: **Effect of designed sub-libraries on the prediction accuracy of the ResNet model.** (a/b) The mean absolute error (MAE) is evaluated for different bins of the experimentally determined IFP$_{0-480min}$ value and several combinations of training and test sets composed of the fully degenerate RBS library (N17) and the designed libraries (High1 − 3). 95% confidence intervals are indicated by shaded areas.

In Figure 3.18(a), we observe that compared to training on the fully degenerate library (N17), training on the sequences from the enriched library (N17+High1 − 3) leads to i) lower MAE for intermediate to strong ground truth IFP$_{0-480min}$ values when testing on sequences distributed according to the enriched library (N17+High1 − 3) (green vs black) and to ii) lower MAE for very high ground truth IFP values when testing on sequences distributed according to the fully degenerate library (N17) (blue vs yellow). Figure 3.18(b) complements this analysis to estimate the capacity of the model to generalise across High1, High2 and High3. This last figure shows a lower MAE for medium-to-strong or for strong sequences when training on the fully degenerate library to which two enriched libraries out of three were added.

3.6.5 *Calibration of the uncertainty estimate*

For any prediction model, it is common to evaluate the accuracy of the predicted targets with metrics such as MAE or RMSE. In a similar way, it is also key to be able to evaluate predicted variances, which serve as proxy for predictive uncertainty. As we do not know the ground truth variances, it is not possible to directly compare the predicted values to the ground truth values with common metrics. To this end, different approaches have been developed, one of them is building a reliability diagram [149]. The reliability diagram aims at establishing whether the predicted uncertainty is well-calibrated. It displays the percentage of ground truth values in the test set that fall into the τ%-confidence interval of their predicted beta probability density functions, for any $\tau \in [0{,}100]$. This number is then compared to the theoretical percentage of ground truth values that should fall in the τ%-confidence interval, which is exactly τ%. If these two percentages agree for any τ%-confidence interval, i.e. if the identity mapping holds, we say that the model is well-calibrated and the predictive uncertainty is meaningful. If the percentage of ground truth values in a given τ%-confidence interval is smaller than τ%, it means that the model is over-confident and tends to be certain about weak predictions. In the opposite case, if the number of ground truth values in a given τ%-confidence interval is larger than τ%, it means that the model is underconfident and that the variances tend to be too large.

In order to build the reliability diagram Figure 3.19(a), we use the estimated variances σ^2_{mixt} of the mixture of the beta probability density functions in order to calculate the boundaries of the τ%-confidence interval for each sample and each τ. As the resulting curve is perfectly aligned with the diagonal, these results indicate that our uncertainty estimates are very well-calibrated, indicating that the uncertainty of each predicted target value seems to be accounted for. This is confirmed by the fact that the mean absolute error is positively correlated with the predicted standard deviations (Figure 3.19(b)). Both these results suggest that the predicted standard deviations can be used as scores to evaluate the quality of each individual prediction.

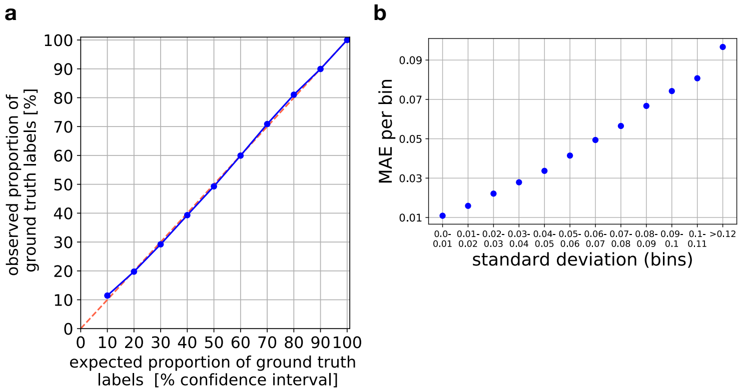


FIGURE 3.19: **Evaluation of the uncertainty estimates.** (a) The confidence intervals of the predicted probability distributions (horizontal axis) fully assess the uncertainty of the prediction values (vertical axis). (b) The mean absolute error (MAE), per bin of 0.01 of standard deviation, is almost linear as a function of the binarised standard deviation of the prediction.

3.6.6 *Generalisation across biological replicates*

We evaluate the ability of our model to generalise across biological replicates, that is, we would like to assess whether a model trained on measurements from one batch can accurately predict targets whose ground-truth values were measured on a different batch. To this end, we first collected data from three distinct biological replicates (batches). Next, each replicate dataset was randomly split into (stratified) training, validation and test subsets as previously done. Then, we removed any sequences from the test sets which did not pass the quality control criteria for all three replicates, allowing our results to be directly comparable across replicates. Finally, we considered test datasets where the targets are normalised to the training replicate (see Section 3.4.3) and where they are not.

After these preprocessing steps, we trained six instances of our model independently on the training subset of each replicate, normalised and not normalised, using the corresponding validation subset to select any hyperparameters. In practice, we ran the models for 150 epochs, used an early-stopping criterion on the validation set and performed random search among 150 sets of hyperparameters.

For each of the six models, we evaluate its predictive performance on: 1) test labels measured in the same batch, to assess the within-replicate performance; 2) test labels measured in the other two batches before normalisation; 3) test labels measured in the other two batches after normalisation, to assess cross-replicate performance before and after proper normalisation.

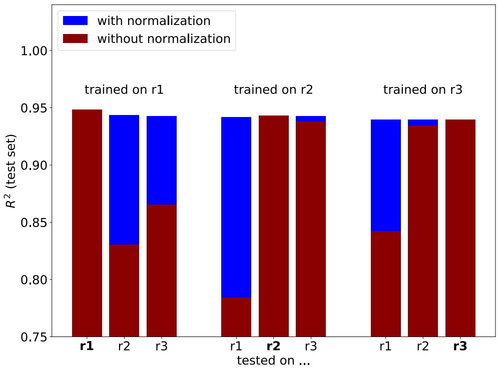


FIGURE 3.20: **Cross-training between biological replicates.** Six instances of the model are trained (indicated by 'trained on') and tested (indicated by 'tested on') independently on each biological replicate, with (blue) and without normalisation (purple). The normalisation consists of using the internal-standard RBSs, as shown Figure 3.11. The replicates are indicated by r1, r2 and r3. The performance is reported with the coefficient of determination R^2.

Figure 3.20, we observe that SAPIENs loses performance when testing cross replicates on non-normalised replicate labels, in particular when training on replicate r1 ($R^2 = 0.95$), and predicting on r2 ($R^2 = 0.83$) or r3 ($R^2 = 0.87$). However, normalising as shown Section 3.4.3 allows the model to be able to generalise between biological replicates. When training on replicate r1 and testing on normalised labels from replicates r2 or r3, we do not observe any significant decrease of the coefficient of determination ($R^2 = 0.94$ in both cases).

3.6.7 *Correlation between the predicted flipping integrals and cellular GFP concentrations*

A last experiment consisted of comparing the measured genetic expression of Bxb1, as obtained with cellular-specific GFP measurements, to the corresponding predicted flipping integrals, as given by the predicted $\mathrm{IFP}_{0-480min}$ values. As an intermediary step, Figure 3.21(a) shows that the ground truth $\mathrm{IFP}_{0-480min}$ values of the 31 internal-standard RBSs are very well predicted from SAPIENs. In Figure 3.21, we observe that cellular GFP concentrations can be reliably predicted from experimentally determined as well as from predicted $\mathrm{IFP}_{0-480min}$ values as shown for the 31 internal-standard RBSs. This indicates that our model reliably predicts cellular protein levels even for unseen sequences.

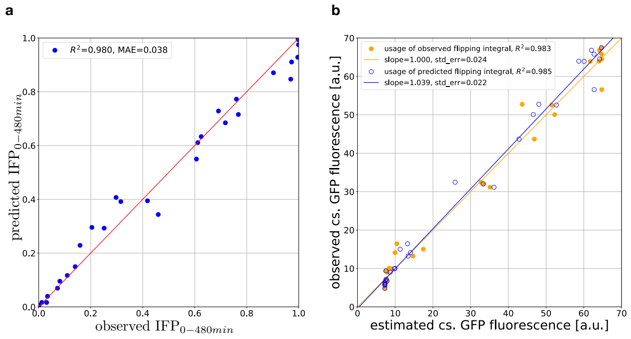


FIGURE 3.21: **Correlation of the ground truth GFP concentrations and $\mathrm{IFP}_{0-480min}$ values with SAPIENs predictions.** (a) The $\mathrm{IFP}_{0-480min}$ values for the 31 internal-standard RBSs as predicted by SAPIENs are highly correlated with the corresponding values experimentally determined by uASPIre, which were held out during training. (b) Correlations between cellular GFP concentrations, as measured by the slope of the cell-specific GFP signal between 0 and 290 minutes after induction, and estimates of GFP concentration calculated from experimentally determined as well as predicted flipping integral ($\mathrm{IFP}_{0-480min}$) values. The estimates of GFP concentration rely on the logistic fit parameters determined earlier (Figure 3.9(b) in the center).

3.7 influence of relevant sequence motifs and model interpretations

This last section presents different approaches that have been explored in order to gain some knowledge on the space of RBS sequence space, and the importance of bases and positions in these sequences. Section 3.7.1 presents descriptive statistics that point up positions of influence of known motifs of importance. Section 3.7.2 analyses the first layer of the network and feature attribution scores in order to highlight the bases and positions of importance according to the model SAPIENs when predicting RBS's activity. Section 3.7.3 shows typical changes that happen when mutating a strong sequence to a weak one and vice-versa.

3.7.1 *Analysis of relevant sequence motifs*

We analyse the fully degenerate sub-library (N17) in order to measure the impact of the position of known motifs of influence on the RBS activity, such as start-codons (AUG, GUG, UUG) or the consensus Shine-Dalgarno (SD) sequence (AGGAGG and subsequences). To this end, for each position, for each group of RBSs that present the motif of interest at the given position, we calculate simple statistics (median, interquartile ranges, 20/80 percentiles) on the target $\text{IFP}_{0-480min}$ of the sequences in the group. We exclude from these groups RBSs that contained at least one start codon other than the one at the position of interest.

Clearly, SD-like motifs exhibit a strong positive effect on translation, which is lost (or even slightly inverted) if the motif is too close to the translational start (Figure 3.22). Similarly, a positive effect was observed for additional in-frame AUG codons (Figure 3.23(a)) and, to a lesser extent, for GUG and UUG (Figure 3.23(b,c)). By contrast, out-of-frame start codons showed no globally consistent tendency but overall favored translation, in particular for positions -17 to -8. This is likely due to Gs in the start codons facilitating 16S-rRNA binding, which expectedly is most prevalent for GUG and difficult to disentangle from a genuine start codon effect.

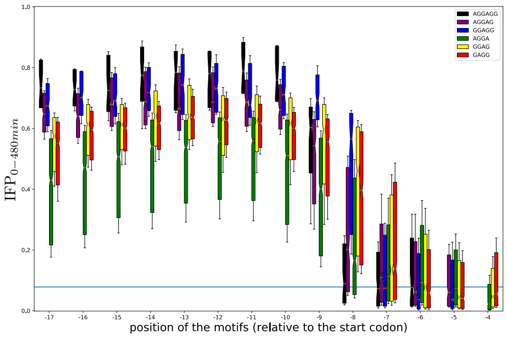


Figure 3.22: **Influence of subsequences of the Shine-Dalgarno consensus motif on the RBS strength.**

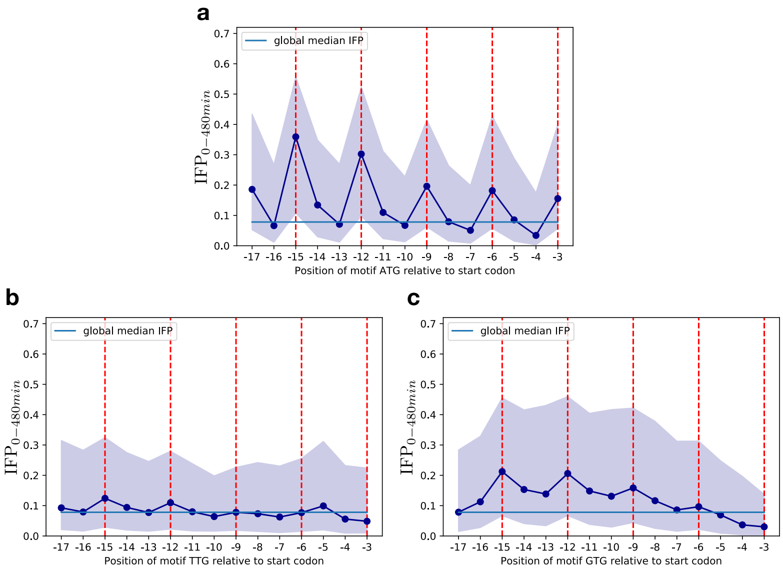


Figure 3.23: **Influence of AUG (a), UUG (b) and GUG (c) codons in the RBS sequence on RBS strength.** The dark blue line represents the median $\mathrm{IFP}_{0-480min}$ per position, for the relevant sequences. The shaded area corresponds to the $\mathrm{IFP}_{0-480min}$ values between the 20th and 80th percentiles. In-frame positions are highlighted in red.

3.7.2 *Model interpretation reveals the existence of positions and bases of influence*

We also analyse the filters of the first convolutional layer (excluding the first skip connection) of a ResNet model of the ensemble chosen at random. To this end, the effect of each filter is evaluated by calculating Pearson's correlation coefficient between the filter activations at each position and the flipping integral for all sequences in the test set. As a consequence, each filter is represented by a vector of correlations of size 17, which corresponds to the number of positions at which the filter influence is estimated. Finally, the filter representations are then clustered in twelve groups, with a complete linkage clustering method with hamming distance as the underlying metric between individual sequences, in order to group filters of similar influence. This analysis allows us to gain an understanding about the relative importance of RBS bases and positions. As illustrated Figure 3.24, we find that the first layer of the model has captured translation-promoting (A, G) and translation-reducing (C) effects of bases. Moreover, a positioning effect is observable: filters with large positive weight for Gs or negative weight for Us/Cs correlate positively with RBS activity when scanning upstream regions but negatively when closer to the translational start (centroids $1-4$). By contrast, filters promoting Us/Cs correlate negatively with RBS strength for most positions (centroid 5).

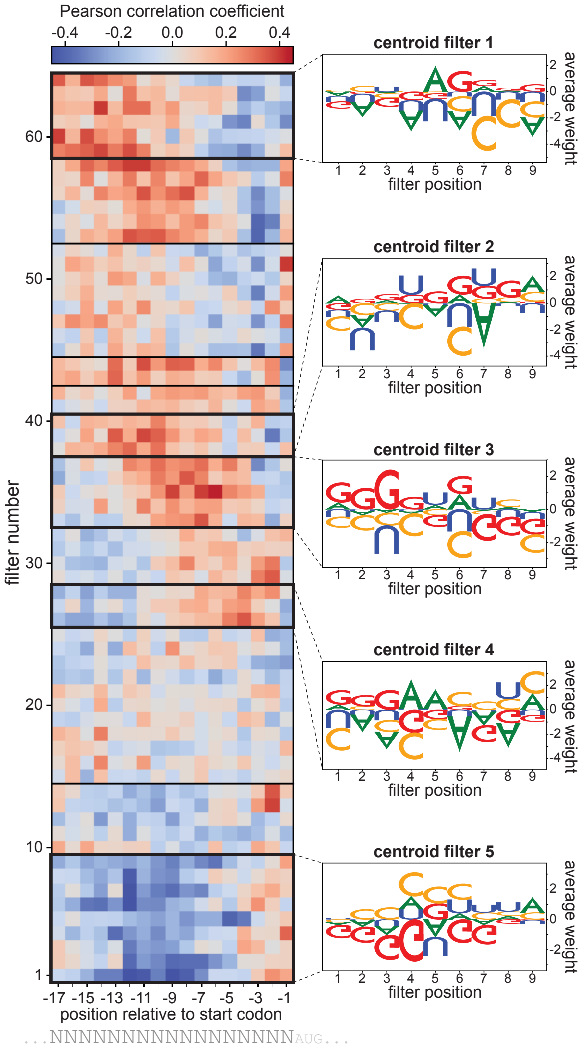


FIGURE 3.24: **SAPIENs first convolutional layer.** Each element in the heatmap represents the correlation between the values of the output of the first convolutional layer and the respective labels $\text{IFP}_{0-480min}$ for all RBS sequences in the test test. The rows of the heatmap are then clustered and centroids of five of the twelve clusters are displayed on the right.

A second method is introduced to further deepen our understanding of the model insights. To this end, we use the **integrated gradients** [158] method, which assigns attribution scores to each base and position by computing the linear path integral between the sequence of interest and a baseline sequence chosen a priori. The attribution scores measure the effect of individual bases on the predicted $\mathrm{IFP}_{0-480min}$, relative to a baseline. We apply the integrated gradients method to SAPIENs and choose a 'blank' one-hot encoded sequence as a neutral baseline (i.e. an all-zeros array). We first use a dimension reduction method, the t-distributed stochastic neighbour embedding (tSNE) method, to visualise how sequences behave in a low dimensional space (perplexity=12, early exaggeration=30) (Figure 3.25). This indicates a clear structure with strong and weak sequences separated almost linearly. We then perform a global analysis of the integrated gradients scores by averaging the attribution scores of all sequences in the test set, per base and per position. It allows to get a better understanding of the important positions and bases, which contribute either to a high RBS activity or to a low one (Figure 3.26). Substantiating the observation from Figure 3.24, Gs strongly promote translation while Cs appear to be consistently adverse. The translation-promoting effect for Gs is only observable if the distance from the start codon is at least 7 bp, while a neutral or even unfavourable effect prevails for other regions. However, no distinct SD-like motif appears because this global analysis only represents per-base and -position averages. Finally, in order to account for non-linearities between positions and to understand the drivers of very strong or very weak sequences, we selected the top 5% and the bottom 5% sequences in the test set after removing outlier sequences and clustered each pool with *k*-means according to their attribution score profiles into five clusters. The medoids of these five clusters are displayed for the strong (Figure 3.27(a)) and weak RBSs (Figure 3.27(b)). It reveals that SD-like motifs are the most impactful with positions ranging from -13 to -6 and invariance or slight preference for weakly pairing bases (A, U) outside the motif. Hence, our model successfully reconstructed SD-like patterns, notably without any prior knowledge about the process of translation.

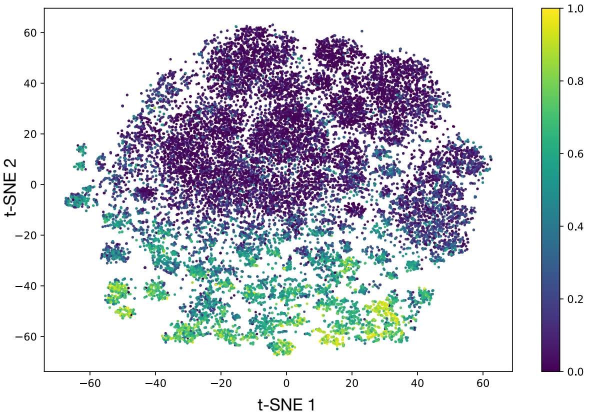


Figure 3.25: **Visualization of the integrated gradients scores of `SAPIENs` in a low-dimensional space.** T-distributed stochastic neighbour embedding (t-SNE) is applied to the integrated gradient scores of the RBS sequences from the test set. t-SNE dim1/2 are the two dimensions resulting from the t-SNE algorithm.

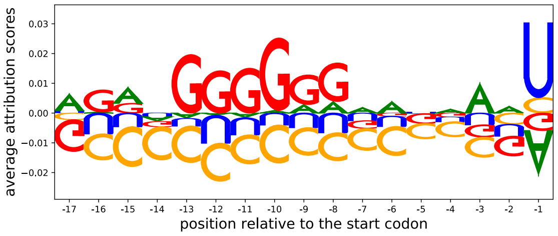


Figure 3.26: **Impact of bases and positions in the 5′-UTR on the RBS activity.** Using an all-zeros input as baseline, the average attribution score per base and position is displayed as determined for the sequences in the test set. The size of the letters corresponds to their importance score and their orientation to the direction of effect (i.e. upward/downward corresponding to a tendency to increase/decrease $\mathrm{IFP}_{0-480min}$).

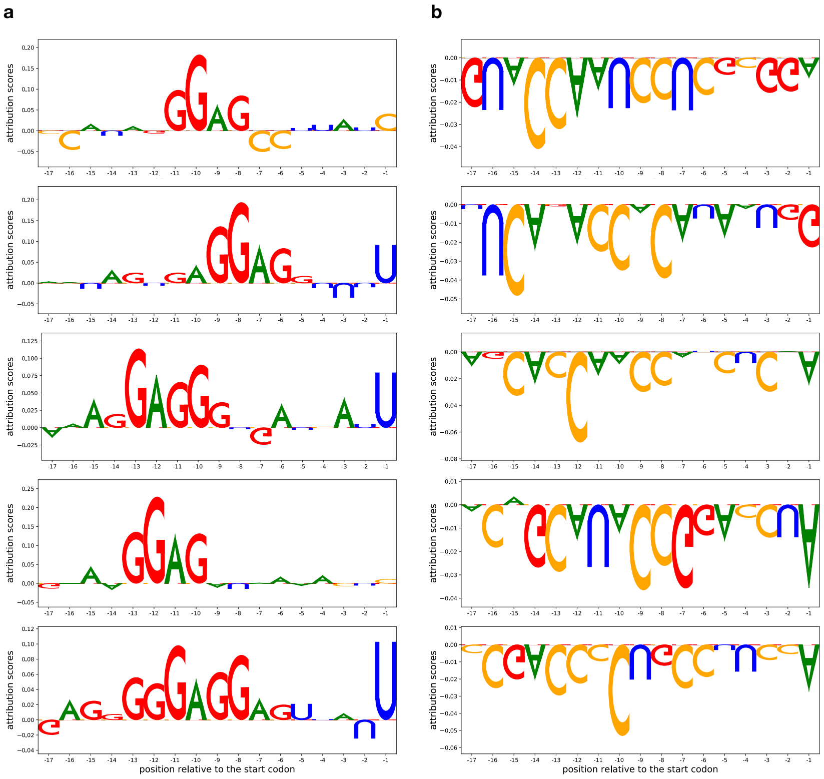


FIGURE 3.27: **Attribution of bases and positions of strong RBSs (a) and weak RBSs (b).** The strongest 5% (respectively weakest) of sequences in the test set were distributed into five clusters using *k*-means algorithm. For both (a) and (b), the displayed motifs are the five medoids of each cluster (i.e. the five individual sequences closest to the respective cluster centroid).

3.7.3 *In silico sequence design confirms previous findings*

For *in silico* evolution, we selected the weakest (respectively strongest) sequence in the test set and aimed to mutate it progressively to a sequence presenting a maximum (respectively minimum) attainable RBS activity as predicted by SAPIENs. To do so, we considered all sequences that could result from applying one or two mutations to the current sequence and kept the strongest (respectively weakest) one in each round until no candidate exhibited a change in predicted $\mathrm{IFP}_{0-480min}$ in the desired direction.

Figure 3.28 is the result of using SAPIENs to perform *in silico* evolution. Confirming our previous findings, the model systematically mutates U or C to A or G to form

SD-like motifs or create in-frame start codons upon increasing RBS strength (Figure 3.28(a)), whilst removing Gs and adding Cs when decreasing it (Figure 3.28(a)). Moreover, we observe that evolving a strong sequence ('gain of function') requires more steps than diminishing RBS activity ('loss of function') due to the sparsity of strong sequences within the search space.

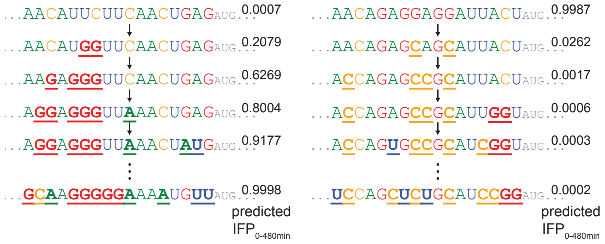

...AACAUUCUUCAACUGAGAUG... 0.0007
...AACAUGGUUCAACUGAGAUG... 0.2079
...AAGAGGGUUCAACUGAGAUG... 0.6269
...AGGAGGGUUAAACUGAGAUG... 0.8004
...AGGAGGGUUAAACUAUGAUG... 0.9177
...GCAAGGGGGAAAAUGUUAUG... 0.9998
predicted IFP0-480min
...AACAGAGGAGGAUUACUAUG... 0.9987
...AACAGAGCAGCAUUACUAUG... 0.0262
...ACCAGAGCCGCAUUACUAUG... 0.0017
...ACCAGAGCCGCAUUGGUAUG... 0.0006
...ACCAGUGCCGCAUCGGUAUG... 0.0003
...UCCAGCUCUGCAUCCGGAUG... 0.0002
predicted IFP0-480min

FIGURE 3.28: ***In silico* evolution of RBSs.** Starting from the sequence with the lowest (left) and highest (right) predicted $\text{IFP}_{0-480min}$ in the test set, pairwise mutations are greedily applied until no further increase (left) or decrease (right) in $\text{IFP}_{0-480min}$ is observed (total of 10 and 8 rounds for (left) and (right), respectively).

3.8 the "rbs predictor" as an easy-to-use webserver

We built a webserver, called the "RBS predictor", whose function is to predict the RBS activity of any RBS sequence that has been uploaded. The core of the webserver is the trained deep learning ensemble `SAPIENs`. As in the related preprint described in this chapter and entitled *Large-scale DNA-based phenotypic recording and deep learning enable highly accurate sequence-function mapping*, the prediction study focuses on ribosome binding sites of 17-pb DNA sequences, located right upstream of the start codon of the gene it is regulating, as shown Figure 3.29.

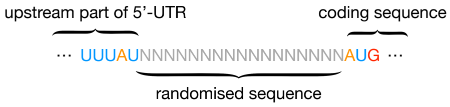
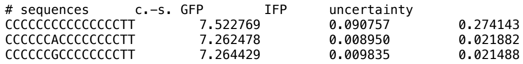


Figure 3.29: **Location of the 17-bp randomised RBS sequence.**

Once the user is connected to the webserver "RBS predictor", she has the possibility to:

- Select the regulatory sequence to analyse (RBS or promoter). The promoter option is not available yet and will be when the data will be shared from the biologists to the machine learning team.
- Upload a text file containing the sequences or write the sequences that the user wants to get predicted. In practice, it is possible to submit sequences identifier with the IUPAC code, up to 4^{10} sequences.
- Name to the three output files, containing respectively the sequences, the predictions and both the sequences and predictions. Each line corresponds to one sequence. The files are tab-separated.
- Submit the prediction job.



```
# sequences      c.-s. GFP        IFP      uncertainty
CCCCCCCCCCCCCCCTT        7.522769         0.090757         0.274143
CCCCCCACCCCCCCCTT        7.262478         0.008950         0.021882
CCCCCCGCCCCCCCCTT        7.264429         0.009835         0.021488
```

Figure 3.30: **Example of output of the webserver for three sequences.**

The output of the combined file, containing the sequences and the respective predictions is illustrated Figure 3.30. The first column indicates the sequences that have been analysed. The second column represents the predicted cell-specific GFP concentration, the third columns the predicted normalised integral of the flipping profiles ($\mathrm{IFP}_{0-480min}$) and the fourth column the uncertainty of the prediction of $\mathrm{IFP}_{0-480min}$. The predictions of the cell-specific GFP concentration are obtained by using the fitting curve between GFP and flipping integral shown Figure 3.31.

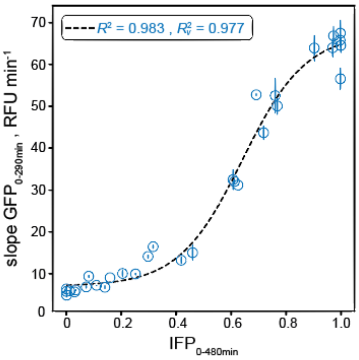


Figure 3.31: **Correlation between Bxb1-mediated recombination ($\mathrm{IFP}_{0-480min}$) with cellular GFP levels.**

The code runs one a single GPU. The webserver is free and its usage only requires the creation of a personal account. The user can access her previous and current jobs by going to the History page. Predicting 4^{10} sequences takes less than 25 minutes.

This webserver "RBS predictor" makes the predictions available from a general public and can help biologists, working with bacteria, design RBSs that would better fit their need. The webserver will be available under the link https://rbs-predictor.bs-ci05.ethz.ch/, upon publication.

4 CONCLUSION AND DISCUSSION

4.1 CONCLUSIONS ON THE ROLE OF INTERACTIONS IN GENETICS AND BIOENGINEERING

In this thesis, we introduced several models that allow taking into account interactions between features, either by (1) explicitly searching for them with the proposed significant pattern mining algorithms to retrieve novel associations that would remain undetectable when using univariate or additive models Section 2, or (2) by leveraging information interactions between features to increase prediction performance with the deep learning model `SAPIENs` Section 3. Both directions successfully find key interactions for their respective tasks, namely, feature selection or prediction. Most importantly, we show through these two representative applications that accounting for interactions is essential in genomics and synthetic biology to increase performance.

First, we have focused on the thesis on methods that show the importance of interactions in feature selection tasks. To this end, we have introduced `FACS` and `FastCMH`, the first algorithms that are capable of discovering statistically significant interactions of features and genomic regions exhibiting genetic heterogeneity, respectively, while correcting for a categorical covariate. With `FACS`, we have developed a method that is able to test all interactions of genomic variants for association with a phenotype of interest while correcting for covariates, without sacrificing statistical power or computational efficiency. We also present `FastCMH`, an algorithm similar to `FACS` that focuses on contiguous genomic regions instead of all interactions. Additionally, `FastCMH` composes representative vectors for interactions with a logical OR operation instead of a logical AND operation. Different types of biological interactions can therefore be considered. In the case of the OR operator, we assume that at least one minor allele among all loci in the set of genetic variants considered is enough to potentially induce biological modifications that can later alter the phenotype. This phenomenon corresponds to the setting of genetic heterogeneity where several genetic variants have a weak but similar effect on a same phenotype. In the case of the AND operator, all loci must display a minor allele to consider that the genes (or coding regions or corresponding proteins) interact and potentially induce a change to the phenotype. For both methods, our experiments on simulated data and COPDGene and/or *A. thaliana* datasets result in improved detection performance and superior computational power, compared to univariate baselines using the CMH test and a naive Bonferroni procedure for multiple testing correction. Additionally, we show that we obtain reduce the number of false discoveries due confounding compared to baselines that do not account for confounding factors such as FAIS-χ^2 (see Figure 2.8 and Table 2.1 for FACS, and Figure 2.11, Figure 2.12 and Table 2.3 for FastCMH).

In the second part of the thesis, we zoomed in a deep learning predictor that leverages non-linear interactions to obtain highly accurate predictive performance. We present a deep learning model SAPIENs that obtains a coefficient of determination $R^2 = 0.93$ on a held-out test set when predicting the activity of ribosome binding site sequences (see Figures 3.16 and 3.17), while providing well-calibrated uncertainty estimates for each predicted value (see Figure 3.19). These accurate predictions enable researchers to obtain a sequence to activity mapping over all the sequence space, which would be unfeasibly costly in practice with wet-lab experiments alone. However, for downstream tasks such as precise manipulation and reprogramming of cells, it would be of interest to know which sequences have predicted functions that can be trusted. While global performance metrics like R^2 or RMSE only give an average (over a large collection of held-out sequences) of the accuracy of the predictions, individual uncertainty estimates per predicted value crucially permit prioritising these sequences for which the model confidently predicts that they possess the sought biological function. To this end, we show that we are able to obtain well-calibrated uncertainty estimates, accounting for data and model uncertainties (see Figure 3.19). In this work, we also showed in a proof-of-concept experiment that a simple genetic algorithm, coupled with a preliminary deep-learning predictor, can design sequences whose activity is in the range of interest, here strong RBSs, even if the initial datasets is skewed towards weak RBSs (see Figure 3.7). Most importantly, we have shown that biologists and machine learning researchers can work together towards thoroughly optimised datasets that are nicely exploitable by machine learning models, and especially deep learning models. Combining emergent technologies that allow to create large labelled datasets in biology, and scalable and highly-performant deep learning models enable to address fundamental questions in regulatory circuits and genotype to phenotype relationship.

However, as will be discussed next, all the approaches developed in this thesis could be enhanced by either removing limitations and assumptions, integrating domain knowledge or solving different but related tasks.

4.2 DISCUSSION AND FUTURE WORK

4.2.1 *Future work in significant pattern mining applied to GWAS and other biological data types*

While the methods presented showed superior performance on binarised GWAS datasets with a categorical covariate to find either SNP interactions or regions of genetic heterogeneity, their applicability to other data formats requires further development. To this end, several directions for future work could be considered.

First, it would be possible to extend the notion of testability and of prunability to non-binarised GWAS datasets. FastCMH has been applied under a dominant encoding hypothesis, i.e., the SNP is encoded as a 1 if at least one of the two copies of the variant in a locus is altered, and as a 0 otherwise. However, this leads to a loss of information as the encodings for the presence of one minor allele and of two minor alleles collide. Therefore, being able to use the GWAS data without alteration, i.e. where each SNP is represented by a categorical covariate with three categories, could in principle lead to a better power. However, we would first need to explore how to properly define interactions between such features. Two options could be tried at first, one is to seek for a definition that allows to, when combining n SNPs, keep the number of categories (here 3) constant in the resulting representative vector. This constraint comes from the fact that we would like the minimum attainable p-values to be comparable across tests. However, as an alternative option, it would also be conceivable to find other solutions to relax this constraint. Another question could concern the underlying biological meaning of combining categorical features, such as how to meaningfully combine features that are encoded as a 2 with features that are encoded as a 0. Additional steps in this topic would require to use the CMH or χ^2 test for non-binary features [159], to prove the existence of a minimum p-value, which now depends on a multidimensional support due to the number of categories of the feature, and if the minimum attainable p-value is not monotonic, to efficiently derive a lower bound to the minimum attainable p-value. Another extension to the use of binarised features by FastCMH and FACS, aside categorical, would be to handle continuous features. It would be of great interest in many biological applications, for example, to find associations between sets of genes and a molecular phenotype in gene expression data. It is even more challenging to fit continuous features to Tarone's framework due to the non-discretness of the data and the resulting non-existence of a minimum attainable p-value. In this context naive discretisation is possible but not ideal. Early work has began exploring more comprehensive discretisation. [160] proposes to use the G-test [161] in order to discover statistically significant interactions of **continuous** features with a binary class label of interest. This paper is inspired by [162], which defines a notion of support for multiplicative interactions between continuous features as the average support of the interaction over an ensemble of binarised datasets obtained by taking all possible discretisation thresholds into account (with equal weights and independently for each features). [160]'s core contribution is to identify the potential of the concept proposed by [161] to apply significant pattern mining for real-valued features, thus going beyong simple frequent (real-valued) itemset mining. Moreover, starting from the definition of support in [162], [160] shows that it is possible to apply Tarone's framework in this new setting. [160] derives a monotonous minimum attainable p-value and the corresponding pruning criterion, for the G-test, which is a test statistic that can handle fractional counts such as those arising from averages over discretised datasets.

Second, besides extending the approaches proposed in this thesis by making them generally applicable to non-binary features, it would be possible to extend them to handle more complex covariates. An example of such complex covariates would be a structured categorical covariate, thus introducing a hierarchy between covariate categories. This is of particular importance in bacterial lineages, where the population structure is strong and can be naturally modelled in a hierarchical manner, for example, with species and genus. This would therefore allow applying SPM-inspired methods to bacterial GWAS, where the hierarchical relationship between bacteria is strongly present and observable [163, 164], as illustrated in Figure 4.1(a). To this end, if we were to correct for a single, non-hierarchal categorical covariate, a key question would be how to automatically detect the level of confounding for which we want to correct, having the possibility to define a covariate at different levels in the hierarchy, thus correcting, for example, for a fine-grained species classification or for a more coarse genus or family grouping structure. Then, other methods that take into account the population structure on its original hierarchical description could be considered, such as using a probabilistic framework or a linear-mixed model.

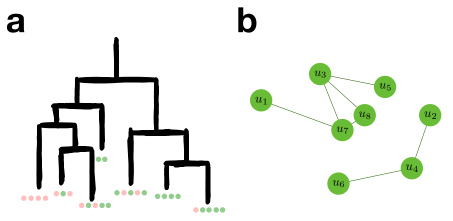


FIGURE 4.1: **(a) Example of population structure in bacteria population. (b) Example of protein-protein interaction network.**

Third, biological prior knowledge and more complex assumptions could be incorporated in the SPM-based algorithms. This could be done in two complementary ways, either by (1) changing the representations used for combining features or (2) by modifying the search space. In this paragraph, we will give an example of both cases. On the one hand, we can notice that both FACS and FastCMH define the representative vector of combinations of the features to be tested in different ways depending on the phenomenon that we are willing to study. FACS uses a product of the individual SNPs as the representative vector and FastCMH uses the OR operator between features (equivalent to taking the MAX operation sample per sample). However, a large number of meaningful ways to combine SNPs exist and could be further explored. A first example would be to account for the direction of effect of each individual SNP. This would enable to detect combinations of SNPs that jointly affect a phenotype, but with some being correlated and others anti-correlated to the phenotype. To this end, one possibility would be to create additional features that are simply obtained by inverting the original features, that is, substituting 1 with 0 and vice-versa. To integrate this idea into an SPM algorithm, we would need to change how patterns are enumerated to find an efficient filtering scheme in this context and limit the additional runtime complexity caused by doubling the number of features. On the other hand, there is an abundance of prior knowledge in the form of biological networks, where nodes correspond to features (e.g. genes) and edges to known (biological) interactions between features (e.g. protein-protein interactions), as illustrated Figure 4.1(b). This observation motivates studying how to exploit graph-structured prior knowledge when searching for multiplicative feature interactions significantly associated with a target of interest. A natural first step in this direction would be to investigate how to best use the graph to restrict the search space. A naive approach would consider to test all feature subsets that form connected subgraphs in the prior knowledge graph. While this sounds biologically plausible, exhaustive and conceptually simple, it would lead to a similar computational and statistical burden as in FACS, as the number of graph-based interactions would be closer to scale as the total number of subsets rather than as the total number of regions, unless the graph is extremely sparse. Potential solutions would be to explore alternative schemes to filter feature subsets based on the graph that lead to a greater reduction in the number of candidate interactions, while keeping those with high a priori (biological) plausibility.

Fourth, Tarone's framework as used in FACS and FastCMH is not able to account for correlations between test statistics which can lead to a loss of power when using Tarone's framework. To this end, [106] presents an adaptation of LAMP in the context of permutation testing, using the Westfall-Young algorithm. Permutation testing applied to FastCMH or FACS would lead to a gain in statistical power with the downside of a higher computational complexity. Such as extension could however be of interest when analysing small scale datasets harbouring weak signals.

Fifth, another direction of research towards gaining power by using adequate statistical tests, without however relying on permutation testing, is to use an unconditional test, such as the Barnard's exact test as proposed in [165]. As opposed to Fisher's exact test used in LAMP or to the CMH test used in FACS, which condition on all the margins of the contingency tables, i.e. the number of samples in each classes and the support of the pattern under study, the Barnard's exact test only conditions on the number of samples in each class and on a nuisance variable π. The nuisance variable π is the assumed probability to have a 0 or a 1 for each pattern, under the null hypothesis that both probabilities are equal to 0.5. The authors of [165] show that it is possible to derive a lower bound to the Barnard's test statistics in order to prune efficiently patterns and avoid enumerating all of them. In simulation experiments, they show that their method, which controls the FWER, has finally a very comparable –slightly lower– power than LAMP. The method developed in [165] could be further extended, using permutation testing or correcting for covariates.

Sixth, in the thesis, we mainly focused on FWER control, as it combines well with Tarone's method, can be controlled regardless of the joint distribution of the test statistics and has been widely adopted to control error rate. However, as mentioned Section 2.3.1, another popular error rate in GWAS analyses is the false discovery rate (FDR) as it is does not have some of the limitations of the FWER. The main disadvantage of the FWER is that it can be overly conservative as it requires that not a single error is made. However, in several tasks, this can be stringent and lead to a too large loss of truly associated patterns. In these tasks, it would be interesting to integrate FDR to significant pattern mining algorithms to obtain a larger statistical power. To this end, it would be possible to use adequate procedures that control the FDR. The most common is the Benjamini-Hochberg (BH) procedure [101], which is correct under the assumption, violated in pattern mining, that the test statistics are independent and under certain assumptions of dependence [166]. In case the BH procedure does not apply, the Benjamini-Yekutieli (BY) [166] has been proposed, which is valid in any condition of dependence between the test statistics. However, this BY procedure might lose the main advantage that BH has over FWER, as it is known to be overly conservative. Additionally, in significant pattern mining, determining which approach applies to the problem at stake is in general not trivial. Another obstacle towards using FDR in significant pattern mining is its computational feasibility. Both BH and BY procedures require to compute all p-values and to sort them, from the least to the most significant, to find the set of hypotheses to be rejected. As we saw in this thesis, computing all p-values naively would be computationally unfeasible. Thus, developing step-down procedures together with an adequate pruning criterion, such as done in the algorithms presented in this thesis, could be more promising. Some algorithms have been developed in this direction. [167] proposes to combine Tarone with the BH procedure, however without implementing an efficient pruning criterion. [168] proposes a pioneer pattern mining algorithm to control the FDR. This novel algorithm is based on two aspects. First, it relies on the notion of quasi-testability, which substitutes the notion of testability introduced by Tarone and requires to know a priori the number of patterns that would be deemed significant. Second, it requires to split the dataset to estimate the number of significant patterns. However, this method could be further improved, as the use of data splitting might be inefficient in terms of statistical power and lead to unstable results. Therefore, further developing FDR-based significant pattern mining algorithms remains an important and active area of research.

At last, moving away from Tarone's framework, it would be possible to use newly developed statistical frameworks such as the model-X knockoff introduced in [169] to control the false discovery rate (FDR). In summary, model-X knockoffs provide a flexible framework to perform hypothesis testing with false discovery rate control. In particular, they can be used in combination with any machine learning model as long as it is possible to generate a set of "knockoff" features that have the same joint distribution as the original features in addition to satisfying two properties: 1) the swap property – informally, the joint distribution of original and "knockoffs" features must be invariant to interchanging any subset of original features with their knockoffs – and 2) being independent of the labels given the original features. Combining the knockoff framework with a graph regularisation constraint [170] would make it possible to discover associated SNPs, guided by the graph prior knowledge, under false discovery rate control. Towards this objective, several questions would need to be answered. A first one concerns the fact that the knockoff framework loses power when features are correlated, which is the case of GWAS datasets. To this end, it would be possible to explore different manners to filter correlated SNPs, the one proposed in [169] would be a good starting point. Another crucial question concerns the need to correct for population structure in GWAS datasets in order to make the method widely applicable. A correction similar to the one introduced in [109] applied to the design matrix could be examined, in which case it would be necessary to verify that the two knockoff conditions still hold. It would also be possible to generate the knockoffs with a (deep) generative model [171–173], as GWAS data contains binary, non-symmetric features, for which the gaussian assumption in [169] no longer applies. Finally, one could also explore various graph regularisation constraints to find the one that fits best the problem at stake.

4.2.2 *Large-scale sequence-function datasets unleash multiple opportunities in machine learning*

While the model `SAPIENs` presented in this thesis allowed to obtain high predictive performance and showed a proof-of-concept example of sequence design, `SAPIENs` could be extended in several directions in order to either improve performance or characterise different aspects of the regulatory region of interest.

First, it would be interesting to train the model on a different types of input sequence, either on longer ribosome binding sites, or on other regulatory regions such as promoters, on the ribosome binding site together with the start codon or 5′-end of the coding sequence of interest. In these cases, as the length of the input might vary across samples, the neural network could use padding or recurrent neural networks [174] to adapt to the input length. It would also be possible to combine the sequence information with secondary structure information obtained with packages such as RNAfold [175].

Second, while we showed with a proof-of-concept experiment that we could successfully design a degenerate RBS whose related library would be enriched in strong RBSs compared to a fully uniformly randomised RBS library, it would be interesting to continue to explore how to best design RBSs. More generally, this fits into the more general problem of sequence design, which includes other regulatory sequences or protein sequences, among others. Such tasks would be particularly interesting when designing sequences that have a given activity and respect design constraints such as length or position-specific nucleotides. Existing generative models [176, 177] would be an interesting first step towards this goal. However, these models would generate sequences that follow the distribution of the training set and are not specific enough. To overcome this problem, it is possible to guide the model towards designing sequences with the activity of interest, as proposed in [178, 179]. However, to the best of my knowledge no generative model to date includes constraints in the generative phase, which would not be simply solved by a filtering step after the generation. Additionally, unlike with text or image generation which can be, preliminary, validated with common sense, the safest way to validate generate DNA or protein sequences is to generate them and measure their activity in the lab. However, this can be costly and time consuming. Therefore, finding a meaningful measure of performance of generative models that design sequences with a minimal access to wet-lab would be a first interesting problem [180, 181].

Third, as we saw in this thesis, several interpretability methods were used to validate prior knowledge or gain new insight on the constituents of the ribosome binding sites. Towards this end, the integrated gradient-based approaches are promising as they enable to attribute to bases and positions in the sequence part of the predicted value. However, this method has a few downsides, such as the need to (often arbitrarily) choose a baseline that might greatly impact the resulting attribution scores and the fact that the attribution scores are sequence-dependent, lacking a principled approach to aggregate them into global feature or motif importance scores. [182] proposes a heuristic approach to aggregate the scores by means of clustering similar informative subsequences together in order to reveal motifs of importance. However, there is still a need for approaches that would directly find regions of interest across a subset of or all samples, accounting for translation invariance. Another interpretability method would be to use an attention mechanism [183] to the model used to perform the predictions, in order to select bases and positions of interest, regardless of the sequence. In summary, the development of interpretable methods for supervised methods in deep learning applied to genomics remains an important domain yet to be explored.

4.2.3 *Closing remarks*

Due to the high complexity of biological mechanisms, accounting for interacting features in machine learning tasks, when attempting to solve biological problems, has been shown to be unavoidable. However, the complexity in the bioinformatics field is, nowadays, not only induced by biological mechanisms but also by the large diversity of biological datatypes. This diversity has grown exponentially the last few years thanks to technologies such as next-generation sequencing, flow cytometry or microfluidics. These different datasets, such as GWAS datasets, single-cell data, protein-protein networks and molecular pathways, epigenetic databases, to only cite a few, call for the development of methods that can account for this technical complexity beyond feature interactions. However, while still being extremely challenging, the substantial progress of machine learning and of bioinformatics in the last few years suggest that working in machine learning applied to biology would stay a promising research topic for the next years to come.

A

APPENDIX

A.1 PROOF OF LEMMA 2 SECTION 2.4.4

Lemma 2 *Let $\mathcal{S} \in \mathcal{P}_P$ be a potentially prunable feature subset. The optimum $\mathbf{x}^*_{\mathcal{S}'}$ of the discrete optimization problem $\min\limits_{\mathbf{x}_{\mathcal{S}'} \leq \mathbf{x}_{\mathcal{S}}} \Psi_{cmh}(\mathbf{x}_{\mathcal{S}'})$ satisfies $x^*_{\mathcal{S}',j} = 0$ or $x^*_{\mathcal{S}',j} = x_{\mathcal{S},j}$ for each $j = 1, \ldots, k$*

PROOF: From the proof of Proposition 1 above, we have $\Psi_{cmh}(\mathcal{S}) = 1 - F_{\chi^2_1}\left(T^{max}_{\mathcal{S}}(\mathbf{x}_{\mathcal{S}})\right)$ with $T^{max}_{\mathcal{S}}(\mathbf{x}_{\mathcal{S}}) = \max\left(T_{\mathcal{S}}\left(a_{\mathcal{S},min}, \mathbf{x}_{\mathcal{S}}\right), T_{\mathcal{S}}\left(a_{\mathcal{S},max}, \mathbf{x}_{\mathcal{S}}\right)\right)$. Since $\mathcal{S} \in \mathcal{P}_P$, we can write:

$$T_l(\mathbf{x}_{\mathcal{S}}) := T_{\mathcal{S}}\left(a_{\mathcal{S},min}, \mathbf{x}_{\mathcal{S}}\right) = \frac{\left(\sum_{j=1}^k \gamma_j x_{\mathcal{S},j}\right)^2}{\sum_{j=1}^k \gamma_j (1-\gamma_j) x_{\mathcal{S},j} \left(1 - \frac{x_{\mathcal{S},j}}{n_j}\right)} \tag{A.1}$$

$$T_r(\mathbf{x}_{\mathcal{S}}) := T_{\mathcal{S}}\left(a_{\mathcal{S},max}, \mathbf{x}_{\mathcal{S}}\right) = \frac{\left(\sum_{j=1}^k (1-\gamma_j) x_{\mathcal{S},j}\right)^2}{\sum_{j=1}^k \gamma_j (1-\gamma_j) x_{\mathcal{S},j} \left(1 - \frac{x_{\mathcal{S},j}}{n_j}\right)} \tag{A.2}$$

where we have used that $\mathcal{S} \in \mathcal{P}_P \Rightarrow x_{\mathcal{S},j} \leq \min(n_{1,j}, n_{2,j})\ \forall j = 1, \ldots, k$ and defined the class ratios $\gamma_j := \min(n_{1,j}, n_{2,j})/n_j$ for each $j = 1, \ldots, k$. Note also that minimising $\Psi_{cmh}(\mathbf{x}_{\mathcal{S}'})$ on $\mathbf{x}_{\mathcal{S}'} \leq \mathbf{x}_{\mathcal{S}}$ is in this case equivalent to maximising between $T_l(\mathbf{x}_{\mathcal{S}'})$ and $T_r(\mathbf{x}_{\mathcal{S}'})$.

As a first step towards proving Lemma 2, we will show that the functions $T_l(\mathbf{x}_{\mathcal{S}'})$ and $T_r(\mathbf{x}_{\mathcal{S}'})$ are both maximised with respect to a single argument $x_{\mathcal{S}',i}$ while keeping the other arguments $x_{\mathcal{S}',j}$, $j \neq i$ fixed at either: (I) $x_{\mathcal{S}',i} = 0$ or (II) $x_{\mathcal{S}',i} = x_{\mathcal{S},i}$. To show that, we compute the partial derivative of $T_l(\mathbf{x}_{\mathcal{S}'})$ and $T_r(\mathbf{x}_{\mathcal{S}'})$ with respect to $x_{\mathcal{S}',i}$:

$$\frac{\partial T_l(\mathbf{x}_{\mathcal{S}'})}{\partial x_{\mathcal{S}',i}} = \Lambda_l(\mathbf{x}_{\mathcal{S}'}) A_l(\mathbf{x}_{\mathcal{S}'}) \tag{A.3}$$

$$\frac{\partial T_r(\mathbf{x}_{\mathcal{S}'})}{\partial x_{\mathcal{S}',i}} = \Lambda_r(\mathbf{x}_{\mathcal{S}'}) A_r(\mathbf{x}_{\mathcal{S}'}) \tag{A.4}$$

with

$$\Lambda_l(\mathbf{x}_{\mathcal{S}'}) = \frac{\gamma_i \sum_{j=1}^k \gamma_j x_{\mathcal{S}',j}}{\left(\sum_{j=1}^k \gamma_j (1-\gamma_j) x_{\mathcal{S}',j} (1 - \frac{x_{\mathcal{S}',j}}{n_j})\right)^2} \tag{A.5}$$

$$\Lambda_r(\mathbf{x}_{\mathcal{S}'}) = \frac{(1-\gamma_i) \sum_{j=1}^k (1-\gamma_j) x_{\mathcal{S}',j}}{\left(\sum_{j=1}^k \gamma_j (1-\gamma_j) x_{\mathcal{S}',j} (1 - \frac{x_{\mathcal{S}',j}}{n_j})\right)^2} \tag{A.6}$$

$$A_l(\mathbf{x}_{\mathcal{S}'}) = \sum_{j=1,j\neq i}^{k} \gamma_j x_{\mathcal{S}',j} \left(2(1-\gamma_j)(1-\frac{x_{\mathcal{S}',j}}{n_j}) - (1-\gamma_i)\right) \tag{A.7}$$

$$+ (1-\gamma_i) \left(\gamma_i + \frac{2}{n_i} \sum_{j=1,j\neq i}^{k} \gamma_j x_{\mathcal{S}',j}\right) x_{\mathcal{S}',i} \tag{A.8}$$

$$A_r(\mathbf{x}_{\mathcal{S}'}) = \sum_{j=1,j\neq i}^{k} (1-\gamma_j) x_{\mathcal{S}',j} \left(2\gamma_j(1-\frac{x_{\mathcal{S}',j}}{n_j}) - \gamma_i\right) \tag{A.9}$$

$$+ \gamma_i \left((1-\gamma_i) + \frac{2}{n_i} \sum_{j=1,j\neq i}^{k} (1-\gamma_j) x_{\mathcal{S}',j}\right) x_{\mathcal{S}',i} \tag{A.10}$$

Because $\Lambda_l(\mathbf{x}_{\mathcal{S}'}) \geq 0$ and $\Lambda_r(\mathbf{x}_{\mathcal{S}'}) \geq 0$, the sign of the partial derivatives are determined by the sign of $A_l(\mathbf{x}_{\mathcal{S}'})$ and $A_r(\mathbf{x}_{\mathcal{S}'})$ respectively. In both cases, $A(\mathbf{x}_{\mathcal{S}'})$ can be expressed as $A(\mathbf{x}_{\mathcal{S}'}) = b(\mathbf{x}_{\neg i,\mathcal{S}'}) + \mu(\mathbf{x}_{\neg i,\mathcal{S}'}) x_{\mathcal{S}',i}$, with $\mathbf{x}_{\neg i,\mathcal{S}'}$ containing $\left\{x_{\mathcal{S}',j}\right\}_{j=1,j\neq i}^{k}$. That is, $A(\mathbf{x}_{\mathcal{S}'})$ is an affine function of $x_{\mathcal{S}',i}$ where the intersect and slope is controlled by all other $k-1$ variables. Moreover, for any $\mathbf{x}_{\neg i,\mathcal{S}'}$ we have $\mu(\mathbf{x}_{\neg i,\mathcal{S}'}) \geq 0$. Therefore the partial derivatives are either always positive, always negative, or negative until a unique point where it crosses zero and then positive. As a consequence, it follows that the only possible maximizers of $T_l(\mathbf{x}_{\mathcal{S}'})$ and $T_r(\mathbf{x}_{\mathcal{S}'})$ with respect to $x_{\mathcal{S}',i}$ are at the boundary of the domain, i.e. either $x_{\mathcal{S}',i} = 0$ or $x_{\mathcal{S}',i} = x_{\mathcal{S},i}$. In other words, we have $\max\limits_{\mathbf{x}_{\mathcal{S}'} \leq \mathbf{x}_{\mathcal{S}}} T_l(\mathbf{x}_{\mathcal{S}'}) = \max\left(\max\limits_{\mathbf{x}_{\neg i,\mathcal{S}'} \leq \mathbf{x}_{\neg i,\mathcal{S}}} T_l(\mathbf{x}_{\neg i,\mathcal{S}'},0), \max\limits_{\mathbf{x}_{\neg i,\mathcal{S}'} \leq \mathbf{x}_{\neg i,\mathcal{S}}} T_l(\mathbf{x}_{\neg i,\mathcal{S}'},x_{\mathcal{S},i})\right)$. Since this holds for any $i = 1,2,\ldots,k$ and $\mathbf{x}_{\mathcal{S}'} \leq \mathbf{x}_{\mathcal{S}}$, the argument can be applied recursively to each of the two terms in the RHS of the last expression. The same reasoning holds for T_r. This completes the proof.

A.2 PROOF OF THEOREM 2 SECTION 2.4.4

Theorem 2 *Let $\mathcal{S} \in \mathcal{P}_P$ be a potentially testable feature subset and define $\beta^l_{\mathcal{S},j} = \frac{n_{2,j}}{n_j}\left(1 - \frac{x_{\mathcal{S},j}}{n_j}\right)$ and $\beta^r_{\mathcal{S},j} = \frac{n_{1,j}}{n_j}\left(1 - \frac{x_{\mathcal{S},j}}{n_j}\right)$ for $j = 1,\ldots,k$. Let π_l and π_r be permutations $\pi_l,\pi_r : [\![1,k]\!] \mapsto [\![1,k]\!]$ such that $\beta^l_{\mathcal{S},\pi_l(1)} \leq \ldots \leq \beta^l_{\mathcal{S},\pi_l(k)}$ and $\beta^r_{\mathcal{S},\pi_r(1)} \leq \ldots \leq \beta^r_{\mathcal{S},\pi_r(k)}$, respectively.*

Then, there exists an integer $\kappa \in [\![1,k]\!]$ such that the optimum $\mathbf{x}^*_{\mathcal{S}'} = \arg\min\limits_{\mathbf{x}_{\mathcal{S}'} \leq \mathbf{x}_{\mathcal{S}}} \Psi_{cmh}(\mathbf{x}_{\mathcal{S}'})$ satisfies one of the two possible conditions: (I) $x^*_{\mathcal{S}',\pi_l(j)} = x_{\mathcal{S},\pi_l(j)}$ for all $j \leq \kappa$ and $x^*_{\mathcal{S}',\pi_l(j)} = 0$ for all $j > \kappa$ or (II) $x^*_{\mathcal{S}',\pi_r(j)} = x_{\mathcal{S},\pi_r(j)}$ for all $j \leq \kappa$ and $x^*_{\mathcal{S}',\pi_r(j)} = 0$ for all $j > \kappa$.

PROOF: The functions $T_l(\mathbf{x}_{\mathcal{S}'})$ and $T_r(\mathbf{x}_{\mathcal{S}'})$ defined in the proof of Lemma 2 above can be rewritten generically as:

$$T = \frac{\left(\sum_{j=1}^{k} l_j(\beta_j)\right)^2}{\sum_{j=1}^{k} \beta_j l_j(\beta_j)} \tag{A.11}$$

with $\beta_j \in [0,1]$ and $l_j(\beta_j) > 0$. In particular, $\beta_j = (1-\gamma_j)(1-\frac{x_{\mathcal{S},j}}{n_j})$ and $l_j(\beta_j) = n_{1,j}\left(1-\frac{\beta_j}{1-\gamma_j}\right)$ for $T_l(\mathbf{x}_{\mathcal{S}'})$ whereas $\beta_j = \gamma_j(1-\frac{x_{\mathcal{S},j}}{n_j})$ and $l_j(\beta_j) = n_{2,j}\left(1-\frac{\beta_j}{1-\gamma_j}\right)$ for $T_r(\mathbf{x}_{\mathcal{S}'})$. Since T is permutation invariant, we assume without loss of generality that the indices $j = 1,\ldots,k$ have been sorted a priori to guarantee that $\beta_j \le \beta_i$ whenever $j \le i$.

Next, we introduce k binary indicator variables $\delta_1,\ldots,\delta_k$ in T as:

$$T(\delta_1,\ldots,\delta_k) = \frac{\left(\sum_{j=1}^{k} \delta_j l_j(\beta_j)\right)^2}{\sum_{j=1}^{k} \delta_j \beta_j l_j(\beta_j)} \tag{A.12}$$

Suppose further that:

$$\underset{\delta_1,\ldots,\delta_k}{\arg\max}\; T(\delta_1,\ldots,\delta_k) = (\underbrace{1,1,\ldots,1}_{r},\underbrace{0,0,\ldots,0}_{k-r}) \tag{A.13}$$

holds with $r > 0$. Informally, Equation (A.13) being true would imply that the maximum is achieved by keeping the terms in the summation corresponding to the r smallest β_j. From Lemma 2, we know that $T_l(\mathbf{x}_{\mathcal{S}'})$ and $T_r(\mathbf{x}_{\mathcal{S}'})$ are maximised for $x^*_{\mathcal{S}',j} = 0$ or $x^*_{\mathcal{S}',j} = x_{\mathcal{S},j}$ for all $j = 1,2,\ldots,k$. Thus, if Equation (A.13) holds and we have $\beta_i > \beta_j$ and $x^*_{\mathcal{S}',i} = x_{\mathcal{S},i}$ then it follows that $x^*_{\mathcal{S}',j} = x_{\mathcal{S},j}$. The alternative case $x^*_{\mathcal{S}',j} = 0$ cannot occur, since it would contradict Equation (A.13). This would suggest the following strategy to solve $\min_{\mathbf{x}_{\mathcal{S}'} \le \mathbf{x}_{\mathcal{S}}} \Psi_{cmh}(\mathbf{x}_{\mathcal{S}'})$. Firstly, as shown in the proof of Lemma 2, $\underset{\mathbf{x}_{\mathcal{S}'} \le \mathbf{x}_{\mathcal{S}}}{\arg\min} \Psi_{cmh}(\mathbf{x}_{\mathcal{S}'}) = \underset{\mathbf{x}_{\mathcal{S}'} \le \mathbf{x}_{\mathcal{S}}}{\arg\max} \max\left(T_l(\mathbf{x}_{\mathcal{S}'}),T_r(\mathbf{x}_{\mathcal{S}'})\right)$. Next, we obtain and sort the coefficients $\left\{\beta^l_j\right\}_{j=1}^{k}$ and $\left\{\beta^r_j\right\}_{j=1}^{k}$ corresponding to the representation of T_l and T_r in the form of Equation (A.11). The computational complexity of that step would be dominated by the sorting steps, hence being $O(k\log(k))$. Then, by Equation (A.13) and Lemma 2, we can solve the subproblems $\underset{\mathbf{x}_{\mathcal{S}'} \le \mathbf{x}_{\mathcal{S}}}{\arg\max} T_l(\mathbf{x}_{\mathcal{S}'})$ and $\underset{\mathbf{x}_{\mathcal{S}'} \le \mathbf{x}_{\mathcal{S}}}{\arg\max} T_r(\mathbf{x}_{\mathcal{S}'})$ in $O(k)$ time each, increasing the candidate r in Equation (A.13) from 1 up to at most k. Note that this is exactly the strategy suggested by Theorem 2. In summary, proving Theorem 2 amounts to showing the validity of Equation (A.13) for functions of the form given in Equation (A.12).

We will prove it by induction. First, we show that the statement holds for $k = 2$. That is, we want to show that:

$$\underset{\delta_1,\delta_2}{\arg\max}\; T(\delta_1,\delta_2) \in \{(1,0),(1,1)\} \tag{A.14}$$

The only possible contradicting case would be $\underset{\delta_1,\delta_2}{\arg\max}\, T(\delta_1,\delta_2) = (0,1)$, since the case (0,0) yields a trivial value for the function T. We show directly that under the assumption $\beta_1 \leq \beta_2$, the contradiction cannot happen. Indeed we have:

$$\frac{(l_1(\beta_1)+l_2(\beta_2))^2}{\beta_1 l_1(\beta_1)+\beta_2 l_2(\beta_2)} - \frac{l_2^2(\beta_2)}{\beta_2 l_2(\beta_2)} = \frac{(l_1(\beta_1)+l_2(\beta_2))^2\beta_2 l_2(\beta_2) - l_2^2(\beta_2)(\beta_1 l_1(\beta_1)+\beta_2 l_2(\beta_2))}{(\beta_1 l_1(\beta_1)+\beta_2 l_2(\beta_2))\beta_2 l_2(\beta_2)} \tag{A.15}$$

$$= l_1(\beta_1)\frac{(l_1(\beta_1)+2l_2(\beta_2))\beta_2 l_2(\beta_2) - \beta_1 l_2^2(\beta_2)}{(\beta_1 l_1(\beta_1)+\beta_2 l_2(\beta_2))\beta_2 l_2(\beta_2)}$$

$$= l_1(\beta_1)\frac{\beta_2 l_1(\beta_1) l_2(\beta_2) + (2\beta_2-\beta_1) l_2^2(\beta_2)}{(\beta_1 l_1(\beta_1)+\beta_2 l_2(\beta_2))\beta_2 l_2(\beta_2)}$$

Since $l_i(\beta_i) \geq 0$ and $\beta_1 \leq \beta_2$, it follows that the numerator in the expression above is positive, thus $T(1,1) > T(0,1)$ contradicting the statement that $\underset{\delta_1,\delta_2}{\arg\max}\, T(\delta_1,\delta_2) = (0,1)$.

Next we prove the induction step. Suppose the statement holds for an arbitrary dimension k, we will show then it also holds for dimension $k+1$. That is, if we have:

$$\underset{\delta_1,\ldots,\delta_k}{\arg\max} \frac{\left(\sum_{i=1}^k \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^k \delta_i \beta_i l_i(\beta_i)} = (\underbrace{1,1,\ldots,1}_{r},\underbrace{0,0,\ldots,0}_{k-r}) \tag{A.16}$$

Then we want to show that:

$$\underset{\delta_1,\ldots,\delta_k,\delta_{k+1}}{\arg\max} \frac{\left(\sum_{i=1}^k \delta_i l_i(\beta_i) + \delta_{k+1} l_{k+1}(\beta_{k+1})\right)^2}{\sum_{i=1}^k \delta_i \beta_i l_i(\beta_i) + \delta_{k+1}\beta_{k+1} l_{k+1}(\beta_{k+1})} = (\underbrace{1,1,\ldots,1}_{r'},\underbrace{0,0,\ldots,0}_{(k+1)-r'}) \tag{A.17}$$

We can start by writing:

$$\max_{\delta_1,\ldots,\delta_k,\delta_{k+1}} \frac{\left(\sum_{i=1}^k \delta_i l_i(\beta_i) + \delta_{k+1} l_{k+1}(\beta_{k+1})\right)^2}{\sum_{i=1}^k \delta_i \beta_i l_i(\beta_i) + \delta_{k+1}\beta_{k+1} l_{k+1}(\beta_{k+1})} \tag{A.18}$$

$$= \max\left(\max_{\delta_1,\ldots,\delta_k} \frac{\left(\sum_{i=1}^k \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^k \delta_i \beta_i l_i(\beta_i)}, \max_{\delta_1,\ldots,\delta_k} \frac{\left(\sum_{i=1}^k \delta_i l_i(\beta_i) + l_{k+1}(\beta_{k+1})\right)^2}{\sum_{i=1}^k \delta_i \beta_i l_i(\beta_i) + \beta_{k+1} l_{k+1}(\beta_{k+1})}\right)$$

If:

$$\max_{\delta_1,\ldots,\delta_k} \frac{\left(\sum_{i=1}^k \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^k \delta_i \beta_i l_i(\beta_i)} \geq \max_{\delta_1,\ldots,\delta_k} \frac{\left(\sum_{i=1}^k \delta_i l_i(\beta_i) + l_{k+1}(\beta_{k+1})\right)^2}{\sum_{i=1}^k \delta_i \beta_i l_i(\beta_i) + \beta_{k+1} l_{k+1}(\beta_{k+1})} \tag{A.19}$$

Then the statement is trivially true. Suppose now that Equation (A.19) does **not** hold. We show next that:

$$(\hat{\delta}_1, \ldots, \hat{\delta}_k) = \arg\max_{\delta_1,\ldots,\delta_k} \frac{\left(\sum_{i=1}^{k} \delta_i l_i(\beta_i) + l_{k+1}(\beta_{k+1})\right)^2}{\sum_{i=1}^{k} \delta_i \beta_i l_i(\beta_i) + \beta_{k+1} l_{k+1}(\beta_{k+1})} = (\underbrace{1,1,\ldots,1}_{k}) \tag{A.20}$$

which would complete the proof. To show that Equation (A.20) is true when Equation (A.19) does **not** hold, we proceed by contradiction in two steps. First we prove that there is at most a single $j \in \{1, \ldots, k\} \mid \hat{\delta}_j = 0$. To see that, we assume $\exists j \mid \hat{\delta}_j = 0$ and $\exists j \mid \hat{\delta}_j = 0$ such that:

$$\max_{\delta_1,\ldots,\delta_{k+1} | \delta_j = 0} \frac{\left(\sum_{i=1}^{k+1} \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^{k+1} \delta_i \beta_i l_i(\beta_i)} = \max_{\delta_1,\ldots,\delta_{k+1}} \frac{\left(\sum_{i=1}^{k+1} \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^{k+1} \delta_i \beta_i l_i(\beta_i)} \tag{A.21}$$

At the same time, as we assume Equation (A.19) does not hold, we have:

$$\max_{\delta_1,\ldots,\delta_{k+1} | \delta_j = 0} \frac{\left(\sum_{i=1}^{k+1} \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^{k+1} \delta_i \beta_i l_i(\beta_i)} \tag{A.22}$$

$$= \max_{\delta_1,\ldots,\delta_{k+1}} \frac{\left(\sum_{i=1}^{k+1} \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^{k+1} \delta_i \beta_i l_i(\beta_i)} \tag{A.23}$$

$$= \max_{\delta_1,\ldots,\delta_k} \frac{\left(\sum_{i=1}^{k} \delta_i l_i(\beta_i) + l_{k+1}(\beta_{k+1})\right)^2}{\sum_{i=1}^{k} \delta_i \beta_i l_i(\beta_i) + \beta_{k+1} l_{k+1}(\beta_{k+1})} \tag{A.24}$$

$$> \max_{\delta_1,\ldots,\delta_k} \frac{\left(\sum_{i=1}^{k} \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^{k} \delta_i \beta_i l_i(\beta_i)} \tag{A.25}$$

However, we could also write:

$$\max_{\delta_1,\ldots,\delta_{k+1} | \delta_j = 0} \frac{\left(\sum_{i=1}^{k+1} \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^{k+1} \delta_i \beta_i l_i(\beta_i)} \tag{A.26}$$

$$= \max(\max_{\delta_1,\ldots,\delta_{k+1} | \delta_j = 0, \delta_{k+1} = 1} \frac{\left(\sum_{i=1}^{k+1} \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^{k+1} \delta_i \beta_i l_i(\beta_i)}, \max_{\delta_1,\ldots,\delta_{k+1} | \delta_j = 0, \delta_{k+1} = 0} \frac{\left(\sum_{i=1}^{k+1} \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^{k+1} \delta_i \beta_i l_i(\beta_i)}) \tag{A.27}$$

By Equation (A.25), we know that:

$$\max_{\delta_1,\ldots,\delta_{k+1}|\delta_j=0} \frac{\left(\sum_{i=1}^{k+1} \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^{k+1} \delta_i \beta_i l_i(\beta_i)} \tag{A.28}$$

$$> \max_{\delta_1,\ldots,\delta_{k+1}|\delta_j=0,\delta_{k+1}=0} \frac{\left(\sum_{i=1}^{k+1} \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^{k+1} \delta_i \beta_i l_i(\beta_i)} \tag{A.29}$$

We can write that:

$$\max_{\delta_1,\ldots,\delta_{k+1}|\delta_j=0} \frac{\left(\sum_{i=1}^{k+1} \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^{k+1} \delta_i \beta_i l_i(\beta_i)} \tag{A.30}$$

$$= \max_{\delta_1,\ldots,\delta_{k+1}|\delta_j=0,\delta_{k+1}=1} \frac{\left(\sum_{i=1}^{k+1} \delta_i l_i(\beta_i)\right)^2}{\sum_{i=1}^{k+1} \delta_i \beta_i l_i(\beta_i)} \tag{A.31}$$

As this last equation corresponds to the optimisation done with only k variables, as written Equation (A.13), and as $\delta_{k+1} = 1$, we can conclude that:

$$\arg\max_{\delta_1,\ldots,\delta_{k+1}|\delta_j=0} \quad T(\delta_1,\ldots,\delta_{k+1}|\delta_j = 0) = (1,1,\ldots,1) \tag{A.32}$$

Therefore, in the case with $k+1$ variables and if Equation (A.19) does not hold, there is at most a single index that can be equal to 0.

To end the proof, we need to show that, indeed, it is not possible to have $\hat{\delta}_j = 0$ either. To do so we will show that the statement of monotonicity holds for $k = 3$, then we could easily show $\hat{\delta}_j = 1$. We use a change of variables to make it clearer.

Indeed, we rewrite $T(\delta_1,\delta_2,\ldots,\delta_j,\ldots,\delta_{k-1},\delta_k)$ the following way :

$$T(\delta_1,\delta_2,\ldots,\delta_{k-1},\delta_k) = \frac{(l'_0 + l_j\delta_j + l_k)^2}{\beta'_0 l'_0 + \beta_j l_j \delta_j + \beta_k l_k}$$

with

$$l'_0 = \sum_{l=1,l\neq j}^{k-1} l_l$$

and

$$l'_0\beta'_0 = \sum_{l=1,l\neq j}^{k-1} l_l\beta_l \Leftrightarrow \beta'_0 = \frac{\sum_{l=1,l\neq j}^{k-1} l_l\beta_l}{\sum_{l=1,l\neq j}^{k-1} l_l}$$

Then, if Equation (A.19) does not hold, we would have:

$$\max_{\delta_1,\ldots,\delta_k,\delta_{k+1}} T(\delta_1,\ldots,\delta_k,\delta_{k+1}) = \max_{\delta_0,\delta_j,\delta_{k+1}} \frac{(\delta_0 l'_0 + \delta_j l_j + \delta_{k+1} l_{k+1})^2}{\delta_0 \beta'_0 l'_0 + \delta_j \beta_j l_j + \delta_{k+1} \beta_{k+1} l_{k+1}} \tag{A.33}$$

where we know, by assumption, that the optimum in the right hand side is achieved when $\delta_0 = 1$ and $\delta_{k+1} = 1$. If we knew monotonicity holds for k=3, it would then follow that $\delta_j = 1$ if $\beta_j \geq \beta'_0$.

To rephrase it, we want to show that the two following cases: $T(\delta_j = 1,\delta_0 = 1,\delta_k = 1) < T(\delta_j = 0,\delta_0 = 1,\delta_k = 1)$ with $\beta_j < \beta'_0$ and $T(\delta_j = 1,\delta_0 = 1,\delta_k = 1) < T(\delta_j = 0,\delta_0 = 1,\delta_k = 1)$ with $\beta_j > \beta'_0$ are impossible with the hypothesis that $\forall \quad \{\delta_1,\delta_2,...,\delta_{k-1}\} \quad T(\delta_1,\delta_2,...,\delta_{k-1},0) < \max_{\delta_1,...,\delta_{k-1}} T(\delta_1,\delta_2,...,\delta_{k-1},1)$

First, we show that when $\beta_j < \beta'_0$, then $T(\delta_j = 1,\delta_0 = 1,\delta_k = 1) > T(\delta_j = 0,\delta_0 = 1,\delta_k = 1)$. Indeed, after developing the difference we obtain :

$$T(\delta_j = 1,\delta_0 = 1,\delta_k = 1) - T(\delta_j = 0,\delta_0 = 1,\delta_k = 1) \tag{A.34}$$

$$= \frac{l_j}{(l'_0\beta'_0 + l_k\beta_k)(l_j\beta_j + l'_0\beta'_0 + l_k\beta_k)} \tag{A.35}$$

$$\times\,(l_0^{2\prime}(2\beta'_0 - \beta_j) + l_k^2(2\beta_k - \beta_j) + l'_0 l_k(2\beta'_0 + 2\beta_k - \beta_j + l'_0 l_j \beta'_0 + l_k l_j \beta_k)) \tag{A.36}$$

$$> 0 \tag{A.37}$$

As $\beta_j < \beta'_0 < \beta_k$, all the terms of the previous sum are positive, which implies that $T(\delta_j = 1,\delta_0 = 1,\delta_k = 1) > T(\delta_j = 0,\delta_0 = 1,\delta_k = 1)$.

In a second time we want to show that the case $T(\delta_0 = 1,\delta_j = 1,\delta_k = 1) < T(\delta_0 = 1,\delta_j = 0,\delta_k = 1)$ with $\beta_j > \beta'_0$ is not possible either. In this case we use a Reductio ad absurdum: we are going to show that we can not have both $T(\delta_0 = 1,\delta_j = 0,\delta_k = 1) > T(1,1,1)$ and $T(\delta_0 = 1,\delta_j = 0,1) > T(\delta_0 = 0,\delta_j = 1,0)$. Indeed after developing both inequalities, we find

$$T(\delta_0 = 1,\delta_j = 0,1) > T(1,1,1) \Leftrightarrow \beta'_0 < \frac{1}{l_0}(\beta_j \frac{(l'_0 + l_k)^2}{2(l'_0 + l_k) + l_j} - \beta_k l_k) \tag{A.38}$$

$$T(\delta_0 = 1,\delta_j = 0,1) > T(\delta_0 = 1,\delta_j = 0,0) \Leftrightarrow \beta'_0 > \frac{l'_0}{2l'_0 + l_k}\beta_k \tag{A.39}$$

The first inequality of Equation (A.38) can be simplified the following way, by using the following inequalities $\beta'_0 < \beta_j < \beta_k$ and $\forall i \quad l_i > 0$.

$$\beta'_0 < \frac{1}{l'_0}(\beta_i \frac{(l'_0 + l_k)^2}{2(l'_0 + l_k) + l_j} - \beta_k l_k) \tag{A.40}$$

$$< \frac{1}{l'_0}(\beta_k \frac{(l'_0 + l_k)^2}{2(l'_0 + l_k) + l_j} - \beta_k l_k) = \beta_k(\frac{1}{l'_0}\frac{(l'_0 + l_k)^2}{2(l'_0 + l_k) + l_j} - l_k) \tag{A.41}$$

Using Equation (A.38) and Equation (A.40) we have the following result :

$$\frac{l'_0}{2l'_0 + l_k}\beta_k < \beta'_0 < \beta_k \frac{1}{l'_0}\left(\frac{(l'_0 + l_k)^2}{2(l'_0 + l_k) + l_j} - l_k\right) \tag{A.42}$$

$$\Rightarrow \frac{l'_0}{2l'_0 + l_k} < \frac{1}{l'_0}\left(\frac{(l'_0 + l_k)^2}{2(l'_0 + l_k) + l_i} - l_k\right) \tag{A.43}$$

$$\Rightarrow 0 < -(l'_0 + l_k)^2(l_k + l_j) \tag{A.44}$$

The last line of the previous equation set shows clearly the contradiction. Those two results Equation (A.34) and Equation (A.42) end the proof.

A.3 STATISTICALLY SIGNIFICANT GENOMIC REGIONS FOUND BY FastCMH (*A. thaliana*)

SNPs in the significant genomic region	Gene overlap	*p*-values FastCMH
avrB		
▷ *Chr*3_2225653 - *Chr*3_2225893	NA	3.15e-13
▷ *Chr*3_2221399 - *Chr*3_2222856	AT3G07020	1.61e-10
▷ *Chr*3_2227817	AT3G07040	1.69e-10
▷ *Chr*3_2288913 - *Chr*3_2289178 - *Chr*3_2289559	AT3G07195	5.34e-10
avrRpm1		
▷ *Chr*3_2225653 - *Chr*3_2225893	NA	3.00e-13
▷ *Chr*3_2227817	AT3G07040	1.15e-11
▷ *Chr*3_2310055 - *Chr*3_2311035 - *Chr*3_2311574	AT3G07260	6.90e-11
avrPphB		
▷ *Chr*1_4146714	AT1G12220	1.87e-13
▷ *Chr*1_4143163	AT1G12210	1.87e-13
▷ *Chr*1_4141624	AT1G12210	1.17e-12
▷ *Chr*1_4139802 - *Chr*1_4140044	AT1G12200	2.43e-12
LES		
▷ *Chr*4_8297892	AT4G14400	1.37e-09
▷ *Chr*5_6485290	NA	7.39e-09
▷ *Chr*4_8307440 - *Chr*4_8307761 - *Chr*4_8307910 - *Chr*4_8308076 - *Chr*4_8308306 - *Chr*4_8308768 - *Chr*4_8308977	AT4G14440	2.25e-08
LY		
▷ *Chr*5_18925351	AT5G46640	1.27e-08

TABLE A.1: **Details of the most statistically significant genomic regions reported by FastCMH in the *A. thaliana* datasets.** Underlined SNPs are contained in genes (including markers at a distance smaller than 10 kb). The SNP notation in the format: *Chr*4_2398754 indicates a SNP located on the 4th chromosome at the position 2398754.

A.4 STATISTICALLY SIGNIFICANT GENOMIC REGIONS FOUND BY FastCMH (COPD)

Genomic region	Gene overlap	*p*-values FastCMH	
		region	single SNP
chr15 78863472–78865893	CHRNA5	2.03e-10	
rs667282			9.06e-08
rs6495306			4.60e-02
chr15 78907656–78909480	CHRNA3	4.54e-10	
rs6495308			3.55e-05
rs12443170			2.96e-03
rs3743074			4.30e-02
chr15 78917399–78928264	CHRNB4	1.41e-10	
rs1948			1.00e-02
rs950776			6.18e-02
rs12441088			1.70e-05

TABLE A.2: **Details of the statistically significant genomic regions reported by FastCMH and of the SNPs they contain in the COPD dataset.**

Corrected significance threshold for:

- all testable intervals: 7.26e-09
- all testable single SNPs: 8.12e-08

A.5 statistically significant genomic regions found by burden tests (*A. thaliana*)

Significant gene	Number of SNPs	`FastCMH`	Burden tests and p-values				
			dummy - (I)	*dummy - (II)*	*PCs - (I)*	*PCs - (II)*	*CMH*
avrB							
▷ **AT3G07050**	10	1.66e-09	|	1.01e-06	|	1.76e-07	|
▷ **AT3G07195**	4	5.34e-10	1.23e-10	6.45e-10	2.01e-11	1.78e-10	2.77e-10
▷ **AT3G07010**	10	5.24e-09	1.44e-06	|	|	|	|
▷ **AT3G07020**	|	1.61e-10	|	|	|	|	|
▷ **AT3G07040**	|	1.69e-10	|	|	|	|	|
▷ **AT3G07060**	|	3.58e-09	|	|	|	|	|
▷ **AT3G07070**	|	3.58e-09	|	|	|	|	|
▷ **AT3G07260**	|	6.68e-10	|	|	|	|	|
▷ **AT3G07330**	|	2.12e-09	|	|	|	|	|
avrRpm1							
▷ **AT3G07050**	10	6.42e-10	|	4.04e-07	|	8.75e-08	|
▷ **AT3G07195**	4	5.06e-10	4.04e-10	1.07-10	5.11e-11	1.39e-11	5.07e-10
▷ AT3G07005	10	|	7.20e-07	7.20e-07	|	|	|
▷ **AT3G07020**	|	1.81e-10	|	|	|	|	|
▷ **AT3G07040**	|	1.15e-11	|	|	|	|	|
▷ **AT3G07060**	|	6.26e-09	|	|	|	|	|
▷ **AT3G07070**	|	6.26e-09	|	|	|	|	|
▷ **AT3G07200**	|	1.77e-08	|	|	|	|	|
▷ **AT3G07250**	|	1.77e-08	|	|	|	|	|
▷ **AT3G07260**	|	6.90e-11	|	|	|	|	|
▷ **AT3G07330**	|	7.5e-09	|	|	|	|	|
avrPphB							
▷ **AT1G12210**	9	1.87e-13	1.67e-06	1.18e-15	6.02e-08	7.94e-20	|
▷ **AT1G12220**	3	1.87e-13	3.92e-14	6.18e-16	3.25e-17	2.43e-19	6.12e-13
▷ AT1G12230	3	|	7.78e-14	3.83e-15	2.76e-14	1.57e-16	3.19e-12
▷ AT5G11340	5	|	|	|	|	8.91e-07	|
▷ AT5G11350	3	|	1.08e-06	|	4.77e-08	1.58e-07	9.94e-07
▷ AT1G12170	5	|	|	|	7.89e-07	|	|
▷ **AT1G12200**	|	2.43e-12	|	|	|	|	|
▷ **AT1G12190**	|	7.95e-09	|	|	|	|	|
LES							
▷ AT3G06120	3	|	|	2.01e-07	|	|	|
▷ AT4G28890	7	|	4.38e-07	4.28e-07	1.77e-07	1.77e-07	|
▷ AT4G14410	9	|	|	|	|	2.74e-07	|
▷ AT1G34420	3	|	|	|	|	1.87e-06	|
▷ AT1G08500	2	|	|	|	1.28e-07	1.28e-07	|
▷ AT5G45780	6	|	|	|	3.99e-07	2.45e-07	|
▷ AT3G18535	5	|	|	|	|	|	1.25e-06
▷ AT4G39955	7	|	|	|	|	|	4.36e-07
▷ **AT4G14440**	|	2.25e-08	|	|	|	|	|
▷ **AT4G14400**	|	1.37e-09	|	|	|	|	|

Significant gene	Number of SNPs	`FastCMH`	Burden tests and p-values				
			dummy - (I)	*dummy - (II)*	*PCs - (I)*	*PCs - (II)*	*CMH*
LY							
▷ AT1G34420	3	|	|	1.21e-07	|	|	|
▷ AT2G38995	8	|	|	1.63e-06	|	|	|
▷ AT3G61480	5	|	1.57e-06	1.57e-06	|	|	|
▷ AT5G46660	5	|	|	1.99e-06	|	5.06e-07	|
▷ AT5G49620	1	|	1.61e-06	1.61e-06	|	|	|
▷ AT5G45780	6	|	|	|	|	1.49e-06	|
▷ AT1G08500	2	|	|	|	3.36e-08	3.36e-08	|
▷ AT2G18120	3	|	|	2.08e-07	1.87e-06	|	|
▷ **AT5G46640**	|	1.27e-08	|	|	|	|	|

TABLE A.3: **The statistically significant genomic regions reported by the different gene-based burden tests (resp. `FastCMH`) and the corresponding gene (resp. genomic region) p-values when significant.** A vertical bar | indicates that the gene is not significant for the given test. In bold, we indicate genes that are found by `FastCMH`. Keys to abbreviations: *dummy* indicates that the covariates are coded as k dummy indicator variables, *PCs* means that we chose the three first principal components of the kinship matrix as covariates, *(I)* and *(II)* correspond to the encodings described in Section 2.5.4.3. Finally, *CMH* corresponds to the burden test using the CMH test applied to encoding *(I)* for each gene.

A.6 STATISTICALLY SIGNIFICANT GENOMIC REGIONS FOUND BY BURDEN TESTS (COPD)

For the COPDGene study, when performing the gene-based burden tests as described Section 2.5.4.3, none of the three genes (CHRNA5-CHRNA3-CHRNB4) found by `FastCMH` were significant using any of the burden tests. When taking the smallest p-value across all burden tests performed, only CHRNB4 was close to significance (p-value 5.72e-6) while CHRNA5 and CHRNA3 had p-values 0.24 and 0.41, respectively. While each of the three significantly associated genomic regions found by `FastCMH` overlaps with one gene in the cluster (CHRNA5-CHRNA3-CHRNB4), the significant regions do not span the entire gene. Burden tests, which do not consider sub-regions, include too many markers in the test, diluting the signal among noise and missing the association. In contrast, gene-based burden tests identified the gene ZRANB3 as significant, with the smallest p-value across all burden tests being 1.56e-6. `FastCMH` assigns the genomic region corresponding to ZRANB3 a very similar p-value, 2.31e-6. However, ZRANB3 is not significantly associated for `FastCMH` because it uses a more stringent significance threshold. This behavior is to be expected, as there are many more testable genomic regions ($\approx 7 \cdot 10^6$) than genes ($\approx 1.7 \cdot 10^3$) in the COPDGene dataset.

When the window-based burden tests were conducted on the two window sizes used to partition the genome (500 kilobases and 1 megabase) as described Section 2.5.4.3, the results coincided with the findings of the gene-based tests (i.e., overlap with the gene ZRANB3). For 500 kb windows, only the region chr2:136,018,946:136,518,946 is found when the encoding is (I). Likewise, for 1 megabase windows, the region chr2:136,018,810:137,018,810 is found when the encoding used is (II).

BIBLIOGRAPHY

1. Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., Simonovic, M., Roth, A., Santos, A., Tsafou, K. P., *et al.* STRING v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic acids research* **43**, D447 (2015).
2. Li, T., Wernersson, R., Hansen, R. B., Horn, H., Mercer, J., Slodkowicz, G., Workman, C. T., Rigina, O., Rapacki, K., Stærfeldt, H. H., *et al.* A scored human protein–protein interaction network to catalyze genomic interpretation. *Nature methods* **14**, 61 (2017).
3. Kanehisa, M., Goto, S., Kawashima, S. & Nakaya, A. The KEGG databases at GenomeNet. *Nucleic acids research* **30**, 42 (2002).
4. Peled, S., Leiderman, O., Charar, R., Efroni, G., Shav-Tal, Y. & Ofran, Y. De-novo protein function prediction using DNA binding and RNA binding proteins as a test case. *Nature communications* **7**, 1 (2016).
5. Zhu, F., Farnung, L., Kaasinen, E., Sahu, B., Yin, Y., Wei, B., Dodonova, S. O., Nitta, K. R., Morgunova, E., Taipale, M., *et al.* The interaction landscape between transcription factors and the nucleosome. *Nature* **562**, 76 (2018).
6. Qian, Z., Zhurkin, V. B. & Adhya, S. DNA–RNA interactions are critical for chromosome condensation in Escherichia coli. *Proceedings of the National Academy of Sciences* **114**, 12225 (2017).
7. Howard, T. D., Koppelman, G. H., Xu, J., Zheng, S. L., Postma, D. S., Meyers, D. A. & Bleecker, E. R. Gene-gene interaction in asthma: IL4RA and IL13 in a Dutch population with asthma. *The American Journal of Human Genetics* **70**, 230 (2002).
8. Cho, Y. M., Ritchie, M. D., Moore, J., Park, J., Lee, K.-U., Shin, H., Lee, H. & Park, K. S. Multifactor-dimensionality reduction shows a two-locus interaction associated with Type 2 diabetes mellitus. *Diabetologia* **47**, 549 (2004).
9. Borrell, S., Teo, Y., Giardina, F., Streicher, E. M., Klopper, M., Feldmann, J., Müller, B., Victor, T. C. & Gagneux, S. Epistasis between antibiotic resistance mutations drives the evolution of extensively drug-resistant tuberculosis. *Evolution, medicine, and public health* **2013**, 65 (2013).
10. Visscher, P. M., Brown, M. A., McCarthy, M. I. & Yang, J. Five years of GWAS discovery. *The American Journal of Human Genetics* **90**, 7 (2012).
11. Visscher, P. M., Wray, N. R., Zhang, Q., Sklar, P., McCarthy, M. I., Brown, M. A. & Yang, J. 10 years of GWAS discovery: biology, function, and translation. *The American Journal of Human Genetics* **101**, 5 (2017).

12. Dimitrakopoulos, C. M. & Beerenwinkel, N. Computational approaches for the identification of cancer genes and pathways. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* **9**, e1364 (2017).

13. Yeang, C.-H., McCormick, F. & Levine, A. Combinatorial patterns of somatic gene mutations in cancer. *The FASEB journal* **22**, 2605 (2008).

14. Han, K., Jeng, E. E., Hess, G. T., Morgens, D. W., Li, A. & Bassik, M. C. Synergistic drug combinations for cancer identified in a CRISPR screen for pairwise genetic interactions. *Nature biotechnology* **35**, 463 (2017).

15. Sudlow, C., Gallacher, J., Allen, N., Beral, V., Burton, P., Danesh, J., Downey, P., Elliott, P., Green, J., Landray, M., *et al.* UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS medicine* **12** (2015).

16. Stephens, Z. D., Lee, S. Y., Faghri, F., Campbell, R. H., Zhai, C., Efron, M. J., Iyer, R., Schatz, M. C., Sinha, S. & Robinson, G. E. Big data: astronomical or genomical? *PLoS biology* **13**, e1002195 (2015).

17. Leung, M. K., Delong, A., Alipanahi, B. & Frey, B. J. Machine learning in genomic medicine: a review of computational problems and data sets. *Proceedings of the IEEE* **104**, 176 (2015).

18. Ashley, E. A. Towards precision medicine. *Nature Reviews Genetics* **17**, 507 (2016).

19. Poste, G. Bring on the biomarkers. *Nature* **469**, 156 (2011).

20. Cordell, H. J. Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. *Human molecular genetics* **11**, 2463 (2002).

21. Fisher, R. A. XV.—The correlation between relatives on the supposition of Mendelian inheritance. *Earth and Environmental Science Transactions of the Royal Society of Edinburgh* **52**, 399 (1919).

22. Listgarten, J., Lippert, C., Kadie, C. M., Davidson, R. I., Eskin, E. & Heckerman, D. Improved linear mixed models for genome-wide association studies. *Nature methods* **9**, 525 (2012).

23. Hastie, T., Tibshirani, R. & Friedman, J. *The elements of statistical learning: data mining, inference, and prediction* (Springer Science & Business Media, 2009).

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

25. Zuk, O., Hechter, E., Sunyaev, S. R. & Lander, E. S. The mystery of missing heritability: Genetic interactions create phantom heritability. *Proceedings of the National Academy of Sciences* **109**, 1193 (2012).

26. Saeys, Y., Inza, I. & Larrañaga, P. A review of feature selection techniques in bioinformatics. *Bioinformatics* **23**, 2507 (2007).

27. Costanzo, M., Kuzmin, E., van Leeuwen, J., Mair, B., Moffat, J., Boone, C. & Andrews, B. Global genetic networks and the genotype-to-phenotype relationship. *Cell* **177**, 85 (2019).

28. Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A. & Reich, D. Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics* **38**, 904 (2006).

29. Lippert, C. Linear mixed models for genome-wide association studies. *Eberhard Karls Universität Tübingen* (2013).

30. Bogard, N., Linder, J., Rosenberg, A. B. & Seelig, G. A deep neural network for predicting and engineering alternative polyadenylation. *Cell* **178**, 91 (2019).

31. De Boer, C. G., Vaishnav, E. D., Sadeh, R., Abeyta, E. L., Friedman, N. & Regev, A. Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. *Nature Biotechnology* **38**, 56 (2020).

32. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436 (2015).

33. Franke, A., McGovern, D. P., Barrett, J. C., Wang, K., Radford-Smith, G. L., Ahmad, T., Lees, C. W., Balschun, T., Lee, J., Roberts, R., *et al.* Genome-wide meta-analysis increases to 71 the number of confirmed Crohn's disease susceptibility loci. *Nature genetics* **42**, 1118 (2010).

34. Anderson, C. A., Boucher, G., Lees, C. W., Franke, A., D'Amato, M., Taylor, K. D., Lee, J. C., Goyette, P., Imielinski, M., Latiano, A., *et al.* Meta-analysis identifies 29 additional ulcerative colitis risk loci, increasing the number of confirmed associations to 47. *Nature genetics* **43**, 246 (2011).

35. McCarthy, M. I. Genomics, type 2 diabetes, and obesity. *New England Journal of Medicine* **363**, 2339 (2010).

36. Teslovich, T. M., Musunuru, K., Smith, A. V., Edmondson, A. C., Stylianou, I. M., Koseki, M., Pirruccello, J. P., Ripatti, S., Chasman, D. I., Willer, C. J., *et al.* Biological, clinical and population relevance of 95 loci for blood lipids. *Nature* **466**, 707 (2010).

37. Craddock, N. J., Jones, I. R., *et al.* Wellcome Trust Case Control Consortium. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature* **447**, 661 (2007).

38. MacArthur, J., Bowler, E., Cerezo, M., Gil, L., Hall, P., Hastings, E., Junkins, H., McMahon, A., Milano, A., Morales, J., *et al.* The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog). *Nucleic acids research* **45**, D896 (2016).

39. LaFramboise, T. Single nucleotide polymorphism arrays: a decade of biological, computational and technological advances. *Nucleic acids research* **37**, 4181 (2009).

40. Pearson, K. X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **50**, 157 (1900).

41. Fisher, R. A. On the interpretation of χ^2 from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society* **85**, 87 (1922).

42. Koch, K.-R. *Parameter estimation and hypothesis testing in linear models* (Springer Science & Business Media, 1999).

43. Cochran, W. G. Some methods for strengthening the common χ^2 tests. *Biometrics* **10**, 417 (1954).

44. Mantel, N. & Haenszel, W. Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of the national cancer institute* **22**, 719 (1959).

45. Marchini, J., Donnelly, P. & Cardon, L. R. Genome-wide strategies for detecting multiple loci that influence complex diseases. *Nature genetics* **37**, 413 (2005).

46. Niel, C., Sinoquet, C., Dina, C. & Rocheleau, G. A survey about methods dedicated to epistasis detection. *Frontiers in genetics* **6**, 285 (2015).

47. Goudey, B., Rawlinson, D., Wang, Q., Shi, F., Ferra, H., Campbell, R. M., Stern, L., Inouye, M. T., Ong, C. S. & Kowalczyk, A. GWIS-model-free, fast and exhaustive search for epistatic interactions in case-control GWAS. *BMC genomics* **14**, S10 (2013).

48. Llinares-López, F., Grimm, D. G., Bodenham, D. A., Gieraths, U., Sugiyama, M., Rowan, B. & Borgwardt, K. Genome-wide detection of intervals of genetic heterogeneity associated with complex traits. *Bioinformatics* **31**, i240 (2015).

49. Shaffer, J. P. Multiple hypothesis testing. *Annual review of psychology* **46**, 561 (1995).

50. Noble, W. S. How does multiple testing correction work? *Nature biotechnology* **27**, 1135 (2009).

51. Bonferroni, C. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze* **8**, 3 (1936).

52. Llinares-Lopez, F. & Borgwardt, K. 8 Machine Learning for Biomarker Discovery: Significant Pattern Mining. *Analyzing Network Data in Biology and Medicine: An Interdisciplinary Textbook for Biological, Medical and Computational Scientists*, 313 (2019).

53. Llinares López, F. *Significant Pattern Mining for Biomarker Discovery* PhD thesis (ETH Zurich, 2018).

54. Terada, A., Okada-Hatakeyama, M., Tsuda, K. & Sese, J. Statistical significance of combinatorial regulations. *Proceedings of the National Academy of Sciences* **110**, 12996 (2013).

55. Minato, S.-i., Uno, T., Tsuda, K., Terada, A. & Sese, J. *A fast method of statistical assessment for combinatorial hypotheses based on frequent itemset enumeration* in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2014), 422.

56. Papaxanthos, L., Llinares-López, F., Bodenham, D. & Borgwardt, K. *Finding significant combinations of features in the presence of categorical covariates* in *Advances in neural information processing systems* (2016), 2279.

57. Llinares-López, F., Papaxanthos, L., Bodenham, D., Roqueiro, D., Investigators, C. & Borgwardt, K. Genome-wide genetic heterogeneity discovery with categorical covariates. *Bioinformatics* **33**, 1820 (2017).

58. Llinares-López, F., Papaxanthos, L., Roqueiro, D., Bodenham, D. & Borgwardt, K. CASMAP: detection of statistically significant combinations of SNPs in association mapping. *Bioinformatics* **35**, 2680 (2019).

59. McClellan, J. & King, M.-C. Genetic heterogeneity in human disease. *Cell* **141**, 210 (2010).

60. Burrell, R. A., McGranahan, N., Bartek, J. & Swanton, C. The causes and consequences of genetic heterogeneity in cancer evolution. *Nature* **501**, 338 (2013).

61. Lee, S., Abecasis, G. R., Boehnke, M. & Lin, X. Rare-variant association analysis: study designs and statistical tests. *The American Journal of Human Genetics* **95**, 5 (2014).

62. Cordell, H. J. Detecting gene–gene interactions that underlie human diseases. *Nature Reviews Genetics* **10**, 392 (2009).

63. Kam-Thong, T., Azencott, C.-A., Cayton, L., Pütz, B., Altmann, A., Karbalai, N., Sämann, P. G., Schölkopf, B., Müller-Myhsok, B. & Borgwardt, K. M. GLIDE: GPU-based linear regression for detection of epistasis. *Human heredity* **73**, 220 (2012).

64. Kam-Thong, T., Czamara, D., Tsuda, K., Borgwardt, K., Lewis, C. M., Erhardt-Lehmann, A., Hemmer, B., Rieckmann, P., Daake, M., Weber, F., *et al.* EPIBLASTER-fast exhaustive two-locus epistasis detection strategy using graphical processing units. *European Journal of Human Genetics* **19**, 465 (2011).

65. Kam-Thong, T., Pütz, B., Karbalai, N., Müller- Myhsok, B. & Borgwardt, K. Epistasis detection on quantitative phenotypes by exhaustive enumeration using GPUs. *Bioinformatics* **27**, i214 (2011).

66. Ritchie, M. D., Hahn, L. W., Roodi, N., Bailey, L. R., Dupont, W. D., Parl, F. F. & Moore, J. H. Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *The American Journal of Human Genetics* **69**, 138 (2001).

67. Cattaert, T., Calle, M. L., Dudek, S. M., Mahachie John, J. M., Van Lishout, F., Urrea, V., Ritchie, M. D. & Van Steen, K. Model-Based Multifactor Dimensionality Reduction for detecting epistasis in case–control data in the presence of noise. *Annals of human genetics* **75**, 78 (2011).

68. Van Lishout, F., Gadaleta, F., Moore, J. H., Wehenkel, L. & Van Steen, K. gammaMAXT: a fast multiple-testing correction algorithm. *BioData mining* **8**, 36 (2015).

69. Wan, X., Yang, C., Yang, Q., Xue, H., Fan, X., Tang, N. L. & Yu, W. BOOST: A fast approach to detecting gene-gene interactions in genome-wide case-control studies. *The American Journal of Human Genetics* **87**, 325 (2010).

70. Yung, L. S., Yang, C., Wan, X. & Yu, W. GBOOST: a GPU-based tool for detecting gene–gene interactions in genome–wide case control studies. *Bioinformatics* **27**, 1309 (2011).

71. Grady, B. J., Torstenson, E., Dudek, S. M., Giles, J., Sexton, D. & Ritchie, M. D. Finding unique filter sets in PLATO: a precursor to efficient interaction analysis in GWAS data. *Biocomputing*, 315 (2010).

72. Greene, C. S., Penrod, N. M., Kiralis, J. & Moore, J. H. Spatially uniform relieff (SURF) for computationally-efficient filtering of gene-gene interactions. *BioData mining* **2**, 5 (2009).

73. Greene, C. S., Himmelstein, D. S., Kiralis, J. & Moore, J. H. *The informative extremes: using both nearest and farthest individuals can improve relief algorithms in the domain of human genetics* in *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics* (2010), 182.

74. Wang, M. H., Sun, R., Guo, J., Weng, H., Lee, J., Hu, I., Sham, P. C. & Zee, B. C.-Y. A fast and powerful W-test for pairwise epistasis testing. *Nucleic acids research* **44**, e115 (2016).

75. Emily, M., Mailund, T., Hein, J., Schauser, L. & Schierup, M. H. Using biological networks to search for interacting loci in genome-wide association studies. *European Journal of Human Genetics* **17**, 1231 (2009).

76. Pendergrass, S. A., Verma, S. S., Holzinger, E. R., Moore, C. B., Wallace, J., Dudek, S. M., Huggins, W., Kitchner, T., Waudby, C., Berg, R., *et al.* Next-generation analysis of cataracts: determining knowledge driven gene-gene interactions using Biofilter, and gene-environment interactions using the PhenX Toolkit. *Biocomputing*, 147 (2013).

77. Yang, C., He, Z., Wan, X., Yang, Q., Xue, H. & Yu, W. SNPHarvester: a filtering-based approach for detecting epistatic interactions in genome-wide association studies. *Bioinformatics* **25**, 504 (2009).

78. Zhang, X., Zou, F. & Wang, W. *Fastanova: an efficient algorithm for genome-wide association study* in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (2008), 821.

79. Zhang, X., Pan, F., Xie, Y., Zou, F. & Wang, W. *COE: a general approach for efficient genome-wide two-locus epistasis test in disease association study* in *Annual International Conference on Research in Computational Molecular Biology* (2009), 253.

80. Zhang, X., Huang, S., Zou, F. & Wang, W. TEAM: efficient two-locus epistasis tests in human genome-wide association study. *Bioinformatics* **26**, i217 (2010).

81. Zhang, Y. & Liu, J. S. Bayesian inference of epistatic interactions in case-control studies. *Nature genetics* **39**, 1167 (2007).

82. Han, B., Chen, X.-w., Talebizadeh, Z. & Xu, H. Genetic studies of complex human diseases: characterizing SNP-disease associations using Bayesian networks. *BMC systems biology* **6**, S14 (2012).

83. Alekseyenko, A. V., Lytkin, N. I., Ai, J., Ding, B., Padyukov, L., Aliferis, C. F. & Statnikov, A. Causal graph-based analysis of genome-wide association data in rheumatoid arthritis. *Biology direct* **6**, 25 (2011).

84. Yanlan, L. & Jiawei, L. An improved markov blanket approach to detect SNPs-Disease Associations in case-control studies. *International Journal of Digital Content Technology and its Applications* **6** (2012).

85. Statnikov, A., Lytkin, N. I., Lemeire, J. & Aliferis, C. F. Algorithms for discovery of multiple Markov boundaries. *Journal of Machine Learning Research* **14**, 499 (2013).

86. Achlioptas, P., Schölkopf, B. & Borgwardt, K. *Two-locus association mapping in subquadratic time* in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* (2011), 726.

87. Paturi, R., Rajasekaran, S. & Reif, J. *The light bulb problem* in *Proceedings of the second annual workshop on Computational learning theory* (1989), 261.

88. Schwarz, D. F., König, I. R. & Ziegler, A. On safari to Random Jungle: a fast implementation of Random Forests for high-dimensional data. *Bioinformatics* **26**, 1752 (2010).

89. Botta, V., Louppe, G., Geurts, P. & Wehenkel, L. Exploiting SNP correlations within random forest for genome-wide association studies. *PloS one* **9** (2014).

90. Wei, C. & Lu, Q. GWGGI: software for genome-wide gene-gene interaction analysis. *BMC genetics* **15**, 101 (2014).

91. Van de Haar, J., Canisius, S., Michael, K. Y., Voest, E. E., Wessels, L. F. & Ideker, T. Identifying epistasis in cancer genomes: a delicate affair. *Cell* **177**, 1375 (2019).

92. Szczurek, E. & Beerenwinkel, N. *Modeling mutual exclusivity of cancer mutations* in *International Conference on Research in Computational Molecular Biology* (2014), 307.

93. Wappett, M., Dulak, A., Yang, Z. R., Al-Watban, A., Bradford, J. R. & Dry, J. R. Multi-omic measurement of mutually exclusive loss-of-function enriches for candidate synthetic lethal gene pairs. *BMC genomics* **17**, 65 (2016).

94. Kuipers, J., Thurnherr, T., Moffa, G., Suter, P., Behr, J., Goosen, R., Christofori, G. & Beerenwinkel, N. Mutational interactions define novel cancer subgroups. *Nature communications* **9**, 1 (2018).

95. Lu, X., Megchelenbrink, W., Notebaart, R. A. & Huynen, M. A. Predicting human genetic interactions from cancer genome evolution. *PloS one* **10** (2015).

96. Rauscher, B., Heigwer, F., Henkel, L., Hielscher, T., Voloshanenko, O. & Boutros, M. Toward an integrated map of genetic interactions in cancer cells. *Molecular systems biology* **14** (2018).

97. Matlak, D. & Szczurek, E. Epistasis in genomic and survival data of cancer patients. *PLoS computational biology* **13**, e1005626 (2017).

98. Hemani, G., Shakhbazov, K., Westra, H.-J., Esko, T., Henders, A. K., McRae, A. F., Yang, J., Gibson, G., Martin, N. G., Metspalu, A., *et al.* Detection and replication of epistasis influencing transcription in humans. *Nature* **508**, 249 (2014).

99. Phillips, P. C. Epistasis—the essential role of gene interactions in the structure and evolution of genetic systems. *Nature Reviews Genetics* **9**, 855 (2008).

100. Dudoit, S., Shaffer, J. P., Boldrick, J. C., *et al.* Multiple hypothesis testing in microarray experiments. *Statistical Science* **18**, 71 (2003).

101. Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* **57**, 289 (1995).

102. Webb, G. I. *Discovering significant rules* in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (2006), 434.

103. Webb, G. I. Discovering significant patterns. *Machine learning* **68**, 1 (2007).

104. Webb, G. I. Layered critical values: a powerful direct-adjustment approach to discovering significant patterns. *Machine Learning* **71**, 307 (2008).

105. Tarone, R. E. A modified Bonferroni method for discrete data. *Biometrics*, 515 (1990).

106. Llinares-López, F., Sugiyama, M., Papaxanthos, L. & Borgwardt, K. *Fast and memory-efficient significant pattern mining via permutation testing* in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* (2015), 725.

107. Vilhjálmsson, B. J. & Nordborg, M. The nature of confounding in genome-wide association studies. *Nature Reviews Genetics* **14**, 1 (2013).

108. Devlin, B. & Roeder, K. Genomic control for association studies. *Biometrics* **55**, 997 (1999).

109. Rakitsch, B., Lippert, C., Stegle, O. & Borgwardt, K. A Lasso multi-marker mixed model for association mapping with population structure correction. *Bioinformatics* **29**, 206 (2012).

110. Listgarten, J., Kadie, C., Schadt, E. E. & Heckerman, D. Correction for hidden confounders in the genetic analysis of gene expression. *Proceedings of the National Academy of Sciences* **107**, 16465 (2010).

111. Loh, P.-R., Kichaev, G., Gazal, S., Schoech, A. P. & Price, A. L. Mixed-model association for biobank-scale datasets. *Nature genetics* **50**, 906 (2018).

112. Sugiyama, M., López, F. L., Kasenburg, N. & Borgwardt, K. Mining significant subgraphs with multiple testing correction. *SIAM Data Mining (SDM)* (2015).

113. Ogihara, Z. P., Zaki, M., Parthasarathy, S., Ogihara, M. & Li, W. *New algorithms for fast discovery of association rules* in *3rd Intl. Conf. on Knowledge Discovery and Data Mining* (1997).

114. Borgelt, C. Frequent item set mining. *Wiley interdisciplinary reviews: data mining and knowledge discovery* **2**, 437 (2012).

115. Zaki, M. J. & Gouda, K. *Fast vertical mining using diffsets* in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (2003), 326.

116. Leisch, F., Weingessel, A. & Hornik, K. On the generation of correlated artificial binary data (1998).

117. Atwell, S., Huang, Y. S., Vilhjálmsson, B. J., Willems, G., Horton, M., Li, Y., Meng, D., Platt, A., Tarone, A. M., Hu, T. T., *et al.* Genome-wide association study of 107 phenotypes in Arabidopsis thaliana inbred lines. *Nature* **465**, 627 (2010).

118. Grimm, D. G., Roqueiro, D., Salomé, P. A., Kleeberger, S., Greshake, B., Zhu, W., Liu, C., Lippert, C., Stegle, O., Schölkopf, B., *et al.* easyGWAS: a cloud-based platform for comparing the results of genome-wide association studies. *The Plant Cell* **29**, 5 (2017).

119. Novembre, J., Johnson, T., Bryc, K., Kutalik, Z., Boyko, A. R., Auton, A., Indap, A., King, K. S., Bergmann, S., Nelson, M. R., *et al.* Genes mirror geography within Europe. *Nature* **456**, 98 (2008).

120. Schmid, K. & Yang, Z. The trouble with sliding windows and the selective pressure in BRCA1. *PloS one* **3**, e3746 (2008).

121. Casale, F. P., Rakitsch, B., Lippert, C. & Stegle, O. Efficient set tests for the genetic analysis of correlated traits. *Nature methods* **12**, 755 (2015).

122. Regan, E. A., Hokanson, J. E., Murphy, J. R., Make, B., Lynch, D. A., Beaty, T. H., Curran-Everett, D., Silverman, E. K. & Crapo, J. D. Genetic epidemiology of COPD (COPDGene) study design. *COPD: Journal of Chronic Obstructive Pulmonary Disease* **7**, 32 (2011).

123. Cho, M. H., McDonald, M.-L. N., Zhou, X., Mattheisen, M., Castaldi, P. J., Hersh, C. P., DeMeo, D. L., Sylvia, J. S., Ziniti, J., Laird, N. M., *et al.* Risk loci for chronic obstructive pulmonary disease: a genome-wide association study and meta-analysis. *The lancet Respiratory medicine* **2**, 214 (2014).

124. Cho, M. H., Boutaoui, N., Klanderman, B. J., Sylvia, J. S., Ziniti, J. P., Hersh, C. P., DeMeo, D. L., Hunninghake, G. M., Litonjua, A. A., Sparrow, D., *et al.* Variants in FAM13A are associated with chronic obstructive pulmonary disease. *Nature genetics* **42**, 200 (2010).

125. Yoshizoe, K., Terada, A. & Tsuda, K. MP-LAMP: parallel detection of statistically significant multi-loci markers on cloud platforms. *Bioinformatics* **34**, 3047 (2018).

126. Valeri, J. A., Collins, K. M., Lepe, B. A., Lu, T. K. & Camacho, D. M. Sequence-to-function deep learning frameworks for synthetic biology. *bioRxiv*, 870055 (2019).

127. Atwal, G. S. & Kinney, J. B. Learning quantitative sequence–function relationships from massively parallel experiments. *Journal of Statistical Physics* **162**, 1203 (2016).

128. Salis, H. M. in *Methods in enzymology* 19 (Elsevier, 2011).

129. Jeschek, M., Gerngross, D. & Panke, S. Rationally reduced libraries for combinatorial pathway optimization minimizing experimental effort. *Nature communications* **7**, 11163 (2016).

130. Bonde, M. T., Pedersen, M., Klausen, M. S., Jensen, S. I., Wulff, T., Harrison, S., Nielsen, A. T., Herrgård, M. J. & Sommer, M. O. Predictable tuning of protein expression in bacteria. *Nature methods* **13**, 233 (2016).

131. Sample, P. J., Wang, B., Reid, D. W., Presnyak, V., McFadyen, I. J., Morris, D. R. & Seelig, G. Human 5′ UTR design and variant effect prediction from a massively parallel translation assay. *Nature biotechnology* **37**, 803 (2019).

132. Cuperus, J. T., Groves, B., Kuchina, A., Rosenberg, A. B., Jojic, N., Fields, S. & Seelig, G. Deep learning of the regulatory grammar of yeast 5′ untranslated regions from 500,000 random sequences. *Genome research* **27**, 2015 (2017).

133. Hoellerer, S., Papaxanthos, L., Gumpinger, A. C., Fischer, K., Beisel, C., Borgwardt, K., Benenson, Y. & Jeschek, M. Large-scale DNA-based phenotypic recording and deep learning enable highly accurate sequence-function mapping. *Nature Communications* **11**, 3551 (2020).

134. Kosuri, S., Goodman, D. B., Cambray, G., Mutalik, V. K., Gao, Y., Arkin, A. P., Endy, D. & Church, G. M. Composability of regulatory sequences controlling transcription and translation in Escherichia coli. *Proceedings of the National Academy of Sciences* **110**, 14024 (2013).

135. Yus, E., Yang, J.-S., Sogues, A. & Serrano, L. A reporter system coupled with high-throughput sequencing unveils key bacterial transcription and translation determinants. *Nature communications* **8**, 368 (2017).

136. Cambray, G., Guimaraes, J. C. & Arkin, A. P. Evaluation of 244,000 synthetic sequences reveals design principles to optimize translation in Escherichia coli. *Nature biotechnology* **36**, 1005 (2018).

137. Park, Y. & Kellis, M. Deep learning for regulatory genomics. *Nature biotechnology* **33**, 825 (2015).

138. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology* **33**, 831 (2015).

139. Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature methods* **12**, 931 (2015).

140. Zou, J., Huss, M., Abid, A., Mohammadi, P., Torkamani, A. & Telenti, A. A primer on deep learning in genomics. *Nature genetics* **51**, 12 (2019).

141. Senior, A., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Zidek, A., Nelson, A., Bridgland, A., *et al.* Improved protein structure prediction using potentials from deep learning. *Nature* (2020).

142. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nature methods* **15**, 1053 (2018).

143. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. & Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural computation* **1**, 541 (1989).

144. Pérez-Cruz, F. *Estimation of information theoretic measures for continuous random variables* in *Advances in neural information processing systems* (2009), 1257.

145. Zeng, H., Edwards, M. D., Liu, G. & Gifford, D. K. Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics* **32**, i121 (2016).

146. Kelley, D. R., Reshef, Y. A., Bileschi, M., Belanger, D., McLean, C. Y. & Snoek, J. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome research* **28**, 739 (2018).

147. He, K., Zhang, X., Ren, S. & Sun, J. *Deep residual learning for image recognition* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 770.

148. Xie, S., Girshick, R., Dollár, P., Tu, Z. & He, K. *Aggregated residual transformations for deep neural networks* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 1492.

149. Lakshminarayanan, B., Pritzel, A. & Blundell, C. *Simple and scalable predictive uncertainty estimation using deep ensembles* in *Advances in neural information processing systems* (2017), 6402.

150. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

151. Maas, A. L., Hannun, A. Y. & Ng, A. Y. *Rectifier nonlinearities improve neural network acoustic models* in *Proceedings of the 30th International Conference on Machine Learning* **30** (2013), 3.

152. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)* **9** (2015).

153. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., *et al. Tensorflow: A system for large-scale machine learning* in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* (2016), 265.

154. Bergstra, J. & Bengio, Y. Random search for hyper-parameter optimization. *Journal of machine learning research* **13**, 281 (2012).

155. Altman, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician* **46**, 175 (1992).

156. Breiman, L. Random forests. *Machine learning* **45**, 5 (2001).

157. Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189 (2001).

158. Sundararajan, M., Taly, A. & Yan, Q. *Axiomatic attribution for deep networks* in *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017), 3319.

159. Friedman, J., Hastie, T. & Tibshirani, R. *The elements of statistical learning* **10** (Springer series in statistics New York, 2001).

160. Sugiyama, M. & Borgwardt, K. *Finding Statistically Significant Interactions between Continuous Features* in *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019)* (2019), 3490.

161. Woolf, B. The log likelihood ratio test (the G-test). *Annals of human genetics* **21**, 397 (1957).

162. Tatti, N. *Itemsets for real-valued datasets* in *IEEE 13th International Conference on Data Mining* (2013), 717.

163. Jaillard, M., Lima, L., Tournoud, M., Mahé, P., Van Belkum, A., Lacroix, V. & Jacob, L. A fast and agnostic method for bacterial genome-wide association studies: Bridging the gap between k-mers and genetic events. *PLoS genetics* **14**, e1007758 (2018).

164. Jaillard, M., Tournoud, M., Lima, L., Lacroix, V., Veyrieras, J.-B. & Jacob, L. Representing genetic determinants in bacterial GWAS with compacted De Bruijn graphs. *bioRxiv*, 113563 (2017).

165. Pellegrina, L., Riondato, M. & Vandin, F. *SPuManTE: Significant Pattern Mining with Unconditional Testing* in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019).

166. Benjamini, Y. & Yekutieli, D. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, 1165 (2001).

167. Gilbert, P. B. A modified false discovery rate multiple-comparisons procedure for discrete data, applied to human immunodeficiency virus genetics. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **54**, 143 (2005).

168. Komiyama, J., Ishihata, M., Arimura, H., Nishibayashi, T. & Minato, S.-i. *Statistical emerging pattern mining with multiple testing correction* in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2017), 897.

169. Candes, E., Fan, Y., Janson, L. & Lv, J. Panning for gold:'model-X'knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **80**, 551 (2018).

170. Li, C. & Li, H. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics* **24**, 1175 (2008).

171. Romano, Y., Sesia, M. & Candès, E. Deep knockoffs. *Journal of the American Statistical Association*, 1 (2019).

172. Jordon, J., Yoon, J. & van der Schaar, M. *KnockoffGAN: Generating Knockoffs for Feature Selection using Generative Adversarial Networks* in *International Conference on Learning Representations* (2019).

173. Sesia, M., Sabatti, C. & Candès, E. J. Gene hunting with hidden Markov model knockoffs. *Biometrika* **106**, 1 (2019).

174. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H. & Schmidhuber, J. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence* **31**, 855 (2008).

175. Gruber, A. R., Lorenz, R., Bernhart, S. H., Neuböck, R. & Hofacker, I. L. The vienna RNA websuite. *Nucleic acids research* **36**, W70 (2008).

176. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. *Generative adversarial nets* in *Advances in neural information processing systems* (2014), 2672.

177. Doersch, C. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* (2016).

178. Brookes, D., Park, H. & Listgarten, J. *Conditioning by adaptive sampling for robust design* in *Proceedings of the 36th International Conference on Machine Learning* **97** (2019), 773.

179. Killoran, N., Lee, L. J., Delong, A., Duvenaud, D. & Frey, B. J. Generating and designing DNA with deep generative models. *arXiv preprint arXiv:1712.06148* (2017).

180. Miyato, T. & Koyama, M. cGANs with projection discriminator. *arXiv preprint arXiv:1802.05637* (2018).

181. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X. & Metaxas, D. N. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence* **41**, 1947 (2018).

182. Avsec, Ž., Weilert, M., Shrikumar, A., Alexandari, A., Krueger, S., Dalal, K., Fropf, R., McAnany, C., Gagneur, J., Kundaje, A., *et al.* Deep learning at base-resolution reveals motif syntax of the cis-regulatory code. *bioRxiv*, 737981 (2019).

183. Bahdanau, D., Cho, K. & Bengio, Y. *Neural machine translation by jointly learning to align and translate* in *3rd International Conference on Learning Representations* (2015).