

DISS. ETH NO. 25072

SIGNIFICANT PATTERN MINING
FOR BIOMARKER DISCOVERY

A dissertation submitted to
ETH ZURICH

for the degree of
DOCTOR OF SCIENCES

presented by
FELIPE LLINARES LÓPEZ
Ingeniero de Telecomunicación (Universidad Carlos III de Madrid)
born 4 December 1989
citizen of Spain

accepted on the recommendation of
Prof. Dr. Karsten Borgwardt, examiner
Prof. Dr. Koji Tsuda, co-examiner
Prof. Dr. Volker Roth, co-examiner

2018

To my parents, María Jesús and Felipe.
Without you, this would not have been possible.

ABSTRACT

BACKGROUND: In recent years, far-reaching technological improvements have vastly enhanced our ability to gather large amounts of molecular and clinical data. This wealth of information has been poised to revolutionise computational biology and medicine. However, reliably and efficiently extracting knowledge from data in these domains is often a difficult task, as datasets tend to be characterised by very low sample sizes relative to the number of features and small signal-to-noise ratios. In this regime, where associations between features and targets tend to be faint, methods based on classical statistical significance testing have proven to be a powerful tool for data exploration, allowing to detect salient patterns in the data that can be prioritised for further study by domain experts. However, owing to the inherent challenges posed by statistical inference in high-dimensional spaces, most existing methods either rely on univariate statistical association tests, thus considering the effect of each feature in isolation from the rest, or utilise sparse linear models, describing the joint effect of all features as the sum of a small number of individual effects. As a result, *these approaches are unable to detect nonlinear signals due to interactions between features*. This shortcoming has profound implications in many crucial problems including, but not limited to, accounting for epistasis in genome-wide association studies, modelling tissue-specific combinatorial transcription factor regulation of gene expression or discovering patterns of co-occurring mutational events in tumours.

Assessing the statistical significance of *all* high-order interactions between features is an exceedingly challenging problem, mainly due to two major difficulties: (i) the vast number of statistical association tests to be performed would cause an extreme multiple comparisons problem that goes well beyond what classical tools such as the Bonferroni correction are able to cope with in practice, and (ii) the computational complexity of a naive approach would grow exponentially with the number of features. Nonetheless, despite being long considered an unsolvable problem by many, recent work has provided a solution for the particular case that all features are discrete, *de facto* kickstarting the field of *significant pattern mining*, the subject of study of this thesis.

CONTRIBUTIONS: Significant pattern mining is a young field, offering a myriad of open problems, some of which severely hinder its applicability to analyse data in computational biology and medicine. *The goal of this thesis is to develop novel significant pattern mining algorithms, aiming to overcome some of the most crucial limitations of the state of the art.*

The first part of this thesis provides a self-contained introduction to significant pattern mining, proposing a general formulation that encompasses multiple variations of the problem and describing the statistical and algorithmic techniques that make significant pattern mining possible.

Next, original contributions on three different topics of fundamental importance for applications in the life sciences are presented.

(i) Many associations between features and targets in biomedical data are weak, being barely above the noise level. This difficulty is exacerbated when high-order interactions between features are taken into account, requiring a stringent significance threshold. However, the search space consisting of all feature interactions is redundant by construction, inducing strong statistical dependencies between test statistics. In practice, these reduce the effective number of tests to account for when correcting for the multiple comparisons problem. *One of the contributions in this thesis is a fast and memory-efficient algorithm that combines significant pattern mining with permutation testing to estimate a less stringent significance threshold that accounts for the dependencies between test statistics.* Compared to the single existing approach that also aims to exploit this phenomenon, our method is one to three orders of magnitude faster and requires two to three orders of magnitude less memory while providing the same improvement in statistical power.

(ii) Another ubiquitous problem when exploring data in the life sciences is the need to correct for covariate factors such as age, gender, socioeconomic status or population structure. Neglecting to account for factors of variation that might have a potentially confounding effect could create a large number of spurious associations, jeopardising the reliability of any discoveries reported as a result of the analysis. One of the biggest limitations of state-of-the-art significant pattern mining algorithms is their inability to incorporate such covariate factors into the model. *A second contribution of this thesis solves this shortcoming by devising a novel method that uses the Cochran-Mantel-Haenszel test to correct for a categorical covariate.* Computational tractability is achieved by means of a specialised pruning criterion that can be evaluated in almost-linear time in the number of categories of the covariate. Results on both synthetic data and genome-wide association studies of the plant model organism *A. thaliana* suggest a drastic reduction in false positives due to confounding effects without sacrificing neither statistical power nor computational efficiency.

(iii) Genetic heterogeneity, the phenomenon that multiple genomic markers might affect a phenotype in a similar manner, can be exploited to gain statistical power in genome-wide association studies. By carrying out association studies at a region level, as opposed to testing single markers, the (possibly weak) effects of multiple neighbouring markers can be aggregated into a stronger, easier to detect signal. A crucial limitation of existing approaches in this domain is their requirement that the user preselects *a priori* a small subset of candidate genomic regions. *The final contribution discussed in this thesis is a new family of methods to carry out genome-wide association studies at a region level based on significant pattern mining.* Unlike other approaches, these methods are able to test *all* genomic regions, regardless of their length or position. The resulting resilience to misspecification of the subset of genomic regions to be tested empirically translates into sharp gains in statistical power in situations for which reliable prior knowledge about the length or location of the causal regions is unavailable. This was corroborated by experiments on synthetic data as well as genome-wide association studies on human and *A. thaliana* samples.

OUTLOOK: We strongly believe that these contributions substantially strengthen the state of the art in significant pattern mining, particularly in regards to potential applications in computational biology and medicine. Remarkable challenges still

lie ahead of this young field, as discussed extensively in the final part of this thesis. Nevertheless, rapid progress in the development of methods at the intersection between machine learning, statistical significance testing and data mining suggest that significant pattern mining will play a key role in knowledge discovery for years to come.

RÉSUMÉ

CONTEXTE : Les progrès technologiques de ces dernières années ont décuplé notre capacité de récolter de grandes quantités de données moléculaires et cliniques. Cette richesse d'information est destinée à révolutionner la biologie computationnelle et la médecine. Cependant, il est souvent difficile d'extraire des connaissances pertinentes à partir de ces données car elles présentent des tailles d'échantillon très petites par rapport au nombre de leurs caractéristiques et ont de petits rapport signal sur bruit. Dans ce cadre de faibles associations entre caractéristiques et objectifs, les méthodes basées sur les tests statistiques classiques ont été largement utilisées pour l'exploration des données, permettant de découvrir des motifs remarquables qui ont donné suite à des études plus approfondies des effets observés. Néanmoins, à cause des difficultés causées par l'inférence statistique dans les espaces à haute dimension, la plupart des méthodes statistiques à disposition sont basées soit sur des tests univariés - qui ne considèrent que les effets de chaque caractéristique en isolation des autres - soit sur des modèles linéaires creux - qui décrivent l'effet combiné de plusieurs variables comme la somme d'un petit nombre d'effets individuels. Par conséquent, *ces approches ne sont pas aptes à détecter des signaux non-linéaires causés par l'interaction de plusieurs caractéristiques*. Cette limitation a des conséquences marquées pour de nombreuses applications, notamment pour la détection d'épistasie dans les études d'association pangénomiques, la modélisation de la régulation combinatoire de l'expression génétique par les facteurs de transcriptions ou encore la découverte de mutations co-occurentes dans les tumeurs.

Déterminer la signification statistique de *toutes* les interactions d'ordre supérieur entre caractéristiques est un problème excessivement difficile pour deux raisons : (i) l'immense quantité d'hypothèses qui doivent être testées cause un problème de comparaison multiple extrême, qui ne peut être résolu par de simples instruments tels que la correction de Bonferroni et (ii) la complexité computationnelle d'une approche naïve grandit exponentiellement avec le nombre de caractéristiques. Toutefois, malgré avoir été longtemps considéré comme un problème insoluble par de nombreux chercheurs, des développements récents ont proposé des solutions pour le cas particulier où les caractéristiques sont discrètes, donnant naissance au domaine de la *découverte de motifs significatifs*, le sujet de recherche de cette thèse.

CONTRIBUTIONS : La découverte de motifs significatifs est un domaine de recherche nouveau qui offre une myriade de problèmes irrésolus, parmi lesquels certains qui empêchent son application à l'analyse de données médicales ou biologiques. *Le but de cette thèse est de développer de nouveaux algorithmes de découverte de motifs significatifs dépassant les limites actuelles de l'état de la technique.*

La première partie de cette thèse propose une introduction au domaine de la découverte de motifs significatifs et est composée d'une formulation générale du problème ainsi que d'une description des techniques statistiques et algorithmiques qui rende la découverte de motifs significatifs possible.

Ensuite, trois contributions fondamentales pour l'application de ces méthodes dans les sciences de la vie sont présentées.

(i) De nombreuses associations entre caractéristiques et objectifs dans les données biomédicales sont faibles, à peine au dessus du niveau du bruit. Les difficultés de détection s'accroissent lorsque sont considérées les interactions d'ordre supérieur entre plusieurs caractéristiques, exigeant un seuil de signification plus strict. Cependant, l'espace de recherche constitué par toutes les interactions entre toutes les caractéristiques étant redondant par construction, de fortes dépendances statistiques sont présentes entre les différentes statistiques de test. En pratique, ces dépendances réduisent le nombre effectif de tests pour lesquels il faut corriger lors de problèmes de comparaisons multiples. *Une des contributions de cette thèse est un algorithme rapide et efficient en terme d'utilisation de mémoire qui combine la découverte de motifs significatifs avec le test de permutation pour estimer un seuil de signification moins strict qui tient en compte les dépendances entre les statistiques de test.* Notre méthode est d'un à trois ordres de magnitude plus rapide et demande entre deux et trois ordres de magnitude moins de mémoire que la seule autre solution existante, tout en permettant le même gain de puissance statistique.

(ii) Un autre problème omniprésent lors du traitement de données dans les sciences de la vie est le besoin de corriger pour différentes covariables telles que l'âge, le genre, le statut socioéconomique ou la structure d'une population. Ne pas considérer ces covariables lors des analyses statistiques peut nuire considérablement aux résultats obtenus, en générant de fausses associations et en compromettant leur analyse. Une des principales limitations des algorithmes existants de découverte de motifs significatifs est leur incapacité à considérer des covariables dans leur modèle. *Une deuxième contribution de cette thèse résout cette restriction en présentant une nouvelle méthode qui utilise le test de Cochran-Mantel-Haenszel pour corriger en tenant compte des covariables catégoriques.* Un critère d'élagage spécialisé qui peut être évalué en un temps quasi-linéaire par rapport au nombre de catégories de la covariable garantit la faisabilité computationnelle de l'algorithme. Les résultats obtenus sur des données synthétiques ainsi que sur des études d'association pangénomiques de l'organisme modèle *A. thaliana* suggèrent une réduction drastique du nombre de faux positifs dus aux effets des covariables sans perdre de puissance statistique ni d'efficacité computationnelle.

(iii) L'hétérogénéité génétique, un phénomène pour lequel plusieurs marqueurs génétiques peuvent influencer d'une manière similaire un seul phénotype, peut être utilisée pour gagner de la puissance statistique lors d'études d'association pangénomiques. Les effets (potentiellement faibles) de plusieurs marqueurs voisins peuvent être combinés dans un signal plus fort et plus facile à détecter en testant des régions entières plutôt que des marqueurs uniques. Une limitation cruciale des méthodes existantes dans ce domaine est qu'elles exigent de recevoir des régions prédéterminées à l'avance par l'utilisateur. *La dernière contribution décrite dans cette thèse est une nouvelle famille de méthodes basées sur la découverte de motifs significatifs qui permet l'étude d'association pangénomique au niveau des régions.* Ces méthodes, contrairement aux solutions existantes, permettent de tester toutes les régions génomiques, indépendamment de leur taille ou position. Ceci permet de considérables augmentations de puissance statistique dans les situations où très peu d'informations à propos des régions causales sont connues, comme l'ont prouvés nos expériences sur des données synthétiques et

sur des résultats d'études d'association pangénomiques d'échantillons humains et de *A. thaliana*.

PERSPECTIVES : Nous sommes convaincus que ces contributions renforcent considérablement l'état de la technique de la découverte de motifs significatifs, en particulier dans les applications pour la biologie computationnelle et la médecine. De nombreux défis sont encore présents dans ce jeune domaine, comme la partie finale de cette thèse le témoigne. Néanmoins, le progrès rapide dans le développement de méthodes à l'intersection de l'apprentissage machine, des tests statistiques significatifs et du data mining suggère que la découverte de motifs significatifs va jouer un rôle essentiel dans le domaine de l'extraction de connaissances lors des prochaines années.

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my advisor Prof. Karsten Borgwardt for his excellent advice and support throughout my PhD. I am deeply thankful for the exciting research opportunities, the generous funding to attend many of the main conferences in our field and the guidance and mentoring during my PhD studies. Working as a PhD student in Prof. Borgwardt's lab has been a fantastic learning experience and a great way to start what I hope will be a successful career as a machine learning researcher.

I would also like to thank Prof. Koji Tsuda and Prof. Volker Roth for kindly agreeing to act as referees in my doctoral examination committee, as well as Prof. Tanja Stadler for chairing the examination.

I cannot emphasise enough how grateful I am to all of my co-authors for their essential contributions in the four articles which form the backbone of this thesis. Laetitia Papaxanthos has co-authored three of these publications, and her contributions in two of them have been at least as important as mine. Her role in making the research described in this thesis a reality has been irreplaceable, the successful completion of these projects being to a great extent the result of her talent, knowledge and hard work. I am also deeply indebted to Mahito Sugiyama. Our collaboration had a decisive influence in all of my PhD's work but, most importantly, it was a source of inspiration and taught me a great deal about how to become a better machine learning researcher. Damián Roqueiro has been a selfless source of support in all projects described in this thesis. I benefited immensely from his expertise in carrying out research at the interface between computational biology and machine learning. His advice provided invaluable guidance throughout my PhD and will continue to shape my research for many years to come. The efforts of Dean Bodenham were instrumental in pushing forward the research on significant pattern mining in Prof. Borgwardt's lab. In addition to co-authoring three of the articles described in this thesis, I would like to thank him in particular for taking the lead in the development of a toolbox allowing to access our methods with a more user-friendly interface. I am also grateful to Dominik Grimm, who suggested to use datasets originating from genome-wide association studies of the plant model organism *A. thaliana* as a testbed to develop our algorithms. His extensive experience in that domain greatly helped speedup the pace of our research and increase its practical impact. Finally, I wish to thank Matteo Togninalli for helping me translate the abstract of this thesis.

During my PhD, I was in a privileged position to be exposed to stimulating discussions with many talented individuals. For that, I am particularly thankful to my former colleagues in Prof. Borgwardt's lab in Tübingen, Chloé-Agathe Azencott, Aasa Feragen, Carl Johann Simon-Gabriel, Barbara Rakitsch, Veronika Cheplygina and Niklas Kasenburg, as well as many other researchers in the Max Planck Institute for Intelligent Systems. I thoroughly enjoyed the opportunity provided by the "Machine Learning for Personalized Medicine" Marie Curie Initial Training Network to meet many amazing fellow PhD students and postdocs. Discussing all sorts of topics with

Melanie F. Pradier in long email chains, conferences and summer schools has been one of the highlights of my PhD, as were the experiences shared with Cankut Çubuk, Daniel Urda Muñoz, Ramouna Fouladi and Yuanlong Liu during our respective secondments. I also feel grateful for the opportunity to work with all past and present members of Prof. Borgwardt's lab in Basel, Menno Witteveen, Udo Gieraths, Xiao He, Lukas Folkman, Elisabetta Ghisu, Matteo Togninalli, Anja Gumpinger, Thomas Gumbsch, Christian Bock, Bastian Rieck, Caroline Weis, Michael Moor, Eric Wolf, Birgit Knapp and Katharina Heinrich. In addition, I would also like to thank Koji Tsuda, Ichiro Takeuchi and Junpei Komiyama for the insightful exchanges regarding significant pattern mining and machine learning.

Finally, I would like to express my gratitude to all professors who taught me during my undergraduate studies, specially those from the Signal Theory and Communications department in Universidad Carlos III de Madrid. In particular, I feel indebted to Prof. Fernando Pérez-Cruz and Prof. Antonio Artés-Rodríguez for their extraordinary professional advice, which continued throughout my PhD, and, above all, to my Master's Thesis supervisors, Prof. Emilio Parrado-Hernández and Prof. Matilde Sánchez-Fernández, who instilled in me the desire to become a researcher and provided invaluable, untiring mentoring to that end.

FUNDING: This work was funded in part by the Marie Curie Initial Training Network MLPM2012, Grant No. 316861 and the SNSF Starting Grant "Significant Pattern Mining".

CONTENTS

i	INTRODUCTION AND BACKGROUND	1
1	INTRODUCTION	3
1.1	Organisation of this thesis	7
2	STATISTICAL ASPECTS OF SIGNIFICANT PATTERN MINING	11
2.1	Problem Statement and Terminology	11
2.2	Statistical Association Testing in Significant Pattern Mining	16
2.3	The Multiple Comparisons Problem	21
3	ALGORITHMIC ASPECTS OF SIGNIFICANT PATTERN MINING	25
3.1	Overview	26
3.2	Pattern enumeration	26
3.3	Evaluating Tarone’s Minimum Attainable P-value	29
3.4	Designing a Pruning Condition	34
3.5	Implementation Considerations	35
ii	CONTRIBUTIONS	39
4	EXPLOITING THE DEPENDENCE BETWEEN PATTERNS	41
4.1	Introduction	41
4.2	Empirically Approximating the FWER Via Random Permutations	45
4.3	Permutation Testing in Significant Pattern Mining	47
4.3.1	Related work: the FastWY algorithm	48
4.3.2	Contribution: the Westfall-Young light algorithm	50
4.4	Experiments	54
4.4.1	Experimental Setup	54
4.4.2	Runtime and memory usage	56
4.4.3	Final support for pattern mining	59
4.4.4	Statistical power	60
5	CORRECTING FOR A CATEGORICAL COVARIATE	65
5.1	Introduction	65
5.2	Conditional Association Testing in Significant Pattern Mining	68
5.3	The Minimum Attainable P-value for the CMH Test	69
5.4	A Search Space Pruning Condition for the CMH Test	71
5.5	Miscellaneous aspects of the FACS algorithm	83
5.6	Experiments	85
5.6.1	Experimental setup	85
5.6.2	Simulation experiments	86
5.6.3	Applications to genome-wide association studies	93
6	GENOME-WIDE ASSOCIATION STUDIES AT A REGION LEVEL	99
6.1	Introduction	100
6.2	Method	104
6.3	Experiments	116
6.3.1	Experimental setup	116

CONTENTS

6.3.2	Simulation experiments	117
6.3.3	Experiments on real-world human and plant GWAS datasets . .	130
iii	DISCUSSION AND OUTLOOK	141
7	SUMMARY AND OUTLOOK	143
7.1	Summary	143
7.2	Outlook	149
7.3	Closing Remarks	163
iv	APPENDICES	165
A	CHAPTER SUMMARIES	167
B	AVAILABLE SOFTWARE	191
	BIBLIOGRAPHY	193

LIST OF FIGURES

Figure 2.1	Illustration of significant itemset mining on a toy dataset.	12
Figure 2.2	Illustration of significant subgraph mining on a toy dataset.	15
Figure 3.1	Illustration of a pattern enumeration tree for a significant itemset mining problem with $p = 5$ binary features.	27
Figure 3.2	Minimum attainable P-value $p_{\mathcal{S},\min}$ for Pearson’s χ^2 test and Fisher’s exact test as a function of the number $r_{\mathcal{S}}$ of occurrences of pattern \mathcal{S} in a dataset \mathcal{D}	32
Figure 3.3	Illustration of the concept of testability at an arbitrary significance level δ when Fisher’s exact test is the test statistic of choice.	33
Figure 3.4	Illustration of the effect of search space pruning in a significant itemset mining problem with $p = 5$ binary features.	35
Figure 4.1	Illustration of how dependence between patterns arises in significant itemset mining due to inclusion relationships $\mathcal{S} \subset \mathcal{S}'$ between candidate patterns $\mathcal{S}, \mathcal{S}' \in \mathcal{M}$	42
Figure 4.2	Illustration of how subset/superset relationships between patterns and the sharing of pattern sub-structures can result in a pattern $\mathcal{S} \in \mathcal{M}$ being statistically associated with many other patterns in the search space \mathcal{M}	43
Figure 4.3	Illustration of permutation testing-based resampling in the context of significant itemset mining.	46
Figure 4.4	Comparison of runtime and memory usage between FastWY and Westfall-Young light in 20 significant itemset mining experiments.	57
Figure 4.5	Comparison of runtime and memory usage between FastWY and Westfall-Young light in 12 significant subgraph mining experiments.	58
Figure 4.6	Comparison of the final support corresponding to FastWY and Westfall-Young light.	61
Figure 4.7	Empirical FWER versus j_p for two representative significant itemset mining datasets.	62
Figure 4.8	Empirical FWER versus j_p for two representative significant subgraph mining datasets.	62
Figure 5.1	An illustration of the effect of confounding on a toy significant itemset mining problem.	67
Figure 5.2	Application of the CMH test to the toy significant pattern mining dataset in Figure 5.1.	70
Figure 5.3	Minimum attainable P-value function $p_{\min}(\mathbf{r}_{\mathcal{S}})$ for the CMH test in a problem with $k = 2$ categories for the covariate.	72
Figure 5.4	Illustration of the lower envelope $\tilde{p}_{\mathcal{S},\min}$ of the minimum attainable P-value $p_{\mathcal{S},\min}$	73

Figure 5.5	FDR as a function of the signal strength ρ for our proposed approach FACS and two baseline algorithms: Bonf-CMH and LAMP- χ^2	90
Figure 5.6	Statistical power at different values of the signal strength ρ for our proposed approach FACS and two baseline algorithms: Bonf-CMH and LAMP- χ^2	91
Figure 5.7	Runtime as a function of the number of features p for our proposed approach FACS and four baseline algorithms: 2^k -FACS, m^k -FACS, Bonf-CMH and LAMP- χ^2	92
Figure 5.8	Runtime as a function of the number of categories for the covariate k for our proposed approach FACS and four baseline algorithms: 2^k -FACS, m^k -FACS, Bonf-CMH and LAMP- χ^2	93
Figure 6.1	An illustration of how genetic heterogeneity can be exploited to gain statistical power in GWASs.	102
Figure 6.2	Statistical power at different values of the signal strength ρ for our proposed approaches FAIS- χ^2 and FastCMH, as well as four baseline algorithms: Bonf- χ^2 , Bonf-CMH, Univariate- χ^2 and Univariate-CMH.	120
Figure 6.3	FWER at different values of the signal strength ρ for our proposed approaches FAIS- χ^2 and FastCMH, as well as four baseline algorithms: Bonf- χ^2 , Bonf-CMH, Univariate- χ^2 and Univariate-CMH.	121
Figure 6.4	Runtime as a function of the number of features p for our proposed approaches FAIS- χ^2 and FastCMH, as well as the baseline Bonf-CMH.	122
Figure 6.5	Runtime as a function of the number of samples n for our proposed approaches FAIS- χ^2 and FastCMH, as well as the baseline Bonf-CMH.	123
Figure 6.6	Runtime as a function of the number of categories for the covariate k for our proposed approach FastCMH and the baseline 2^k -FastCMH.	124
Figure 6.7	A comparison of the statistical power of FastCMH and several burden tests with (a) sliding windows and (b) non-overlapping windows.	127
Figure 6.8	Statistical power at different values of the signal strength ρ for our proposed approach FastCMH, as well as two baseline algorithms: Bonf-CMH and Univariate-CMH.	129
Figure 6.9	Visualisation of the three-dimensional embedding obtained by EIGENSTRAT for each sample in the COPDGene study.	132

Figure 6.10	Q-Q plots for the P-values of all testable genomic regions obtained with FAIS- χ^2 and FastCMH for all six datasets under consideration.	135
-------------	---	-----

LIST OF TABLES

Table 4.1	Characteristics of the significant itemset mining datasets.	55
Table 4.2	Characteristics of the significant subgraph mining datasets.	56
Table 5.1	Total number of feature interactions (hits) deemed significantly associated by LAMP- χ^2 , FACS and Bonf-CMH and average genomic inflation factor $\bar{\lambda}$	97
Table 5.2	Genomic inflation factor λ for different <i>univariate</i> analyses of the two GWAS datasets under consideration.	98
Table 6.1	Characteristics of the six GWAS datasets used in this chapter.	131
Table 6.2	Summary of the results of our proposed approaches FAIS- χ^2 and FastCMH.	134
Table B.1	A non-exhaustive list of existing software for significant pattern mining.	191

Part I

INTRODUCTION AND BACKGROUND

INTRODUCTION

Biomarker discovery, the search for measurable biological indicators of a phenotypic trait of interest, is a fundamental problem in healthcare and computational biology. Biomarkers help researchers better understand the biological mechanisms underlying phenotypic variation and can ultimately lead to significant advances in prevention, diagnosis and treatment of many medical conditions [1]. In recent years, drastic improvements in our ability to affordably collect large amounts of molecular data has led to a rapid growth in data availability. In particular, the past three decades have shown a (roughly) doubling, every 18 months, of the number of sequenced bases readily available in public databases [2]; a growth rate that outpaces Moore's "law" [3]. As a result, large-scale datasets containing millions of biological measurements for thousands of individuals have now become customary. Perhaps most importantly, current trends show no indications that this explosion in data availability will slow down in the foreseeable future [4]. *Precision medicine* [5–7] aims to make use of this wealth of molecular data, alongside information about a patient's lifestyle and environment, in order to personalise disease prevention and medical treatment. Thus, developing methods to efficiently and reliably discover promising biomarkers in such large-scale datasets is of utmost importance to make precision medicine a reality.

From a statistical perspective, biomarker discovery is particularly challenging due to the nature of the datasets involved [8, 9], typically containing many more features than samples. For instance, such a dataset may represent the allelic value of millions of single nucleotide polymorphisms (features or markers) for thousands of individuals (samples). In this example, the goal would be to identify the single nucleotide polymorphisms that help us differentiate between individuals of distinct phenotype (biomarkers). The difficulty of analysing datasets with such a large number of features relative to the sample size has motivated the development of novel tools for statistical inference in high-dimensional spaces. Existing work has predominantly focused on either univariate methods or multivariate linear models with sparsity-inducing regularisers. Approaches based on univariate association tests (e.g. [10]) consider the effect of each candidate marker in isolation from the others. In contrast, multivariate linear models with sparsity-inducing regularisers (e.g. [11–15]) jointly model the effect of all candidate markers as a weighted additive combination of individual effects. These techniques have been a fundamental part of many successes in biology and medicine. For instance, they have been widely used to analyse data in genome-wide association studies, helping discover more than 61,000 variant-trait associations [16, 17], many of which have led to substantial biological insight and even clinical applications [18, 19]. However, both families of approaches share a key limitation: *they are unable to discover nonlinear signals due to interactions between features*. For instance, this "blind spot" has been hypothesised as a factor that could account for at least a fraction of the "missing heritability" in genome-wide association studies [20–23]. The missing heritability problem, the phenomenon that loci discovered by

genome-wide association studies only account for a small proportion of the estimated heritability of the phenotypes, is one of the main open problems in statistical genetics. In order to test the hypothesis that nonlinear interactions between candidate markers could explain part of this missing heritability, novel biomarker discovery methods able to take feature interactions into consideration are necessary. More generally, biological mechanisms for which feature interactions have been found to play a crucial role abound in a variety of topics of utmost relevance. For example, the regulation of gene expression in different tissues is known to be dictated at least in part by combinatorial interactions among transcription factors [24–26]. Analysing feature interactions has also proven to be fruitful in oncology, where identifying co-occurring mutational events from tumour sequence data has facilitated the detection of cancer genes and pathways [27, 28].

Nevertheless, assessing the statistical association of *all* high-order feature interactions with a phenotypic trait of interest is an exceedingly difficult problem. The gap between the number of features and sample size, already large in traditional analyses, is further exacerbated when all high-order feature interactions are considered, leading to a combinatorial explosion in the effective number of features in the model. To give a sense of scale, in a dataset with $p = 266$ features, which could be considered a small number by current standards, one could explore up to $2^p \approx 10^{80}$ high-order feature interactions; as many as the estimated number of electrons in the observable universe [29, Appendix C.4]. Two fundamental difficulties arise from the daunting number of feature interactions that would need to be tested for association with the trait of interest:

- (i) A *statistical* challenge, in the form of an extreme instance of the so called *multiple comparisons problem*. When such an enormous number of associations tests are performed, it is extraordinarily difficult to control the probability of reporting false associations while maintaining enough statistical power to discover the truly significant feature interactions.
- (ii) A *computational* challenge, caused by the necessity to explore the vast search space consisting of all candidate feature interactions.

In order to tackle the computational challenge, decades of research have led to a plethora of data mining algorithms able to quickly navigate the search space of all candidate feature interactions. Most of these approaches rely on efficient schemes to enumerate feature interactions in combination with pruning criteria that allow removing a large proportion of the search space without affecting the validity of the results (e.g. [30–33]). These *discriminative pattern mining* algorithms have been largely successful in many applications, however, they are unable to account for the multiple comparisons problem. As a consequence, the statistical significance of their reported associations cannot be evaluated. While this might only be a minor inconvenience in some domains, strictly assessing the statistical significance of any positive findings is essential in biomarker discovery. Indeed, in this “big data” era, the current “reproducibility crisis” in many scientific disciplines [34–39] has made abundantly clear that, as the role of data science becomes more prominent in most fields of research, properly accounting for all sources of uncertainty when reporting any discoveries is extraordinarily important.

Despite this pressing motivation to develop approaches to correct for the multiple comparisons problem in discriminative pattern mining, an effective, uncompromising solution remained elusive for decades. If all candidate feature interactions are taken into consideration, the resulting multiple comparisons problem is enormous, way beyond what had been successfully handled in the statistics literature, leading many field experts to believe that a solution to this problem was unlikely to be found. Early attempts to account for the multiple comparisons problem in pattern mining were pioneered by [40, 41]. However, these approaches resorted to imposing limits in the number of interacting features in order to reduce the total number of candidate feature interactions in the search space and alleviate the multiple comparisons burden or, alternatively, proposed to randomly split the original dataset into separate “exploratory” and “holdout” data, leading to a potential loss of statistical power and hindering reproducibility. Follow-up work [42] proposed a method to soften the hard constraints in the maximum number of interacting features, effectively allowing to prioritise the discovery of feature interactions involving a certain number of interacting features at the expense of decreasing statistical power for other candidate feature interactions. Nevertheless, recent work presented in [26] showed that it is entirely possible to solve both the computational and statistical challenges described above while keeping *all* candidate feature interactions in the search space. This feat, achieved through a combination of classical discriminative pattern mining algorithms and highly-specialised techniques for statistical association testing with discrete data, effectively created a new branch of machine learning which we refer to as *significant pattern mining*.

The method in [26] was groundbreaking, as it showed for the first time that assessing the statistical significance of *all* high-order feature interactions with a target of interest is, in general, possible. However, the approach is not devoid of limitations, some of which severely hinder its applicability to biomarker discovery problems. The aim of this thesis is to propose novel significant pattern mining methods that overcome some of those limitations, selected on the basis of their crucial importance for biomarker discovery. Next, we briefly introduce each of our contributions, leaving an in-depth description for subsequent chapters.

Exploiting the dependence between patterns

Unlike other machine learning problems such as computer vision, speech recognition and other perceptual tasks, many datasets in computational biology are characterised by containing only extremely faint signals, barely above the noise level. This partly explains why a large body of work in machine learning for computational biology focuses on association testing rather than prediction and, most importantly, why methods to improve statistical power are among the most relevant contributions a computational scientist might make to the field.

If there is a single characteristic that defines significant pattern mining, it is the daunting size of the search space that needs to be explored. As discussed above, this is the source of statistical and computational difficulties that make significant pattern mining challenging. However, it is also the source of opportunities to improve upon the state of the art. The exhaustive nature of the search space of all candidate feature interactions makes it redundant by construction: subset/superset relationships

between distinct feature interactions induce strong statistical dependencies between the different association tests that need to be performed. Intuitively, these statistical dependencies reduce the effective number of tests one must account for when correcting for the multiple comparisons problem. While this offers an opportunity to significantly increase statistical power, modelling these dependencies appropriately is difficult, again, due to the enormous number of association tests to be considered. Consequently, existing approaches opt for ignoring these statistical dependencies between feature interactions, effectively sacrificing statistical power in exchange for simplicity. Moreover, the single approach that aims to exploit the dependencies in the search space [43] is too computationally demanding in practice, both in terms of runtime and memory usage, making it suitable only for small datasets.

In this thesis, we propose a new method to combine permutation testing with significant pattern mining. Permutation testing allows empirically estimating the joint global null distribution of all test statistics, making it possible to compute a tighter significance threshold that takes the dependence between feature interactions into account. Compared to the method introduced in [43], our algorithm is one to three orders of magnitude faster and requires two to three orders of magnitude less memory without affecting its ability to exploit the dependencies in the search space to improve statistical power. The resulting approach scales-up to a broader range of datasets, thus making it applicable to many problems relevant for computational biology.

Correcting for a categorical covariate

The need to account for covariates that might have a potentially *confounding* effect is ubiquitous in most applications for medicine and computational biology. If these covariates are not incorporated in the model, a large number of spurious false positives whose association signal is exclusively mediated by the confounder might be erroneously reported. For instance, some possible sources of confounding in biomarker discovery are factors such as gender, age, education level, pre-existing conditions or population structure, among others. Despite the potential that significant pattern mining has for biomarker discovery, it is unlikely that these methods will ever gain acceptance among practitioners unless effective ways to correct for confounders are proposed.

Accounting for covariate factors in significant pattern mining had remained an open problem since the first significant pattern mining approach was proposed in [26]. This method heavily relies on low-level properties of some specific statistical association tests for discrete data, such as Pearson’s χ^2 test [44] or Fisher’s exact test [45]. However, these properties do not apply for tests that have been traditionally used to account for covariates when testing associations in discrete data, such as the Cochran-Mantel-Haenszel test [46]. Hence, the framework in [26] cannot be used in combination with covariate factors, limiting the use of significant pattern mining to datasets for which confounding effects could be ruled out *a priori* according to domain knowledge.

A fundamental contribution of this thesis is to propose a significant pattern mining algorithm that can account for a categorical covariate. Our method makes use of the Cochran-Mantel-Haenszel test alongside a novel pruning criterion that can be evaluated in almost-linear time in the number of categories of the covariate. As a

result, our approach is able to incorporate categorical covariates such as gender, age or population structure, drastically reducing false positives due to confounding without sacrificing neither computational efficiency nor statistical power.

Genome-wide association studies at a region level

Genome-wide association studies aim to discover how common genetic variation in the population can be mapped to phenotypic differences between individuals, hoping to shed light on the underlying biology and eventually lead to better approaches for disease prevention and medical treatment. According to the common disease-common variant hypothesis, phenotypic variation for complex traits might be largely polygenic, being governed by a considerable number of variants, each accounting for a exceedingly small proportion of the variation [18]. An alternative hypothesis postulates that rare variants, i.e. those which occur in less than 1% of the population, could be the source of phenotypic variability among individuals [47]. This model expects these variants to have a large effect size. However, their rarity and the fact that their effects can only be indirectly measured in genome-wide association studies via linkage with common variants imply that the observed effect sizes would again be small.

In order to improve statistical power, many approaches have been proposed to aggregate the (possibly weak) effects of multiple neighbouring variants into a stronger, easier to detect association signal. Some of these approaches have been successful in discovering associations that would have been otherwise missed by a univariate analysis, specially for associations which involve rare variants [48]. Nonetheless, all these methods share a common limitation: they can only test a relatively small subset of genomic regions chosen *a priori*. Practitioners have used multiple criteria to select which genomic regions to test. For instance, some have used prior knowledge, defining the regions as either genes or other functional units. Others prefer to perform an agnostic, genome-wide scan and define the genomic regions to be tested by splitting the genome into (possibly overlapping) windows of a pre-specified size. Regardless of the choice, only a small proportion of all possible genomic regions will be covered. If the windows the genome is split into are chosen too small or too big, or if part of the signal happens to arise from variants outside the regions chosen according to prior knowledge, these methods will suffer a sharp loss in statistical power.

This application offers an ideal opportunity to showcase the potential of significant pattern mining for computational biology. In this thesis, we propose a new family of methods that are able to test *all* genomic regions, regardless of their starting position or size. Unlike competing approaches, our algorithm is robust to misspecification of the genomic regions to be tested and reduces the number of hyperparameters to be adjusted by the data analyst, aiding reproducibility.

1.1 ORGANISATION OF THIS THESIS

The content of this thesis is organised into three parts: (i) introduction and background, (ii) contributions, and (iii) discussion and outlook.

The first part comprises a self-contained introduction to the key background concepts the rest of this thesis relies on. In particular, Chapter 2 begins by formalising the objective of significant pattern mining and describing its two most popular instances: (i) *significant itemset mining*, which looks for statistically significant feature interactions in binary data, and ii) *significant subgraph mining*, which aims at finding significant subgraphs in a dataset of graph-structured samples. Statistical association testing for binary random variables, a fundamental sub-component of significant pattern mining, is discussed next. The chapter concludes with a detailed introduction to the multiple comparisons problem and a description of Tarone’s improved Bonferroni correction for discrete data, the key statistical tool that made it possible to manage the multiple comparisons problem in significant pattern mining. Finally, Chapter 3 describes how Tarone’s improved Bonferroni correction can be combined with techniques from classical discriminative pattern mining to obtain an efficient algorithm for significant pattern mining.

The next part is devoted to presenting all of the novel contributions of this thesis. These will be organised as follows:

- (i) Chapter 4, which discusses our new approach to exploit the dependence between patterns to improve statistical power, is based on the following publication:
 - 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), 725–734
- (ii) Chapter 5 describes our method to correct for categorical covariates in significant pattern mining, allowing practitioners to deal with potential confounding factors. The work presented in this chapter originates from 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
- (iii) Chapter 6 introduces our work on genome-wide association studies at a region level, proposing a new family of algorithms to test *all* genomic regions regardless of size and starting position. This chapter encompasses two distinct publications, the second of which is also the result of equal contributions by the two first authors:
 - Llinares-López, F., Grimm, D., Bodenham, D., Gieraths, U., Sugiyama, M., Rowan, B. & Borgwardt, K. *Genome-wide detection of intervals of genetic heterogeneity associated with complex traits*. *Bioinformatics* 31, i240–i249 (2015)
 - Llinares-López, F., Papaxanthos, L., Bodenham, D., Roqueiro, D., COPD Investigators & Borgwardt, K. *Genome-wide genetic heterogeneity discovery with categorical covariates*. *Bioinformatics* 33, i1820–i1828 (2017)

The last part of this thesis, which comprises Chapter 7, synthesises the main concepts presented throughout this document and provides a detailed outlook on the main open problems in the field.

How to read this thesis

Chapters in this thesis are not self-contained, hence, they should ideally be read in order. Nevertheless, in hopes of facilitating selective reading, a self-sufficient

summary of the essential ideas and results introduced by each chapter can be found in Appendix A. Additionally, the interested reader can find in Appendix B a brief list of existing open source software for significant pattern mining.

Finally, we wish it to be known that a book chapter heavily based on this thesis is currently under preparation. The chapter, preliminarily titled "*Machine learning for biomarker discovery: significant pattern mining*", will be part of an interdisciplinary textbook aimed at training biological, medical and computational scientists and will cover both the fundamentals of significant pattern mining as well as a simplified description of the contributions detailed in Chapters 4 and 5 of this thesis.

STATISTICAL ASPECTS OF SIGNIFICANT PATTERN MINING

2.1 PROBLEM STATEMENT AND TERMINOLOGY

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be a dataset with n distinct observations x and their corresponding labels y , sampled i.i.d. from an unknown joint probability distribution $p(x, y)$. We consider the case where each observation x exists in a finite input domain \mathcal{X} and belongs to one of two classes, i.e. $y \in \{0, 1\}$. Throughout this thesis, we will informally refer to any discrete substructure that might be part of an input sample $x \in \mathcal{X}$ as a *pattern*. The exact notion of pattern depends on the input domain \mathcal{X} ; some of the most common cases will be introduced later in this section.

Given a search space \mathcal{M} containing all candidate patterns under study, significant pattern mining aims to discover all patterns \mathcal{S} in \mathcal{M} whose occurrence within a sample is statistically significantly associated with the class labels.

We define the *pattern occurrence indicator* $g_{\mathcal{S}}(x)$ as the binary random variable that indicates whether pattern \mathcal{S} is present in an input sample x or not:

$$g_{\mathcal{S}}(x) = \begin{cases} 1, & \text{if } \mathcal{S} \subseteq x, \\ 0, & \text{otherwise.} \end{cases} \quad (2.1)$$

A pattern \mathcal{S} occurs statistically significantly more often in one class of samples than in another if and only if the class labels y and the indicator $g_{\mathcal{S}}(x)$ are statistically associated.

A precise definition of the search space of candidate patterns \mathcal{M} and the concept of inclusion or occurrence $\mathcal{S} \subseteq x$ also depend on the nature of the input domain \mathcal{X} . This abstract framework can be particularised to cover a variety of significant pattern mining instances; the remaining of this section will be devoted to describe some of the most relevant.

Significant itemset mining

Perhaps the most widespread instance of significant pattern mining corresponds to the case where the input samples are p -dimensional binary vectors, i.e. $\mathcal{X} = \{0, 1\}^p$. Essentially, this means that each input sample x comprises a collection of p different binary features $x = (u_1, u_2, \dots, u_p)$, each of which can be either active ($u_j = 1$) or inactive ($u_j = 0$).

Datasets arising from a wide variety of problems in computational biology can be described using this type of representation. For instance, in genome-wide association studies, the genotype of n individuals at a set of p single nucleotide polymorphisms can be represented as p binary features using, for example, a dominant/recessive/overdominant encoding or prior knowledge such as functional annotations. In functional genomics, given a set of n distinct genomic regions, each of the p binary features could be an indicator of whether a certain property of interest, such as exhibiting a particular

chromatin modification in a given cell type or containing a specific transcription factor binding motif, applies to a genomic region or not. In a clinical setting, many datasets contain samples that can be described by high-dimensional binary vectors as well. For instance, this type of representation can be related to Electronic Health Records (EHRs). In this case, the active binary features can be used to encode the set of medical codes from a certain medical ontology (e.g. SNOMED-CT, ICD-9) which apply to the record, among a vocabulary of p distinct medical terms. Moreover, many clinical variables can typically be described using a binary feature that indicates whether its observed value lies within the normal, healthy range or not.

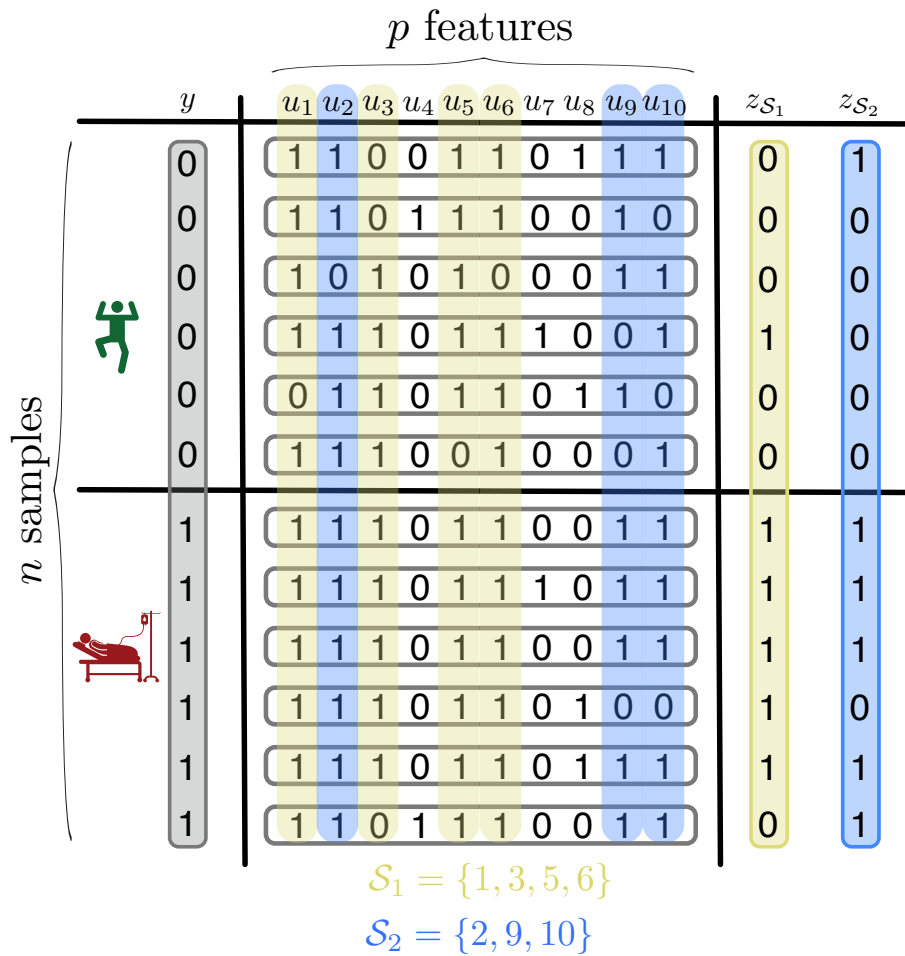


Figure 2.1. – Illustration of significant itemset mining on a toy dataset with $n = 12$ samples, divided into $n_1 = 6$ samples belonging to class $y = 1$ (e.g. cases) and $n_0 = 6$ samples belonging to class $y = 0$ (e.g. controls). Each sample is represented by $p = 10$ binary features, u_1, u_2, \dots, u_{10} . Patterns $\mathcal{S}_1 = \{1, 3, 5, 6\}$ (yellow) and $\mathcal{S}_2 = \{2, 9, 10\}$ (blue) are highlighted in the figure alongside their induced feature interactions $z_{\mathcal{S}_1}(x)$ and $z_{\mathcal{S}_2}(x)$. While none of the ten binary features is individually associated with the class labels, the high-order feature interactions $z_{\mathcal{S}_1}(x)$ and $z_{\mathcal{S}_2}(x)$ are active considerably more often among samples of class $y = 1$.

In *significant itemset mining*, each pattern \mathcal{S} corresponds to a different candidate feature interaction. Let $\mathcal{S} \subseteq \{1, 2, \dots, p\}$ be the index set of an arbitrary subset of the p binary features. We define $z_{\mathcal{S}}(x)$, the *feature interaction* induced by \mathcal{S} , as the multiplicative interaction of the features indexed by \mathcal{S} , i.e. $z_{\mathcal{S}}(x) = \prod_{j \in \mathcal{S}} u_j$. In particular, note that $z_{\mathcal{S}}(x) = 1$ if and only if *all* features indexed by \mathcal{S} are simultaneously active in the sample x and $z_{\mathcal{S}}(x) = 0$ otherwise. We consider that a pattern \mathcal{S} occurs in a sample $x \in \{0, 1\}^p$, i.e. $\mathcal{S} \subseteq x$, if the feature interaction induced by \mathcal{S} is active ($z_{\mathcal{S}}(x) = 1$). Therefore, the occurrence indicator $g_{\mathcal{S}}(x)$ of a pattern \mathcal{S} in significant itemset mining is identical to the feature interaction $z_{\mathcal{S}}(x)$ induced by \mathcal{S} . In order to justify this definition, consider the alternative representation of each input sample $x \in \{0, 1\}^p$ given by the set of indices of the features which are active in x . For instance, the input sample $x_1 = (1, 0, 1, 1, 0)$ can be represented as $x_1 = \{1, 3, 4\}$ and $x_2 = (0, 1, 0, 1, 0)$ as $x_2 = \{2, 4\}$. This defines a one-to-one mapping between the set of p -dimensional binary vectors $\{0, 1\}^p$ and the set of all subsets of $\{1, 2, \dots, p\}$, i.e. the power-set $\mathcal{P}(\{1, 2, \dots, p\})$. Thus, both representations are equivalent and the input domain can also be defined as $\mathcal{X} = \mathcal{P}(\{1, 2, \dots, p\})$. Moreover, $\mathcal{S} \subseteq x$, with x represented as a set of active features and \subseteq denoting traditional set inclusion, holds if and only if the feature interaction $z_{\mathcal{S}}(x)$ as defined above has value 1. Both representations of x therefore lead to the same pattern occurrence indicator $g_{\mathcal{S}}(x)$. The latter notation, describing samples as a set of active features or *items*, is commonly used in the data mining field, being ultimately responsible for giving this particular instance of significant pattern mining the name of significant *itemset* mining.

Whenever *all* feature interactions are of potential interest, the search space of candidate patterns \mathcal{M} would contain all possible feature subsets, i.e. $\mathcal{M} = \mathcal{P}(\{1, 2, \dots, p\})$, thus comprising 2^p different patterns. This setting is perhaps the most common case; however, certain forms of prior knowledge can be incorporated into the model simply by modifying the definition of the search space \mathcal{M} . For instance, in the context of genome-wide association studies, researchers could decide to restrict the analysis to study only interactions between variants belonging to the same biological pathway or to the same genomic region. Provided that the design of \mathcal{M} according to domain knowledge is successful in keeping all relevant feature interactions while discarding many others, the reduction in the number of candidate patterns prior to the analysis will lead to improved computational efficiency and statistical power. This opens the door to the development of novel instances of significant itemset mining, targeting particular problems in medicine and computational biology.

Figure 2.1 depicts a conceptual illustration of significant itemset mining on a toy dataset with $n = 12$ samples, $n_1 = 6$ belonging to class $y = 1$ (e.g. cases) and $n_0 = 6$ to class $y = 0$ (e.g. controls). Each of these samples is represented by a binary vector with $p = 10$ features, u_1, u_2, \dots, u_{10} . Two patterns, $\mathcal{S}_1 = \{1, 3, 5, 6\}$ (yellow) and $\mathcal{S}_2 = \{2, 9, 10\}$ (blue), as well as their induced feature interactions, $z_{\mathcal{S}_1}(x)$ and $z_{\mathcal{S}_2}(x)$, are highlighted in the figure. In this example, none of the ten features is individually associated with the class labels. Thus, univariate analyses or multivariate additive models would be unable to discover any association in this dataset. Nevertheless, the feature interactions $z_{\mathcal{S}_1}(x)$ and $z_{\mathcal{S}_2}(x)$ are active significantly more often in samples which belong to class $y = 1$ than in samples which belong to class $y = 0$. This simple example is sufficient to illustrate the fact that, even in

the absence of univariate associations, nonlinear interactions between features can strongly correlate with a target of interest. Unless methods to explore such feature interactions in high-dimensional datasets are developed, many signals of practical importance might remain undiscovered by existing approaches.

Significant subgraph mining

A different instance of significant pattern mining arises when input samples correspond to graphs. In this case, the input domain \mathcal{X} can be defined as $\mathcal{X} = \{x \mid x = (V, E, l_{V,E})\}$, where V is a set of nodes, $E \subseteq V \times V$ a set of edges and $l_{V,E} : V \cup E \rightarrow \Sigma_{V,E}$ a function that labels each node and edge in the graph with a categorical value from a finite alphabet $\Sigma_{V,E}$. The domain of $l_{V,E}$ can be redefined to account for graphs without edge or node labels. If neither nodes nor edges are labelled, $l_{V,E}$ does not need to be included as part of the definition of the input domain \mathcal{X} .

Graphs are general-purpose objects, being able to represent almost any kind of data. For instance, p -dimensional binary vectors can be modelled as fully-connected graphs with p nodes, one for each binary feature. These nodes would have binary labels, indicating the value taken by the corresponding feature in the sample, while edges would remain unlabelled. Data types such as time-series, images or video can all be accurately described by a grid-like graph. Each node in the graph would again map to a distinct input feature while, in this case, the neighbourhood of the node would represent the corresponding Markov blanket of the feature. This generality is the primary reason why graphs are among the most important types of structured data. Graphs are also ubiquitous in applications for the life sciences. As an example, they are commonly used to describe molecular compounds in cheminformatics: each atom is associated with a different node in the graph, labeled by its atomic symbol, and edges describe atomic bonds, labeled according to the type of bond [53]. Additional molecular properties (e.g. implicit valence, number of implicit hydrogen atoms, aromaticity) can also be incorporated as part of the node or edge labels. Many types of data in computational biology, such as protein structures, biological pathways or co-expression networks, are customarily represented in the form of graphs. Healthcare applications are no less abundant with graph-structured data. For instance, they are commonly used in medicine and neuroscience to describe the result of brain magnetic resonance imaging (MRI) scans, with edges quantifying the connectivity between a predefined set of brain regions (nodes) [54, 55]. As a final example, most medical ontologies, such as those mentioned in the previous section, are structured as directed acyclic graphs.

While patterns correspond to feature interactions in significant itemset mining, here each candidate pattern \mathcal{S} is identified with a different subgraph of an input sample in \mathcal{D} . Hence, this instance of significant pattern mining is commonly referred to as *significant subgraph mining*. A pattern \mathcal{S} is said to occur in an input observation $x \in \mathcal{X}$, i.e. $\mathcal{S} \subseteq x$, if and only if \mathcal{S} is an induced subgraph of x . The search space of candidate patterns \mathcal{M} comprises the set of *all* distinct subgraphs of input graphs present in the dataset \mathcal{D} . While the exact number of candidate patterns in \mathcal{M} will vary from dataset to dataset, it will typically grow combinatorially with the size of the graphs in \mathcal{D} .

Thus, not unlike the case of significant itemset mining, the resulting search space \mathcal{M} will contain an enormous number of candidate patterns.

In summary, the goal of significant subgraph mining is to discover all subgraphs \mathcal{S} of graph samples in a dataset \mathcal{D} which occur statistically significantly more often in one class of graph samples than in another.

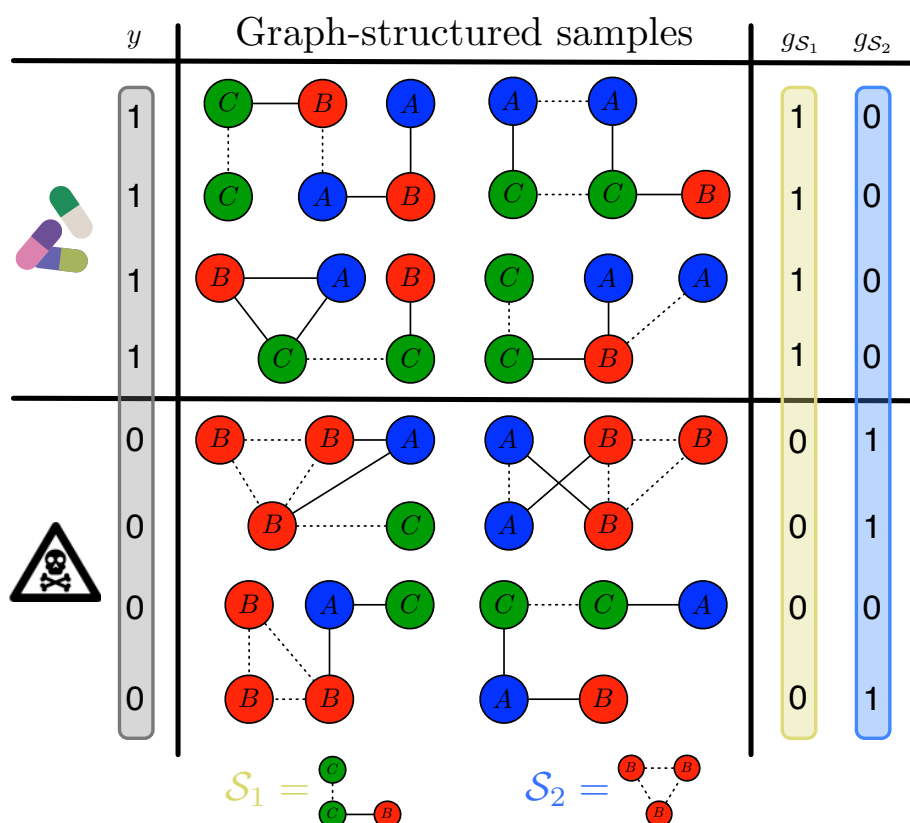


Figure 2.2. – Illustration of significant subgraph mining on a toy dataset with $n = 8$ samples (graphs), divided into $n_1 = 4$ samples belonging to class $y = 1$ (e.g. successful drugs) and $n_0 = 4$ samples belonging to class $y = 0$ (e.g. drugs that trigger an adverse reaction). Samples are represented by graphs with labeled nodes and edges ($|\Sigma_{V,E}| = 5$). Two patterns (subgraphs) which are associated with the class labels have been highlighted in the figure, \mathcal{S}_1 (yellow) and \mathcal{S}_2 (blue).

Figure 2.2 illustrates significant subgraph mining on a toy dataset with $n = 8$ graphs, $n_1 = 4$ belonging to class $y = 1$ (e.g. successful drugs) and $n_0 = 4$ belonging to class $y = 0$ (e.g. drugs which generate an adverse reaction). Each sample (drug) is given by a graph with labeled nodes and edges ($|\Sigma_{V,E}| = 5$). Two different patterns, i.e. subgraphs, \mathcal{S}_1 (yellow) and \mathcal{S}_2 (blue), are highlighted in the figure. Their occurrence indicators $g_{\mathcal{S}_1}$ and $g_{\mathcal{S}_2}$ are also shown on the right of the figure, with entries in the vectors corresponding to graphs in clockwise order for each of the two classes separately. In this particular example, subgraph \mathcal{S}_1 is overrepresented in class $y = 1$ while \mathcal{S}_2 occurs significantly more often in class $y = 0$.

2.2 STATISTICAL ASSOCIATION TESTING IN SIGNIFICANT PATTERN MINING

Irregardless of the type of pattern under study, be it itemsets, subgraphs or any other kind of discrete substructure, all instances of significant pattern mining represent a pattern \mathcal{S} by its occurrence indicator $g_{\mathcal{S}}(x)$. Therefore, from a statistical perspective, the particular nature of the input domain \mathcal{X} does not need to be taken into consideration; what is required is a principled approach to test the statistical association of two binary random variables, the class labels Y and the pattern occurrence indicator $G_{\mathcal{S}}(X)$, according to the n realisations $\{(g_{\mathcal{S}}(x_i), y_i)\}_{i=1}^n$ which can be obtained from the input dataset \mathcal{D} .

Two random variables G and Y are *statistically independent*, denoted $G \perp\!\!\!\perp Y$, if their joint probability distribution $\Pr(G = g, Y = y)$ factorises as $\Pr(G = g, Y = y) = \Pr(G = g)\Pr(Y = y)$. This is equivalent to the conditional probability distributions $\Pr(G = g | Y = y)$ and $\Pr(Y = y | G = g)$ being equal to the marginal distributions $\Pr(G = g)$ and $\Pr(Y = y)$, respectively. In contrast, G and Y are *statistically associated*, denoted $G \not\perp\!\!\!\perp Y$, if and only if they are *not* statistically independent. See [56] for an in-depth, self-contained discussion on statistical independence.

In significant pattern mining, the random variables G and Y both take binary values. Hence, the joint distribution $\Pr(G = g, Y = y)$ consists of only four probabilities: $\Pr(G = 0, Y = 0) = p_{0,0}$, $\Pr(G = 0, Y = 1) = p_{0,1}$, $\Pr(G = 1, Y = 0) = p_{1,0}$ and $\Pr(G = 1, Y = 1) = p_{1,1}$. This joint distribution is typically depicted as a 2×2 contingency table:

Variables	$G = 1$	$G = 0$	Row totals
$Y = 1$	$p_{1,1}$	$p_{0,1}$	p_Y
$Y = 0$	$p_{1,0}$	$p_{0,0}$	$1 - p_Y$
Col. totals	p_G	$1 - p_G$	1

The marginal distributions of G and Y can be obtained from the joint distribution as $\Pr(G = 1) = p_{1,0} + p_{1,1} = p_G$ and $\Pr(Y = 1) = p_{0,1} + p_{1,1} = p_Y$.

The challenge in determining whether two random variables G and Y are statistically independent or statistically associated stems from the fact that the joint distribution $\Pr(G = g, Y = y)$ is generally unknown. In practice, only a set $\{(g_{\mathcal{S}}(x_i), y_i)\}_{i=1}^n$ of n i.i.d. realisations from the joint distribution is available. These n samples can be used to obtain a frequentist estimate of the unknown joint distribution $\Pr(G_{\mathcal{S}}(X) = g, Y = y)$ as the proportion of samples that had the event $\{G_{\mathcal{S}}(x) = g, Y = y\}$ as an outcome, for each $(g, y) \in \{0, 1\}^2$. This estimation process is also often represented by means of a 2×2 contingency table:

Variables	$g_{\mathcal{S}}(x) = 1$	$g_{\mathcal{S}}(x) = 0$	Row totals
$y = 1$	$a_{\mathcal{S}}$	$b_{\mathcal{S}}$	n_1
$y = 0$	$d_{\mathcal{S}}$	$c_{\mathcal{S}}$	n_0
Col. totals	$r_{\mathcal{S}}$	$q_{\mathcal{S}}$	n

with (1) $a_{\mathcal{S}} = \sum_{i=1}^n y_i g_{\mathcal{S}}(x_i)$ being the number of samples for which the event $\{G_{\mathcal{S}} = 1, Y = 1\}$ occurs; (2) $b_{\mathcal{S}} = \sum_{i=1}^n y_i (1 - g_{\mathcal{S}}(x_i))$ the number of samples for which the event $\{G_{\mathcal{S}} = 0, Y = 1\}$ occurs; (3) $c_{\mathcal{S}} = \sum_{i=1}^n (1 - y_i) (1 - g_{\mathcal{S}}(x_i))$ the number of samples for which the event $\{G_{\mathcal{S}} = 0, Y = 0\}$ occurs and (4) $d_{\mathcal{S}} = \sum_{i=1}^n (1 - y_i) g_{\mathcal{S}}(x_i)$

the number of samples for which the event $\{G_S = 1, Y = 0\}$ occurs. The corresponding estimates $\hat{p}_{0,0}, \hat{p}_{0,1}, \hat{p}_{1,0}, \hat{p}_{1,1}$ of the four unknown probabilities $p_{0,0}, p_{0,1}, p_{1,0}, p_{1,1}$ can be derived from these counts as: $\hat{p}_{0,0} = c_S/n$, $\hat{p}_{0,1} = b_S/n$, $\hat{p}_{1,0} = d_S/n$ and $\hat{p}_{1,1} = a_S/n$. The unknown marginal distributions of G and Y can be estimated using these counts as well: $\hat{p}_G = (a_S + d_S)/n = r_S/n$ and $\hat{p}_Y = (a_S + b_S)/n = n_1/n$.

As the empirical estimate of the joint distribution will always incur error due to random sampling, the definition of statistical independence cannot be readily applied. Instead, a procedure able to account for the uncertainty introduced by the approximation is needed. In order to tackle this problem, frequentist *statistical association testing* typically relies on the concept of *P-value*, a certain scalar measure of association that is calibrated against the stochasticity inherent to the sampling process. Most frequentist statistical association testing procedures proceed as follows:

1. Choose an appropriate *test statistic* T . Any function $T: \{(g_S(x_i), y_i)\}_{i=1}^n \rightarrow \mathbb{R}$ which maps the set of n i.i.d. realisations to a scalar value is a valid statistic. Nevertheless, not all such functions will be equally useful for association testing. Intuitively, a suitable test statistic should map data samples $\{(g_S(x_i), y_i)\}_{i=1}^n$ generated from joint distributions for which $G_S(X)$ and Y are statistically independent and data samples $\{(g_S(x_i), y_i)\}_{i=1}^n$ generated from joint distributions for which $G_S(X)$ and Y are statistically associated to sufficiently distinct values. In this way, the output of such a test statistic T could be interpreted as an empirical measure of association between $G_S(X)$ and Y according to the n observed samples $\{(g_S(x_i), y_i)\}_{i=1}^n$.
2. Compute the *null distribution* $\Pr(T = t \mid H_0)$ of the test statistic T . This is the probability distribution of the test statistic T under the *null hypothesis* $H_0 = G_S(X) \perp Y$, i.e., under the hypothesis that the sample $\{(g_S(x_i), y_i)\}_{i=1}^n$ is generated from a joint distribution for which $G_S(X)$ and Y are statistically independent.
3. In order to account for the uncertainty due to random sampling, the value t that the test statistic T takes on the real data sample $\{(g_S(x_i), y_i)\}_{i=1}^n$ will be transformed into a P-value. The P-value is defined as the probability that the test statistic T takes a value at least as extreme as t , i.e. a value representing an association at least as strong, under the null hypothesis H_0 . For most test statistics, larger values of t are indicative of stronger associations. In these cases, the P-value would be obtained as $p = \Pr(T \geq t \mid H_0)$.
4. The random variables $G_S(X)$ and Y are deemed *significantly associated* if the corresponding P-value is smaller or equal than a significance threshold α defined *a priori*, i.e. if $p \leq \alpha$. The significance threshold α can be understood as the *type I error* of the procedure; it is the probability that $G_S(X)$ and Y are deemed significantly associated according to a sample $\{(g_S(x_i), y_i)\}_{i=1}^n$ generated from a joint distribution for which $G_S(X)$ and Y are however statistically independent. Decreasing the significance threshold α will reduce type I error. However, it will likewise decrease statistical power, the probability to correctly identify truly existing associations. Consequently, the optimal value of α will be in general application-specific, reflecting the corresponding costs of false positives and false negatives.

A sufficiently small P-value merely indicates that the null distribution $\Pr(T = t \mid H_0)$ is a poor fit to the observed value t of the test statistic T . This is most typically interpreted as potential evidence against the null hypothesis of independence, supporting the alternative hypothesis that $G_S(X)$ and Y might be statistically associated. Nevertheless, there is a myriad of reasons why the model fit might be poor even in cases for which $G_S(X)$ and Y are indeed statistically independent. For instance, often the null distribution $\Pr(T = t \mid H_0)$ is not known exactly and needs to be approximated. In these cases, low P-values could also arise due to an insufficiently accurate approximation to the null distribution. Alternatively, the derivation of a null distribution $\Pr(T = t \mid H_0)$ for a certain test statistic T could involve additional assumptions beyond the null hypothesis of independence, such as postulating a certain parametric form. If these assumptions are not satisfied by the real data sample, the resulting P-value could be small even if $G_S(X)$ and Y were truly independent. Another common situation that might spuriously lead to low P-values occurs whenever the n samples that conform the input dataset \mathcal{D} are not obtained via i.i.d sampling. This issue arises frequently in many biomarker discovery problems as the result of phenomena such as batch effects, population structure or other confounders.

All techniques discussed throughout this thesis are machine learning methods for data exploration, which use statistical association testing as means to provide practitioners with tools to ease navigating the overwhelming sea of noise that characterises typical biomarker discovery datasets. The output of these techniques, in the form of a set of significantly associated patterns, should therefore *not* be considered as definitive, unquestionable discoveries but, rather, as promising findings that ought to be investigated further.

The remainder of this section will be devoted to introduce Pearson's χ^2 test [44] and Fisher's exact test [45], two of the most widespread tests of statistical association between a pair of binary random variables. A rigorous presentation of both tests can be found in the original articles [44, 45]. However, this section aims to provide a self-contained, informal derivation with emphasis on building intuition about both test statistics.

Pearson's χ^2 test and Fisher's exact test are both defined in terms of the counts of the empirical 2×2 contingency table. In particular, the counts a_S , r_S and n_1 are enough to describe the data sample without loss of generality. These, together with the sample size n , uniquely determine all other counts in the contingency table. Pearson's χ^2 test and Fisher's exact test both make the assumption that the marginal counts r_S and n_1 , respectively related to the marginal distributions $\Pr(G_S(X) = g)$ and $\Pr(Y = y)$, contain little information about the potential existence of an association between the two binary random variables. Thus, both test statistics will be derived conditioning on r_S and n_1 , efficiently treating these margins as fixed quantities. This leaves the count a_S as the single random quantity in the model. As a consequence, the null distribution for both tests will be of the form $\Pr(T(A_S) = t \mid R_S = r_S, N_1 = n_1, H_0)$; it is the specific choice of transformation $T(A_S)$ what will differentiate both tests.

The first step to derive the null distributions of Pearson's χ^2 test and Fisher's exact test will be to compute the conditional probability distribution $\Pr(A_S = a_S \mid R_S = r_S, N_1 = n_1, H_0)$ under the null hypothesis of independence $H_0 = G_S(X) \perp Y$. The result is summarised in the following proposition:

Proposition 2.1. *The conditional probability distribution of A_S given that $R_S = r_S$, $N_1 = n_1$ and sample size n under the null hypothesis of independence $H_0 = G_S(X) \perp\!\!\!\perp Y$ is a hypergeometric distribution with parameters n , n_1 and r_S :*

$$\begin{aligned} \Pr(A_S = a_S \mid R_S = r_S, N_1 = n_1, H_0) &= \text{Hypergeom}(a_S \mid n, n_1, r_S) \\ &= \frac{\binom{n_1}{a_S} \binom{n-n_1}{r_S-a_S}}{\binom{n}{r_S}}. \end{aligned} \quad (2.2)$$

Proof. By definition of conditional probability distribution we have:

$$\Pr(A_S = a_S \mid R_S = r_S, N_1 = n_1, H_0) = \frac{\Pr(A_S = a_S, R_S = r_S \mid N_1 = n_1, H_0)}{\Pr(R_S = r_S \mid N_1 = n_1, H_0)}. \quad (2.3)$$

Since $R_S = A_S + D_S$, the joint distribution in the numerator can be rewritten as $\Pr(A_S = a_S, R_S = r_S \mid N_1 = n_1, H_0) = \Pr(A_S = a_S, D_S = r_S - a_S \mid N_1 = n_1, H_0)$. Note that A_S depends only on samples for which $y_i = 1$ while D_S depends only on samples for which $y_i = 0$. Therefore, under the assumption that all n samples are i.i.d. draws, the random variables A_S and D_S are statistically independent. This allows the joint distribution in the numerator to be decomposed as $\Pr(A_S = a_S, R_S = r_S \mid N_1 = n_1, H_0) = \Pr(A_S = a_S \mid N_1 = n_1, H_0) \Pr(D_S = r_S - a_S \mid N_1 = n_1, H_0)$.

Let $p_{1|0} = \Pr(G_S = 1 \mid Y = 0)$ and $p_{1|1} = \Pr(G_S = 1 \mid Y = 1)$. If the null hypothesis of independence H_0 holds, then $p_{1|1} = p_{1|0} = p_G$. If the n samples are obtained as i.i.d. draws, A_S can be modeled as the sum of n_1 independent Bernoulli random variables, each with success probability p_G . Hence, $\Pr(A_S = a_S \mid N_1 = n_1, H_0) = \text{Binomial}(a_S \mid n_1, p_G) = \binom{n_1}{a_S} p_G^{a_S} (1 - p_G)^{n_1 - a_S}$. Analogously, D_S can be modeled as the sum of $n - n_1$ independent Bernoulli random variables, each with success probability p_G . Hence, $\Pr(D_S = r_S - a_S \mid N_1 = n_1, H_0) = \text{Binomial}(r_S - a_S \mid n - n_1, p_G) = \binom{n-n_1}{r_S-a_S} p_G^{r_S-a_S} (1 - p_G)^{(n-n_1)-(r_S-a_S)}$. Finally, since $R_S = A_S + D_S$, with $A_S \perp\!\!\!\perp D_S$, R_S corresponds to a sum of n Bernoulli variables with success probability p_G , leading to $\Pr(R_S = r_S \mid N_1 = n_1, H_0) = \text{Binomial}(r_S \mid n, p_G) = \binom{n}{r_S} p_G^{r_S} (1 - p_G)^{n-r_S}$.

Substituting those distributions into Equation (2.3) leads to the final result. In particular, the conditioning on $R_S = r_S$ eliminates the influence of the nuisance parameter p_G , leading to a distributional form that depends only on n_1 , r_S and the sample size n . \square

Since Pearson's χ^2 test and Fisher's exact test can both be written as a transformation $T(A_S)$ of the count A_S , Proposition 2.1 easily leads to the specific null distributions for each of the two tests, as shown next.

Pearson's χ^2 test

Pearson's χ^2 test can be seen as the square of a Z-score:

$$\begin{aligned} Z_{\text{pearson}}(a_S \mid n, n_1, r_S) &= \frac{a_S - \mathbb{E}[a_S \mid R_S = r_S, N_1 = n_1, H_0]}{\text{Std}[a_S \mid R_S = r_S, N_1 = n_1, H_0]} \\ T_{\text{pearson}}(a_S \mid n, n_1, r_S) &= Z_{\text{pearson}}^2(a_S \mid n, n_1, r_S), \end{aligned} \quad (2.4)$$

where $\mathbb{E}[a_S \mid R_S = r_S, N_1 = n_1, H_0] = r_S \frac{n_1}{n}$ and $\text{Std}[a_S \mid R_S = r_S, N_1 = n_1, H_0] = \sqrt{\frac{r_S}{n} \frac{n-r_S}{n} \frac{n-n_1}{n-1} n_1}$ are the mean and standard deviation of a hypergeometric distribution

with parameters n , n_1 and r_S . Substituting these into (2.4), the following expression for the test statistic is obtained:

$$T_{\text{pearson}}(a_S | n, n_1, r_S) = \frac{(a_S - r_S \frac{n_1}{n})^2}{\frac{r_S}{n} \frac{n-r_S}{n} \frac{n-n_1}{n-1} n_1}. \quad (2.5)$$

Large values of $T_{\text{pearson}}(a_S | n, n_1, r_S)$ are less likely to occur under the null hypothesis, hence hinting at the potential existence of an association.

Provided that the sample size n is sufficiently large, the central limit theorem can be used to justify approximating the distribution of $Z_{\text{pearson}}(a_S | n, n_1, r_S)$ under the null hypothesis H_0 by a standard normal distribution. Thus, the null distribution of the test statistic $\Pr(T_{\text{pearson}}(A_S) = t | R_S = r_S, N_1 = n_1, H_0)$ can in turn be approximated as a χ_1^2 distribution for sufficiently large n . Finally, the corresponding two-tailed P-value can be obtained from the survival function of a χ_1^2 distribution, i.e.

$$p_{\text{pearson}}(a_S | n, n_1, r_S) = 1 - F_{\chi_1^2}(T_{\text{pearson}}(a_S | n, n_1, r_S)), \quad (2.6)$$

where $F_{\chi_1^2}(\bullet)$ is the cumulative density function of a χ_1^2 distribution.

Fisher's exact test

Multiple approaches have been proposed in the literature to compute two-tailed P-values for Fisher's exact test. For the sake of clarity, this section will present only one such method. Nevertheless, all techniques that will be discussed in subsequent chapters of this thesis can be readily extended to work with order definitions of two-tailed P-values for Fisher's exact test, as well as with one-tailed P-values.

Fisher's exact test can be motivated by considering the probability $\Pr(A_S = a_S | R_S = r_S, N_1 = n_1, H_0)$ to be the test statistic:

$$T_{\text{fisher}}(a_S | n, n_1, r_S) = \text{Hypergeom}(a_S | n, n_1, r_S) = \frac{\binom{n_1}{a_S} \binom{n-n_1}{r_S-a_S}}{\binom{n}{r_S}}. \quad (2.7)$$

By definition, $T_{\text{fisher}}(a_S | n, n_1, r_S)$ will be small for improbable values of a_S under the null distribution $\Pr(A_S = a_S | R_S = r_S, N_1 = n_1, H_0)$. Therefore, in this particular case, small values of the test statistic $T_{\text{fisher}}(a_S | n, n_1, r_S)$ are indicative of a potential association rather than large values. Hence, a two-tailed P-value will be computed as $p = \Pr(T \leq t | H_0)$.

Formally, define $\mathcal{A}(a_S)$ as the set of all possible counts a'_S which are at least as improbable as a_S under the null distribution, i.e.,

$$\mathcal{A}(a_S) = \{a'_S | \text{Hypergeom}(a'_S | n, n_1, r_S) \leq \text{Hypergeom}(a_S | n, n_1, r_S)\}.$$

Then:

$$p_{\text{fisher}}(a_S | n, n_1, r_S) = \sum_{a'_S \in \mathcal{A}(a_S)} \text{Hypergeom}(a'_S | n, n_1, r_S). \quad (2.8)$$

As indicated by its name, Fisher's exact test does not require approximations to model its null distribution. It is therefore typically preferred over Pearson's χ^2 test, most notably in situations where the sample size n is too small to justify asymptotic approximations based on the central limit theorem.

2.3 THE MULTIPLE COMPARISONS PROBLEM

Significant pattern mining aims to retrieve, among all candidate patterns \mathcal{S} in a search space \mathcal{M} , the set of informative patterns whose occurrence within a sample is significantly associated with the class labels. Techniques to test the statistical association of binary random variables, such as Pearson's χ^2 test and Fisher's exact test, provide a principled approach to assess the statistical significance of each pattern $\mathcal{S} \in \mathcal{M}$ according to n labeled observations in an input dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$. Given an appropriate test statistic T and a desired significance threshold α , the framework presented in the previous section could in principle be applied independently to each pattern \mathcal{S} in the search space \mathcal{M} , leading to a set of P-values $\{p_{\mathcal{S}} \mid \mathcal{S} \in \mathcal{M}\}$. These could then be used to obtain a tentative set of significantly associated patterns $\widehat{\mathcal{M}}_{\text{sig}} = \{\mathcal{S} \in \mathcal{M} \mid p_{\mathcal{S}} \leq \alpha\}$. Temporarily obviating the evident computational limitations of such a brute-force approach, this naive application of statistical association testing to pattern mining would however lead to an even more severe statistical caveat. As discussed in the previous section, the significance threshold α can be understood as the type I error of each association test: it is the probability that a pattern \mathcal{S} entirely irrelevant for the purpose of determining which class a sample belongs to is erroneously deemed significantly associated. Therefore, this naive procedure would on average produce $\alpha|\mathcal{M}_0|$ false positives, where $\mathcal{M}_0 \subseteq \mathcal{M}$ denotes the set of patterns whose occurrence within a sample is statistically independent of the class membership of the sample. In many practical situations, including the vast majority of biomarker discovery problems, the search space \mathcal{M} contains an enormous number $|\mathcal{M}|$ of candidate patterns, most of which are irrelevant patterns $\mathcal{S} \in \mathcal{M}_0$, i.e. $|\mathcal{M}_0| \approx |\mathcal{M}|$. If the significance threshold α is fixed *a priori*, irregardless of the number $|\mathcal{M}|$ of association tests being performed, the few truly significant patterns contained in the output of this hypothetical significant pattern mining algorithm would be interspersed with billions or even trillions of false positives, drastically compromising the reliability of all reported findings. The need for more sophisticated statistical association testing procedures when multiple association tests are carried out simultaneously has long been understood by statisticians. This phenomenon, traditionally referred to as the *multiple comparisons problem* or *multiple hypothesis testing problem*, has been extensively studied for several decades (e.g. [57–59]). Nevertheless, the multiple comparisons problem that arises in significant pattern mining is justifiably unique: never before had an application required performing such a gigantic number of simultaneous association tests as significant pattern mining does.

Rather than controlling the per-hypothesis type I error, correcting for the multiple comparisons problem requires adopting new error measures that instead consider the entire collection of associations tests. A popular criterion is the *Family-Wise Error Rate (FWER)*, which is defined as the probability of reporting any false positives in the entire body of tests. Formally, $\text{FWER}(\delta) = \Pr(\text{FP}(\delta) > 0)$, where $\text{FP}(\delta)$ is the number of false positives at significance threshold δ , i.e., the number of patterns $\mathcal{S} \in \mathcal{M}_0$ for which $p_{\mathcal{S}} \leq \delta$. A common goal to account for the multiple comparisons problem is to *control* the FWER, that is, to guarantee that the FWER is bounded above by a pre-specified level α . A simple approach to achieve this is to properly adjust the significance threshold δ . By definition, $\text{FWER}(\delta)$ is a monotonically increasing function

of δ . Therefore, the optimal *corrected significance threshold* δ^* is given by the largest value of δ , in order to maximise statistical power, that still satisfies $\text{FWER}(\delta) \leq \alpha$:

$$\delta^* = \max \{ \delta \mid \text{FWER}(\delta) \leq \alpha \}. \quad (2.9)$$

Nevertheless, evaluating the FWER at a certain significance threshold δ is, in general, an intractable problem. The function $\text{FWER}(\delta)$ is often exceedingly complex, as it depends on the unknown joint distribution of the collection of P-values corresponding to all null hypotheses $\{p_{\mathcal{S}} \mid \mathcal{S} \in \mathcal{M}_0\}$. Rather than using the exact but generally unknown function $\text{FWER}(\delta)$, most existing approaches resort to the introduction of an easier-to-compute surrogate function $\widehat{\text{FWER}}(\delta)$, most often designed to be a strict upper bound of $\text{FWER}(\delta)$. An approximation $\hat{\delta}^*$ of the optimal significance threshold δ^* can then be obtained by using the surrogate $\widehat{\text{FWER}}(\delta)$ in place of $\text{FWER}(\delta)$:

$$\hat{\delta}^* = \max \{ \delta \mid \widehat{\text{FWER}}(\delta) \leq \alpha \}. \quad (2.10)$$

In particular, if $\widehat{\text{FWER}}(\delta)$ is chosen such that $\text{FWER}(\delta) \leq \widehat{\text{FWER}}(\delta)$ for all $\delta \in [0, 1]$, then $\widehat{\text{FWER}}(\delta) \leq \alpha$ implies that $\text{FWER}(\delta) \leq \alpha$ as well. Thus, a solution of (2.10) would also control the FWER at level α but could nevertheless lead to a considerable loss of statistical power if the bound $\widehat{\text{FWER}}(\delta)$ is too loose, i.e. if $\text{FWER}(\delta) \ll \widehat{\text{FWER}}(\delta)$.

In the remaining of this section, two distinct procedures for controlling the FWER will be presented. These differ on the particular form of the upper bound $\widehat{\text{FWER}}(\delta)$, which ultimately determines both the resulting statistical power and the underlying computational complexity of each method. The purpose of this discussion is not to provide a comprehensive review of the state-of-the-art for the multiple comparisons problem but, rather, to introduce the subset of techniques from this field which are most relevant for significant pattern mining. In particular, fundamentally important families of approaches, such as sequential rejection procedures, will *not* be covered in this thesis.

The Bonferroni correction

The most widespread approach to control the FWER is the *Bonferroni correction* [60, 61]. This method approximates the unknown FWER by an extraordinarily simple surrogate: $\widehat{\text{FWER}}(\delta) = \delta |\mathcal{M}|$. This quantity can be interpreted as the expected number of false positives under the assumption that no pattern truly carries information about the class labels, in which case $\mathcal{M}_0 = \mathcal{M}$ would hold. Provided that the total number of candidate patterns $|\mathcal{M}|$ is known, the computational complexity of evaluating $\widehat{\text{FWER}}(\delta)$ is completely negligible. Most importantly, it can be readily shown that $\widehat{\text{FWER}}(\delta)$ is an upper bound of $\text{FWER}(\delta)$:

$$\text{FWER}(\delta) = \Pr \left(\bigcup_{\mathcal{S} \in \mathcal{M}_0} \{p_{\mathcal{S}} \leq \delta\} \right) \leq \sum_{\mathcal{S} \in \mathcal{M}_0} \Pr(p_{\mathcal{S}} \leq \delta) \leq \delta |\mathcal{M}_0| \leq \delta |\mathcal{M}|, \quad (2.11)$$

where $\{p_{\mathcal{S}} \leq \delta\}$ is the event that pattern \mathcal{S} is deemed significantly associated. Obtaining the solution of (2.10) for the choice $\widehat{\text{FWER}}(\delta) = \delta |\mathcal{M}|$ is trivial. The resulting

corrected significance threshold, given by $\delta_{\text{bonf}} = \max\{\delta \mid \delta|\mathcal{M}| \leq \alpha\} = \alpha/|\mathcal{M}|$, controls the FWER at level α since $\widehat{\text{FWER}}(\delta)$ is an upper bound of $\text{FWER}(\delta)$.

In practice, $\widehat{\text{FWER}}(\delta) = \delta|\mathcal{M}|$ tends to overestimate the real value of $\text{FWER}(\delta)$ by a considerable margin. In turn, this causes the corrected significance threshold δ_{bonf} obtained using the Bonferroni correction to be often much smaller than the optimal significance threshold δ^* , leading to a sharp loss of statistical power. However, as a direct consequence of its simplicity, the Bonferroni correction has two favourable properties: (i) it requires no assumptions to guarantee control of the FWER, thus being readily applicable to any kind of data and (ii) as mentioned above, it does not introduce any computational overhead. Those aspects have made the Bonferroni correction the most popular tool to control the FWER, making the loss of statistical power it entails a price happily paid by many practitioners.

While the Bonferroni correction has been used extensively to rigorously analyse experimental findings in many distinct disciplines, it is unfortunately unable to cope with the magnitude of the multiple comparisons problem that arises in significant pattern mining. The inadequacy of the Bonferroni correction in this setting goes beyond a mere loss of statistical power. For typical sizes of the search space $|\mathcal{M}|$, the resulting corrected significance threshold δ_{bonf} would be indistinguishable from zero using standard floating point arithmetic, leading to a trivial algorithm that would always consider no pattern to be significantly associated. Due to the lack of approaches able to control the FWER when such an enormous number of associations tests are simultaneously performed, traditional pattern mining methods resorted to either (1) ignore the multiple comparisons problem altogether, providing a ranking of patterns by association without any statistical guarantees [33, 62, 63] or (2) limit the size of the search space \mathcal{M} *a priori* by introducing implicit or explicit constraints in the maximum pattern size, in order to be able to apply a Bonferroni correction in the much smaller resulting search space [40–42].

Tarone's improved Bonferroni correction for discrete data

Controlling the FWER in significant pattern mining while keeping all candidate patterns in the search space, without any soft or hard constraints in the maximum pattern size, had remained an open problem until the Limitless-Arity Multiple-testing Procedure (LAMP) algorithm was proposed [26]. LAMP relies on Tarone's improved Bonferroni correction for discrete data [64], an alternative approach to control the FWER that drastically improves statistical power over the Bonferroni correction in pattern mining problems.

Tarone's method exploits the nature of test statistics for discrete data to derive a novel upper bound of the FWER that is much closer to the real FWER than the bound provided by a standard Bonferroni correction. Consider a 2×2 contingency table summarising n i.i.d. realisations of a pair of binary random variables with the margins n_1 and r_S treated as constants. To be consistent with the fixed margins, the count a_S must be smaller or equal than $a_{S,\text{max}} = \min(n_1, r_S)$ and larger or equal than $a_{S,\text{min}} = \max(0, r_S - (n - n_1))$. Therefore, a_S can only take $a_{S,\text{max}} - a_{S,\text{min}} + 1$ different values. In turn, this implies that there are at most $a_{S,\text{max}} - a_{S,\text{min}} + 1$ distinct P-values that can be obtained as an outcome of applying a test statistic such as Pearson's

χ^2 test or Fisher's exact test to the 2×2 contingency table. Since there is a finite number of P-values that can be observed, the P-values arising from such discrete test statistics cannot be arbitrarily small. Instead, there exists a *minimum attainable P-value*, $p_{S,\min} = \min \{p_S(a'_S \mid n, n_1, r_S) \mid a'_S \in \llbracket a_{S,\min}, a_{S,\max} \rrbracket\}$ ¹, where $p_S(a'_S \mid n, n_1, r_S)$ is the P-value obtained by applying the test statistic of choice to a 2×2 contingency table with count a'_S , margins n_1 and r_S and sample size n . The existence of a minimum attainable P-value $p_{S,\min}$ strictly larger than zero is a special property of discrete data. In contrast, P-values obtained when testing the association between two continuous random variables could be arbitrarily close to zero. Besides, the minimum attainable P-value $p_{S,\min}$ depends only on n , n_1 and r_S . In particular, $p_{S,\min}$ does *not* depend on the actual value of the count a_S .

The existence of a minimum attainable P-value $p_{S,\min}$ strictly larger than zero has profound implications. Suppose that the minimum attainable P-value $p_{S,\min}$ is larger than the corrected significance threshold, $p_{S,\min} > \delta$. By definition, regardless of the value of a_S , the corresponding association cannot be deemed statistically significant. Thus, it can also never cause a false positive at corrected significance threshold δ . Patterns $\mathcal{S} \in \mathcal{M}$ that satisfy this property are said to be *untestable* at level δ while the remaining patterns are said to be *testable* at level δ . Let $\mathcal{M}_{\text{test}}(\delta) = \{\mathcal{S} \in \mathcal{M} \mid p_{S,\min} \leq \delta\}$ be the set of testable patterns at level δ . Tarone's improved Bonferroni correction for discrete data substitutes the unknown exact value of $\text{FWER}(\delta)$ by $\widehat{\text{FWER}}(\delta) = \delta |\mathcal{M}_{\text{test}}(\delta)|$, which is also an upper bound on $\text{FWER}(\delta)$:

$$\begin{aligned}
 \text{FWER}(\delta) &= \Pr \left(\bigcup_{\mathcal{S} \in \mathcal{M}_0} \{p_S \leq \delta\} \right) = \Pr \left(\bigcup_{\mathcal{S} \in \mathcal{M}_{\text{test}}(\delta)} \{p_S \leq \delta\} \right) \\
 &\leq \sum_{\mathcal{S} \in \mathcal{M}_{\text{test}}(\delta)} \Pr(p_S \leq \delta) \leq \delta |\mathcal{M}_{\text{test}}(\delta)|,
 \end{aligned} \tag{2.12}$$

where the first step follows from the fact that untestable patterns $\mathcal{S} \in \mathcal{M} \setminus \mathcal{M}_{\text{test}}(\delta)$ cannot cause a false positive at corrected significance threshold δ . Thus, Tarone's method also guarantees FWER control without introducing any additional assumptions.

In practice, the number of testable patterns $|\mathcal{M}_{\text{test}}(\delta)|$ is drastically smaller than the total number of candidate patterns $|\mathcal{M}|$. Thus, in significant pattern mining, Tarone's concept of testability leads to a corrected significance threshold $\delta_{\text{tar}} = \max \{\delta \mid \delta |\mathcal{M}_{\text{test}}(\delta)| \leq \alpha\}$ which is much closer to the optimal significance threshold δ^* than δ_{bonf} , bringing forth a dramatic gain of statistical power over the standard Bonferroni correction and making significant pattern mining on the entire search space \mathcal{M} statistically feasible. However, while computing δ_{bonf} is straightforward, the computation of δ_{tar} is considerably more involved, requiring the use of sophisticated data mining approaches. It is precisely at the intersection between statistical association testing for discrete data and classical data mining techniques where significant pattern mining thrives, as will be discussed in the next chapter.

¹. Throughout this thesis, we will use double brackets to denote a range of consecutive integers, i.e. $\llbracket a, b \rrbracket = \{a, a+1, \dots, b\} \subset \mathbb{Z}$, where $a \leq b$.

As discussed in the previous chapter, significant pattern mining is statistically challenging due to the massive multiple comparisons problem it entails. A meaningful way to compensate for the multiple comparisons problem is to guarantee that the FWER, the probability of reporting any false positives among all association tests being performed, is bounded from above by a user-defined level α . Despite the existence of more sophisticated procedures, its simplicity and generality have helped establish the Bonferroni correction as the most popular approach to control the FWER across many scientific disciplines. However, its strengths come at a price: it is an over-conservative method that sacrifices a considerable amount of statistical power, specially in situations where the number of simultaneous association tests is large. Significant pattern mining, which takes the number of association tests to unprecedented extremes, is thus too much of a challenge for the Bonferroni correction. Tarone’s method drastically alleviates that limitation by exploiting the notion of *testability*, the phenomenon that only a subset $\mathcal{M}_{\text{test}}(\delta) \subseteq \mathcal{M}$ of all candidate patterns, the so-called *testable patterns*, can reach significance and, therefore, cause a false positive. Significant pattern mining and testability have a specially strong synergy. In practice, a large proportion of the search space is untestable ($|\mathcal{M}_{\text{test}}(\delta)| \ll |\mathcal{M}|$), making Tarone’s method particularly effective in this application.

Despite the apparent virtues of Tarone’s method in the context of significant pattern mining, a fundamental problem remains open. Computing Tarone’s corrected significance threshold δ_{tar} requires finding the largest $\delta \in [0, 1]$ that satisfies $\delta |\mathcal{M}_{\text{test}}(\delta)| \leq \alpha$. A brute-force approach to evaluate δ_{tar} would involve enumerating every single pattern \mathcal{S} in the search space \mathcal{M} to compute all minimum attainable P-values $\{p_{\mathcal{S}, \min} \mid \mathcal{S} \in \mathcal{M}\}$. Unfortunately, the vast size of the search space \mathcal{M} renders this strategy unfeasible except when dealing with extraordinarily small datasets. This computational challenge acted as a strong deterrent for the use of Tarone’s method in significant pattern mining. More than 20 years after the publication of Tarone’s method, the Limitless-Arity Multiple-testing Procedure (LAMP) algorithm [26] proposed the first effective solution to this problem, *de-facto* kickstarting the field of significant pattern mining. LAMP makes use of very specific properties of certain test statistics, which include Pearson’s χ^2 test and Fisher’s exact test, to devise a pruning criterion as part of a branch-and-bound algorithm to efficiently compute δ_{tar} while only explicitly enumerating a rather small subset of the search space \mathcal{M} . Since its publication in 2013, follow-up work [65, 66] has considerably improved the efficiency of the original LAMP algorithm while still making use of the same core principles, leading to a method that will be informally referred to as LAMP 2.0.

The goal of this chapter is to present the essential algorithmic aspects of significant pattern mining, describing a generic algorithm that incorporates all these recent developments. Thus, this chapter introduces the foundations upon which the contributions discussed in subsequent chapters have been built.

3.1 OVERVIEW

Conceptually, significant pattern mining algorithms can be understood as proceeding in two phases. This idea is represented by the pseudocode in Algorithm 3.1.

Algorithm 3.1 Significant Pattern Mining

Input: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, target FWER α

Output: $\{\mathcal{S} \in \mathcal{M} \mid p_{\mathcal{S}} \leq \delta_{\text{tar}}\}$

1: $(\delta_{\text{tar}}, \mathcal{M}_{\text{test}}(\delta_{\text{tar}})) \leftarrow \text{tarone_spm}(\mathcal{D}, \alpha)$

2: Return $\{\mathcal{S} \in \mathcal{M}_{\text{test}}(\delta_{\text{tar}}) \mid p_{\mathcal{S}} \leq \delta_{\text{tar}}\}$

The goal of the first step, carried out by the routine `tarone_spm` invoked in Line 1 of Algorithm 3.1, is to compute Tarone’s corrected significance threshold δ_{tar} . This routine, described in detail in Algorithm 3.2 below, is the most critical part of the entire algorithm. It is responsible for exploring the search space \mathcal{M} using a branch-and-bound approach in order to efficiently obtain δ_{tar} and the resulting set of testable patterns $\mathcal{M}_{\text{test}}(\delta_{\text{tar}})$. Based on its output, the second phase of Algorithm 3.1, executed in Line 2, uses a test statistic such as Pearson’s χ^2 test or Fisher’s exact test to compute the P-value $p_{\mathcal{S}}$ for all testable patterns $\mathcal{S} \in \mathcal{M}_{\text{test}}(\delta_{\text{tar}})$. Finally, the set of testable patterns deemed significantly associated at level δ_{tar} , i.e. $\{\mathcal{S} \in \mathcal{M}_{\text{test}}(\delta_{\text{tar}}) \mid p_{\mathcal{S}} \leq \delta_{\text{tar}}\}$, is returned as an output. Tarone’s concept of testability implies that untestable patterns $\mathcal{S} \in \mathcal{M} \setminus \mathcal{M}_{\text{test}}(\delta_{\text{tar}})$ cannot be significant. Hence, they do not need to be taken into consideration at this step, greatly alleviating the overall computational burden of this part of the algorithm.

3.2 PATTERN ENUMERATION

In order to efficiently explore the search space of candidate patterns \mathcal{M} , each pattern $\mathcal{S} \in \mathcal{M}$ will be arranged as a node of a *pattern enumeration tree*, which the routine `tarone_spm` then traverses recursively. A valid pattern enumeration tree is any bijective mapping of patterns to nodes of a tree which satisfies the following property: the descendants \mathcal{S}' of a pattern \mathcal{S} must all be super-patterns of \mathcal{S} , i.e., $\mathcal{S} \subseteq \mathcal{S}'$. Pattern enumeration trees are extensively used in many instances of pattern mining [67], including both significant itemset mining and significant subgraph mining. As an example, Figure 3.1 depicts one of the many possible ways to construct a pattern enumeration tree in a significant itemset mining problem with $p = 5$ binary features. In this case, each node in the tree corresponds to a feature interaction \mathcal{S} and children \mathcal{S}' of a feature interaction \mathcal{S} are obtained by incorporating an additional feature to the interaction. Note that this construction is not unique since, depending on the ordering of the features, multiple equally valid pattern enumeration trees could be obtained. Analogously, in significant subgraph mining, each node in the tree corresponds to a subgraph \mathcal{S} and children \mathcal{S}' of \mathcal{S} can be obtained by aggregating additional nodes and edges to subgraph \mathcal{S} .

A direct consequence of enumerating patterns by traversing a pattern enumeration tree is the so-called *apriori property* of pattern mining. Despite being self-evident

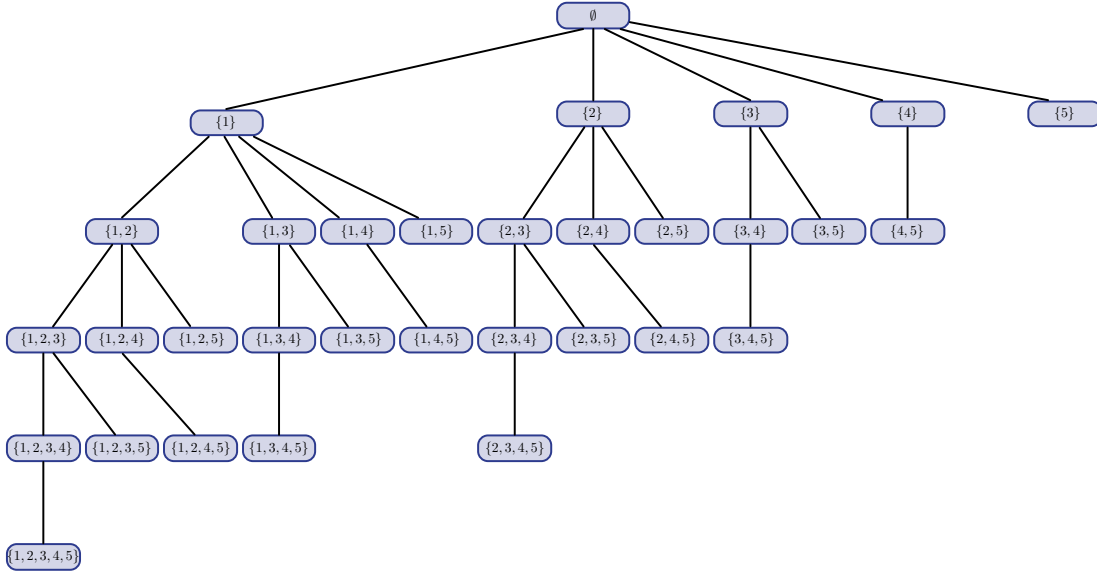


Figure 3.1. – Illustration of a valid pattern enumeration tree for a significant itemset mining problem with $p = 5$ binary features. Each of the $|\mathcal{M}| = 2^5$ feature interactions $\mathcal{S} \in \mathcal{M}$ has been mapped to a distinct node of the tree satisfying that $\mathcal{S}' \in \text{Children}(\mathcal{S})$ implies that $\mathcal{S} \subseteq \mathcal{S}'$ for all patterns $\mathcal{S} \in \mathcal{M}$.

to a large extent, this property plays a central role in a myriad of pattern mining algorithms, including all approaches that will be discussed in this thesis.

Proposition 3.1 (Apriori property). *Let $\mathcal{S}, \mathcal{S}' \in \mathcal{M}$ be two patterns such that \mathcal{S}' is a descendant of \mathcal{S} in a pattern enumeration tree. Then, $r_{\mathcal{S}'} \leq r_{\mathcal{S}}$, where $r_{\mathcal{S}}$ and $r_{\mathcal{S}'}$ are the marginal occurrence counts of patterns \mathcal{S} and \mathcal{S}' in 2×2 contingency tables computed from a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$.*

Proof. By definition of pattern enumeration tree, if \mathcal{S}' is a descendant of \mathcal{S} then $\mathcal{S} \subseteq \mathcal{S}'$. Therefore, $\mathcal{S}' \subseteq x$ implies $\mathcal{S} \subseteq x$ or, equivalently, $g_{\mathcal{S}'}(x) = 1$ implies $g_{\mathcal{S}}(x) = 1$. Since $r_{\mathcal{S}} = \sum_{i=1}^n g_{\mathcal{S}}(x)$ and $r_{\mathcal{S}'} = \sum_{i=1}^n g_{\mathcal{S}'}(x)$ the result follows. \square

The apriori property formalises an intuitive fact. Due to the way the pattern enumeration tree is constructed, deeper levels of the tree consist of increasingly complex patterns, which are guaranteed to occur less frequently in an input dataset \mathcal{D} than its simpler antecedents. As will be described next, the routine `tarone_spm` combines this property with additional statistical considerations related to the concept of testability in order to devise a pruning criterion that allows efficiently traversing the pattern enumeration tree.

The pseudocode of the routine `tarone_spm` is described in Algorithm 3.2. As shown in Line 2, the routine commences by initialising the estimate of the corrected significance threshold $\hat{\delta}_{\text{tar}}$ to 1, the largest value δ_{tar} could possibly take, and the estimate of the set of testable patterns at level $\hat{\delta}_{\text{tar}}$ to the empty set, $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}}) \leftarrow \emptyset$. In order to compute $\delta_{\text{tar}} = \max\{\delta \mid \delta |\mathcal{M}_{\text{test}}(\delta)| \leq \alpha\}$, candidate patterns $\mathcal{S} \in \mathcal{M}$ will be explored recursively by traversing the pattern enumeration tree depth-first.

Algorithm 3.2 tarone_spm

Input: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, target FWER α
Output: Corrected significance threshold δ_{tar} and corresponding set of testable patterns $\mathcal{M}_{\text{test}}(\delta_{\text{tar}})$

```

1: function tarone_spm( $\mathcal{D}, \alpha$ )
2:   Initialise global variables  $\hat{\delta}_{\text{tar}} \leftarrow 1$  and  $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}}) \leftarrow \emptyset$ 
3:   NEXT( $\emptyset$ ) ▷ Start pattern enumeration
4:    $\delta_{\text{tar}} \leftarrow \hat{\delta}_{\text{tar}}$  and  $\mathcal{M}_{\text{test}}(\delta_{\text{tar}}) \leftarrow \widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$ 
5:   Return  $\delta_{\text{tar}}$  and  $\mathcal{M}_{\text{test}}(\delta_{\text{tar}})$ 
6: end function
7: procedure NEXT( $\mathcal{S}$ )
8:   Compute the minimum attainable P-value  $p_{\mathcal{S}, \min}$  ▷ see Section 3.3
9:   if  $p_{\mathcal{S}, \min} \leq \hat{\delta}_{\text{tar}}$  then ▷ if pattern  $\mathcal{S}$  is testable at level  $\hat{\delta}_{\text{tar}}$  then
10:     Append  $\mathcal{S}$  to  $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$ 
11:      $\widetilde{\text{FWER}}(\hat{\delta}_{\text{tar}}) \leftarrow \hat{\delta}_{\text{tar}} | \widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}}) |$ 
12:     while  $\widetilde{\text{FWER}}(\hat{\delta}_{\text{tar}}) > \alpha$  do
13:       Decrease  $\hat{\delta}_{\text{tar}}$ 
14:        $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}}) \leftarrow \{ \mathcal{S}' \in \widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}}) \mid p_{\mathcal{S}', \min} \leq \hat{\delta}_{\text{tar}} \}$ 
15:        $\widetilde{\text{FWER}}(\hat{\delta}_{\text{tar}}) \leftarrow \hat{\delta}_{\text{tar}} | \widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}}) |$ 
16:   if not pruning_condition( $\mathcal{S}, \hat{\delta}_{\text{tar}}$ ) then ▷ see Section 3.4
17:     for  $\mathcal{S}' \in \text{Children}(\mathcal{S})$  do
18:       NEXT( $\mathcal{S}'$ ) ▷ Recursively visit nodes in the tree depth-first
19: end procedure

```

The estimates $\hat{\delta}_{\text{tar}}$ and $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$ will be adjusted incrementally as patterns are enumerated in such a way that, at the end of the execution of the algorithm, $\hat{\delta}_{\text{tar}} = \delta_{\text{tar}}$ and $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}}) = \mathcal{M}_{\text{test}}(\delta_{\text{tar}})$ holds. This enumeration procedure is initiated in Line 3 at the root of the tree, which by convention is assumed to represent the empty pattern¹ $\mathcal{S} = \emptyset$. For each pattern $\mathcal{S} \in \mathcal{M}$ visited during the enumeration process, the following sequence of steps is carried out.

Firstly, the minimum attainable P-value $p_{\mathcal{S}, \min}$ of the pattern \mathcal{S} is computed according to the samples in the dataset \mathcal{D} . This step, performed in Line 8 of the algorithm, will be described in detail in Section 3.3 below. In the next line, the algorithm verifies whether the pattern is testable at the current significance threshold $\hat{\delta}_{\text{tar}}$ (i.e., $p_{\mathcal{S}, \min} \leq \hat{\delta}_{\text{tar}}$) or not. If \mathcal{S} is testable, it will be aggregated to the estimate of the set of testable patterns $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$ in Line 10. Tarone's upper bound on the FWER will be evaluated next in Line 11, using the current estimate $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$ of the set of testable patterns at level $\hat{\delta}_{\text{tar}}$. In Line 12, this value is subsequently used by the algorithm to check if the FWER condition $\hat{\delta}_{\text{tar}} | \widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}}) | = \widetilde{\text{FWER}}(\hat{\delta}_{\text{tar}}) \leq \alpha$ is violated at level $\hat{\delta}_{\text{tar}}$. Note that, as the enumeration process is not yet completed, $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}}) \subseteq \mathcal{M}_{\text{test}}(\hat{\delta}_{\text{tar}})$ holds. Consequently, the FWER approximation $\widetilde{\text{FWER}}(\hat{\delta}_{\text{tar}})$ evaluated in Line 11 satisfies

1. The empty pattern is defined to occur in every input sample, i.e. $g_{\emptyset}(x) = 1$ for all $x \in \mathcal{X}$. Hence, the empty pattern cannot be statistically significant; its only purpose is to act as the starting point of the recursive enumeration.

$\widehat{\text{FWER}}(\hat{\delta}_{\text{tar}}) = \hat{\delta}_{\text{tar}} |\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})| \leq \hat{\delta}_{\text{tar}} |\mathcal{M}_{\text{test}}(\hat{\delta}_{\text{tar}})| = \widehat{\text{FWER}}(\hat{\delta}_{\text{tar}})$. If $\widehat{\text{FWER}}(\hat{\delta}_{\text{tar}}) > \alpha$, then it follows that $\widehat{\text{FWER}}(\hat{\delta}_{\text{tar}}) > \alpha$, implying that the current estimate of the corrected significance threshold $\hat{\delta}_{\text{tar}}$ is too large and violates the FWER condition for target FWER α . Thus, in that case, $\hat{\delta}_{\text{tar}}$ is decreased in Line 13. This in turn causes some patterns currently in $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$ to no longer be testable. These patterns are removed from $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$ in Line 14, thereby reducing $\widehat{\text{FWER}}(\hat{\delta}_{\text{tar}})$, which is re-evaluated in Line 15. This process is repeated, iteratively reducing $\hat{\delta}_{\text{tar}}$ and removing untestable patterns from $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$, until the FWER condition is satisfied again, i.e. $\widehat{\text{FWER}}(\hat{\delta}_{\text{tar}}) \leq \alpha$. Finally, the enumeration process continues recursively by visiting the children of the pattern \mathcal{S} currently being processed (Lines 17-18). Nevertheless, prior to that a pruning condition is evaluated in Line 16. This step, discussed in detail in Section 3.4 below, is the key to the computational feasibility of the algorithm. By construction, if the pruning condition applies, no descendant of pattern \mathcal{S} can be testable and, hence, they do not need to be enumerated, drastically reducing computational complexity. As the algorithm enumerates patterns, $\hat{\delta}_{\text{tar}}$ progressively decreases, making more and more patterns become untestable and the pruning condition in Line 16 to become more stringent. Eventually, the algorithm terminates its execution when all patterns $\mathcal{S} \in \mathcal{M}$ that have not been pruned from the search space have been visited. At termination, $\hat{\delta}_{\text{tar}} = \delta_{\text{tar}}$ and $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}}) = \mathcal{M}_{\text{test}}(\delta_{\text{tar}})$, allowing the exact value of Tarone's corrected significance threshold δ_{tar} and the corresponding set of testable patterns $\mathcal{M}_{\text{test}}(\delta_{\text{tar}})$ to be returned in Line 5.

The significant pattern mining approach described by Algorithms 3.1 and 3.2 can be applied as long as the search space of patterns \mathcal{M} can be arranged as a valid pattern enumeration tree, allowing to abstract away the specific type of patterns under consideration, be it itemsets, subgraphs or any other. While the framework is also in principle agnostic to the choice of test statistic, the two key steps of the algorithm which remain to be discussed, the computation of minimum attainable P-values $p_{\mathcal{S},\min}$ and the design of a valid pruning condition, are closely intertwined with the particular test statistic being used to assess the significance of the occurrence of patterns in a sample. The remainder of this section will be devoted to discuss in detail each of these two key steps for the two test statistics that were introduced in Section 2.2: Pearson's χ^2 test and Fisher's exact test.

3.3 EVALUATING TARONE'S MINIMUM ATTAINABLE P-VALUE

As described in Section 2.3, discrete test statistics can only result in a finite number of distinct P-values, implying the existence of a minimum attainable P-value. In particular, if a test statistic based on 2×2 contingency tables conditions on the observed margins n_1 and $r_{\mathcal{S}}$, modelling them as constants, the cell count $a_{\mathcal{S}}$ can only take values in the range $\llbracket a_{\mathcal{S},\min}, a_{\mathcal{S},\max} \rrbracket$, where $a_{\mathcal{S},\min} = \max(0, r_{\mathcal{S}} - (n - n_1))$ and $a_{\mathcal{S},\max} = \min(n_1, r_{\mathcal{S}})$. Any other outcome of $a_{\mathcal{S}}$ would be inconsistent with the observed margins n_1 and $r_{\mathcal{S}}$ and can be thus be ruled out *a priori*. Therefore, such a test can result in at most $a_{\mathcal{S},\max} - a_{\mathcal{S},\min} + 1$ different P-values, the smallest one being the minimum attainable P-value:

$$p_{\mathcal{S},\min} = \min \{ p_{\mathcal{S}}(a'_{\mathcal{S}} \mid n, n_1, r_{\mathcal{S}}) \mid a'_{\mathcal{S}} \in \llbracket a_{\mathcal{S},\min}, a_{\mathcal{S},\max} \rrbracket \}, \quad (3.1)$$

where $p_S(a'_S | n, n_1, r_S)$ is the P-value that results as an outcome of applying the test statistic to a 2×2 contingency table with cell count a'_S , margins n_1 and r_S and sample size n .

Algorithm 3.2 requires computing the minimum attainable P-value $p_{S,\min}$ of every single pattern S that is enumerated. Hence, billions or even trillions of evaluations of $p_{S,\min}$ will be necessary in a single execution of the algorithm, rendering the computational efficiency of this step critical for the overall feasibility of the entire approach. A naive application of Equation (3.1) would involve computing $p_S(a'_S | n, n_1, r_S)$ for each $a'_S \in \llbracket a_{S,\min}, a_{S,\max} \rrbracket$, leading to $O(n)$ P-value computations for each evaluation of $p_{S,\min}$, constituting an unacceptable computational overhead. This section will be devoted to introduce closed-form expressions of $p_{S,\min}$ for both Pearson's χ^2 test and Fisher's exact test that can be evaluated with $O(1)$ complexity.

The minimum attainable P-value $p_{S,\min}$ is a function of the number r_S of occurrences of pattern S in \mathcal{D} , the number n_1 of samples in \mathcal{D} that belong to the positive class and the total sample size n . However, given an input dataset \mathcal{D} , only r_S varies from pattern to pattern. In order to simplify the notation used in the remainder of this section, we will therefore simply write $p_{S,\min} = p_{\min}(r_S)$, leaving the dependence of $p_{S,\min}$ on n_1 and n implicit.

As Equation (3.1) suggests, the task of computing $p_{\min}(r_S)$ is equivalent to finding the minimiser a_S^* of $p_S(a'_S | n, n_1, r_S)$ in $\llbracket a_{S,\min}, a_{S,\max} \rrbracket$. As a consequence of the way the P-values for Pearson's χ^2 test (Equation (2.6)) and Fisher's exact test (Equation (2.8)) are defined, the minimiser a_S^* must lie in the boundary of $\llbracket a_{S,\min}, a_{S,\max} \rrbracket$, i.e. either $a_S^* = a_{S,\min}$ or $a_S^* = a_{S,\max}$. To derive a closed-form expression of $p_{\min}(r_S)$ for each of the two test statistics under consideration, all that remains to be shown is which of the two cases holds for each value of r_S .

Proposition 3.2 (Minimum attainable P-value function for Pearson's χ^2 test). *Define $n_a = \min(n_1, n - n_1)$ and $n_b = \max(n_1, n - n_1)$. Then, the minimum attainable P-value function for Pearson's χ^2 test is given by:*

$$p_{\min}(r_S) = \begin{cases} 1 - F_{\chi_1^2} \left((n-1) \frac{n_b}{n_a} \frac{r_S}{n-r_S} \right), & \text{if } 0 \leq r_S < n_a, \\ 1 - F_{\chi_1^2} \left((n-1) \frac{n_a}{n_b} \frac{n-r_S}{r_S} \right), & \text{if } n_a \leq r_S < \frac{n}{2}, \\ 1 - F_{\chi_1^2} \left((n-1) \frac{n_a}{n_b} \frac{r_S}{n-r_S} \right), & \text{if } \frac{n}{2} \leq r_S < n_b, \\ 1 - F_{\chi_1^2} \left((n-1) \frac{n_b}{n_a} \frac{n-r_S}{r_S} \right), & \text{if } n_b \leq r_S \leq n. \end{cases} \quad (3.2)$$

Proof. Let $T_{\max}(r_S)$ be the maximum value of Pearson's χ^2 test statistic for a 2×2 contingency table with sample size n and margins r_S and n_1 . As discussed above, $T_{\text{pearson}}(a_S | n, n_1, r_S)$ will be maximised either at $a_S^* = a_{S,\min}$ or at $a_S^* = a_{S,\max}$. Hence:

$$T_{\max}(r_S) = \frac{\max \left((a_{S,\min} - r_S \frac{n_1}{n})^2, (a_{S,\max} - r_S \frac{n_1}{n})^2 \right)}{\frac{r_S}{n} \frac{n-r_S}{n} \frac{n-n_1}{n-1} n_1}. \quad (3.3)$$

Suppose $0 \leq r_S < n_a$. Then, $a_{S,\min} = 0$ and $a_{S,\max} = r_S$, leading to:

$$T_{\max}(r_S) = (n-1) \frac{r_S}{n-r_S} \frac{\max^2(n_1, n-n_1)}{n_1(n-n_1)} = (n-1) \frac{r_S}{n-r_S} \frac{n_b}{n_a}. \quad (3.4)$$

Analogously, if $n_b \leq r_S \leq n$, then $a_{S,\min} = r_S - (n - n_1)$ and $a_{S,\max} = n_1$. Thus:

$$T_{\max}(r_S) = (n-1) \frac{n-r_S}{r_S} \frac{\max^2(n_1, n-n_1)}{n_1(n-n_1)} = (n-1) \frac{n-r_S}{r_S} \frac{n_b}{n_a}. \quad (3.5)$$

Finally, suppose $n_a \leq r_S < n_b$. This case can be studied separately depending on whether $n_1 \leq n - n_1$ or $n_1 > n - n_1$.

Let $n_1 \leq n - n_1$, then $a_{S,\min} = 0$ and $a_{S,\max} = n_1$. This leads to:

$$\begin{aligned} T_{\max}(r_S) &= (n-1) \frac{n_1}{n-n_1} \frac{\max^2(r_S, n-r_S)}{r_S(n-r_S)} = (n-1) \frac{n_a}{n_b} \frac{\max^2(r_S, n-r_S)}{r_S(n-r_S)} \\ &= \begin{cases} (n-1) \frac{n_a}{n_b} \frac{n-r_S}{r_S}, & \text{if } n_a \leq r_S < \frac{n}{2}, \\ (n-1) \frac{n_a}{n_b} \frac{r_S}{n-r_S}, & \text{if } \frac{n}{2} \leq r_S < n_b. \end{cases} \end{aligned} \quad (3.6)$$

If $n_1 > n - n_1$, then $a_{S,\min} = r_S - (n - n_1)$ and $a_{S,\max} = r_S$. Therefore:

$$\begin{aligned} T_{\max}(r_S) &= (n-1) \frac{n-n_1}{n_1} \frac{\max^2(r_S, n-r_S)}{r_S(n-r_S)} = (n-1) \frac{n_a}{n_b} \frac{\max^2(r_S, n-r_S)}{r_S(n-r_S)} \\ &= \begin{cases} (n-1) \frac{n_a}{n_b} \frac{n-r_S}{r_S}, & \text{if } n_a \leq r_S < \frac{n}{2}, \\ (n-1) \frac{n_a}{n_b} \frac{r_S}{n-r_S}, & \text{if } \frac{n}{2} \leq r_S < n_b. \end{cases} \end{aligned} \quad (3.7)$$

Since $p_{\text{pearson}}(a_S \mid n, n_1, r_S) = 1 - F_{\chi_1^2}(T_{\text{pearson}}(a_S \mid n, n_1, r_S))$, this concludes the proof. \square

Proposition 3.3 (Minimum attainable P-value function for Fisher's exact test). *Define $n_a = \min(n_1, n - n_1)$ and $n_b = \max(n_1, n - n_1)$. Then, the minimum attainable P-value function for Fisher's exact test is given by:*

$$p_{\min}(r_S) = \begin{cases} \binom{n_a}{r_S} / \binom{n}{r_S}, & \text{if } 0 \leq r_S < n_a, \\ \binom{n_b}{n-r_S} / \binom{n}{r_S}, & \text{if } n_a \leq r_S < \frac{n}{2}, \\ \binom{n_b}{r_S} / \binom{n}{r_S}, & \text{if } \frac{n}{2} \leq r_S < n_b, \\ \binom{n_a}{n-r_S} / \binom{n}{r_S}, & \text{if } n_b \leq r_S \leq n. \end{cases} \quad (3.8)$$

Proof. Analogously to the proof of Proposition 3.2, $p_{\text{fisher}}(a_S \mid n, n_1, r_S)$ is minimised either at $a_S^* = a_{S,\min}$ or at $a_S^* = a_{S,\max}$. Therefore:

$$p_{\min}(r_S) = \min(\text{Hypergeom}(a_{S,\min} \mid n, n_1, r_S), \text{Hypergeom}(a_{S,\max} \mid n, n_1, r_S)). \quad (3.9)$$

This implies that:

$$p_{\min}(r_S) = \frac{\min\left(\binom{n_1}{a_{S,\min}} \binom{n-n_1}{r_S-a_{S,\min}}, \binom{n_1}{a_{S,\max}} \binom{n-n_1}{r_S-a_{S,\max}}\right)}{\binom{n}{r_S}}. \quad (3.10)$$

The problem will be decomposed in three cases, as done for the derivation of the closed-form expression of the minimum attainable P-value for Pearson's χ^2 test.

Let $0 \leq r_S < n_a$, leading to $a_{S,\min} = 0$ and $a_{S,\max} = r_S$. Then:

$$p_{\min}(r_S) = \frac{\min\left(\binom{n-n_1}{r_S}, \binom{n_1}{r_S}\right)}{\binom{n}{r_S}} = \frac{\binom{n_a}{r_S}}{\binom{n}{r_S}}. \quad (3.11)$$

Similarly, if $n_b \leq r_S \leq n$ then $a_{S,\min} = r_S - (n - n_1)$ and $a_{S,\max} = n_1$. Thus:

$$p_{\min}(r_S) = \frac{\min\left(\binom{n_1}{r_S - (n - n_1)}, \binom{n - n_1}{r_S - n_1}\right)}{\binom{n}{r_S}} = \frac{\min\left(\binom{n_1}{n - r_S}, \binom{n - n_1}{n - r_S}\right)}{\binom{n}{r_S}} = \frac{\binom{n_a}{n - r_S}}{\binom{n}{r_S}}, \quad (3.12)$$

where the second step follows from $\binom{n}{k} = \binom{n}{n-k}$.

Let $n_a \leq r_S < n_b$ and $n_1 \leq n - n_1$, so that $a_{S,\min} = 0$ and $a_{S,\max} = n_1$. Hence:

$$p_{\min}(r_S) = \frac{\min\left(\binom{n - n_1}{r_S}, \binom{n - n_1}{n - r_S}\right)}{\binom{n}{r_S}} = \begin{cases} \frac{\binom{n_b}{n - r_S}}{\binom{n}{r_S}}, & \text{if } n_a \leq r_S < \frac{n}{2}, \\ \frac{\binom{n_b}{r_S}}{\binom{n}{r_S}}, & \text{if } \frac{n}{2} \leq r_S < n_b. \end{cases} \quad (3.13)$$

Alternatively, if $n_a \leq r_S < n_b$ and $n_1 > n - n_1$, then $a_{S,\min} = r_S - (n - n_1)$ and $a_{S,\max} = r_S$, leading to:

$$p_{\min}(r_S) = \frac{\min\left(\binom{n_1}{n - r_S}, \binom{n_1}{r_S}\right)}{\binom{n}{r_S}} = \begin{cases} \frac{\binom{n_b}{n - r_S}}{\binom{n}{r_S}}, & \text{if } n_a \leq r_S < \frac{n}{2}, \\ \frac{\binom{n_b}{r_S}}{\binom{n}{r_S}}, & \text{if } \frac{n}{2} \leq r_S < n_b, \end{cases} \quad (3.14)$$

where again the identity $\binom{n}{k} = \binom{n}{n-k}$ was used. This concludes the proof. \square

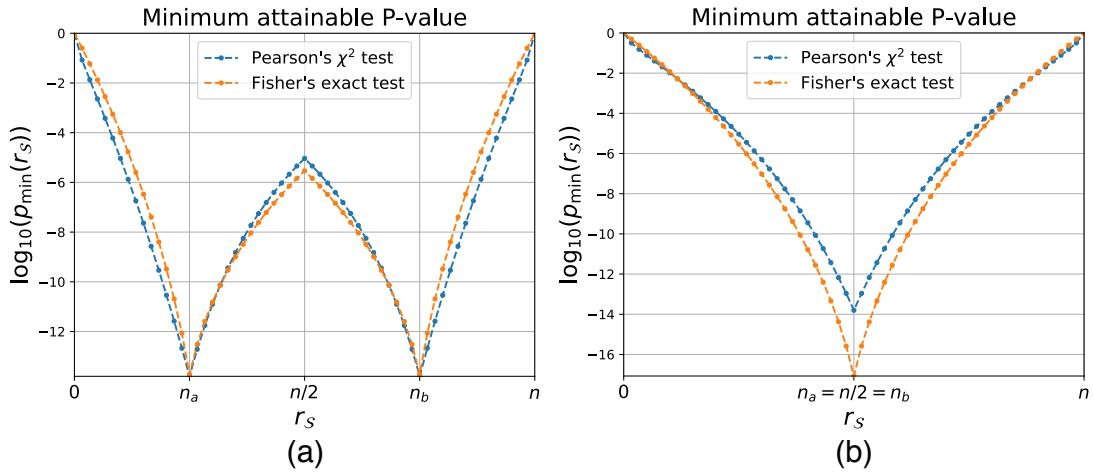


Figure 3.2. – Minimum attainable P-value $p_{S,\min}$ for Pearson's χ^2 test (blue) and Fisher's exact test (orange) as the number r_S of occurrences of pattern S in a dataset \mathcal{D} varies. The number of samples in the positive class n_1 and the sample size n are $n_1 = 15$, $n = 60$ in (a) and $n_1 = 30$, $n = 60$ in (b), respectively.

Figure 3.2 depicts $p_{S,\min}$ as a function of the number r_S of occurrences of pattern S in a dataset \mathcal{D} for Pearson's χ^2 test (blue) and Fisher's exact test (orange). Two particular cases are illustrated in Figures 3.2(a) and 3.2(b) depending on the class ratio n_1/n of the dataset: the former exemplifies an unbalanced dataset with $n_1 = 15$ and $n = 60$ whereas the latter corresponds to a balanced dataset with $n_1 = 30$ and $n = 60$. The qualitative behaviour of $p_{\min}(r_S)$ is identical for both test statistics. As suggested

by the closed-form expressions derived in Propositions 3.2 and 3.3, the minimum attainable P-value $p_{S,\min}$ as a function of r_S is symmetric around $r_S = n/2$ and has minima at $r_S = n_a = \min(n_1, n - n_1)$ and $r_S = n_b = \max(n_1, n - n_1)$. Informally, the fundamental property of $p_{\min}(r_S)$ evidenced by Figure 3.2 is that the minimum attainable P-value $p_{S,\min}$ is large, indicating lower potential to result in a statistically significant association, whenever r_S is small or r_S is large.

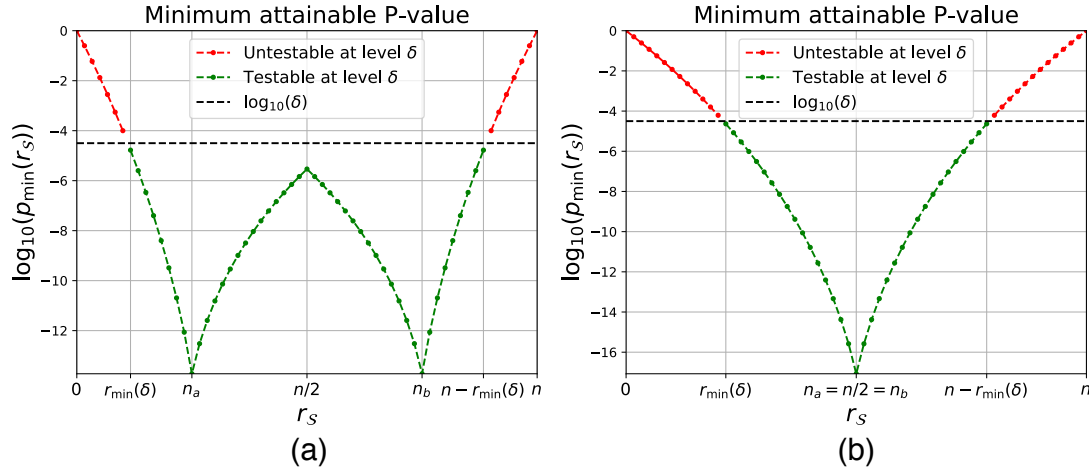


Figure 3.3. – Illustration of the concept of testability at an arbitrary level $\delta = 10^{-4.5}$ when Fisher's exact test is the test statistic of choice for two 2×2 contingency tables differing in their class ratios: (a) unbalanced, with $n_1 = 15$, $n = 60$ and (b) balanced, with $n_1 = 30$, $n = 60$. Values of r_S in the range $[[r_{\min}(\delta), n - r_{\min}(\delta)]]$ lead to pattern \mathcal{S} being testable at level δ (green), while $r_S < r_{\min}(\delta)$ or $r_S > n - r_{\min}(\delta)$ imply that pattern \mathcal{S} is untestable at level δ (red).

This property is investigated further in Figure 3.3, which illustrates how it relates to the concept of testability at a certain corrected significance threshold δ . As a direct consequence of the functional form of $p_{S,\min}$, for fixed margin n_1 , sample size n and corrected significance threshold δ , there exists a value $r_{\min}(\delta)$ such that patterns \mathcal{S} with $r_S < r_{\min}(\delta)$ or $r_S > n - r_{\min}(\delta)$ are untestable at level δ , while those for which r_S lies in $[[r_{\min}(\delta), n - r_{\min}(\delta)]]$ are testable². This formalises the intuition that patterns \mathcal{S} for which r_S is too small or too large, i.e. patterns \mathcal{S} that are either too rare or too common in \mathcal{D} , are less likely to be significantly associated. Tarone's method can thus be understood as a statistically principled way to turn this intuition into a filtering criterion to reduce the number of patterns that contribute to the multiple comparisons problem. However, unlike other alternative approaches, Tarone's method does not use an ad-hoc threshold to filter patterns according to their number of occurrences in the input dataset. Instead, it learns an adaptive threshold in a data-driven manner. As a consequence, Tarone's method is able to guarantee that all patterns that have no chance of resulting in a statistically significant association at level δ , regardless of the

2. Technically, for sufficiently small δ , the set of values of r_S that leads to pattern \mathcal{S} being testable could be the union of two disjoint intervals rather than a single interval. Nevertheless, this poses no additional statistical or algorithmic difficulties. Moreover, this situation is uncommon in practice, as it corresponds to values of δ that would be too small to be of practical relevance in most applications.

actual realisations of the class labels, will be filtered and that all patterns that could possibly result in an association will be kept.

3.4 DESIGNING A PRUNING CONDITION

Being able to evaluate the minimum attainable P-value $p_{S,\min}$ efficiently is of utmost importance for significant pattern mining. However, even if $p_{S,\min}$ can be computed with $O(1)$ complexity, evaluating the minimum attainable P-value $p_{S,\min}$ of every single candidate pattern $S \in \mathcal{M}$ remains computationally intractable in practice due to the sheer size of the search space. As described in Section 3.2, significant pattern mining algorithms circumvent this limitation by leveraging a *pruning condition*, that is, a way to test if descendants S' of a pattern S in the enumeration tree are testable using only information available in the 2×2 contingency table of pattern S . The remainder of this section will describe how specific properties of the function $p_{\min}(r_S)$ for Pearson's χ^2 test and Fisher's exact test can be combined with the apriori property of pattern mining (Proposition 3.1) to design a simple yet highly effective pruning criterion, which is summarised in the following proposition:

Proposition 3.4 (Pruning criterion for Pearson's χ^2 test and Fisher's exact test). *Let $S \in \mathcal{M}$ be a pattern satisfying:*

- (i) $p_{S,\min} > \hat{\delta}_{\text{tar}}$, i.e., S is untestable at level $\hat{\delta}_{\text{tar}}$,
- (ii) $r_S \leq n_a$, with $n_a = \min(n_1, n - n_1)$.

Then, $p_{S',\min} > \hat{\delta}_{\text{tar}} \geq \delta_{\text{tar}}$ for all descendants S' of S in the pattern enumeration tree, implying that they can be pruned from the search space. In conclusion, $\text{pruning_condition}(S, \hat{\delta}_{\text{tar}})$ in Line 16 of Algorithm 3.2 is true if and only if $r_S \leq n_a$ and $p_{S,\min} > \hat{\delta}_{\text{tar}}$.

Proof. Firstly, for fixed sample size n and number n_1 of samples in the positive class, the minimum attainable P-value $p_{S,\min}$ is a monotonically decreasing function of r_S in the range $r_S \in \llbracket 0, n_a \rrbracket$ for both Pearson's χ^2 test and Fisher's exact test, i.e., $r_{S'} \leq r_S \leq n_a$ implies $p_{\min}(r_{S'}) \geq p_{\min}(r_S)$. This property, which does not necessarily hold for all test statistics based on contingency tables, can be readily verified from the specific functional form of $p_{S,\min}$ in the range $r_S \in \llbracket 0, n_a \rrbracket$: $p_{\min}(r_S) = 1 - F_{\chi^2_1} \left((n-1) \frac{n_b r_S}{n_a n - r_S} \right)$ for Pearson's χ^2 test and $p_{\min}(r_S) = \binom{n_a}{r_S} / \binom{n}{r_S}$ for Fisher's exact test. Intuitively, this means that in the range $r_S \in \llbracket 0, n_a \rrbracket$, as the pattern S becomes less rare, the minimum attainable P-value decreases, thereby increasing the potential of pattern S to be statistically significant. Secondly, due to the apriori property, if S' is a descendant of S , then $r_{S'} \leq r_S$ holds.

Combining both facts, if $r_S \leq n_a$ then $p_{S',\min} \geq p_{S,\min}$ for all descendants S' of S in the enumeration tree. Therefore, $p_{S,\min} > \hat{\delta}_{\text{tar}}$ implies that $p_{S',\min} > \hat{\delta}_{\text{tar}}$ for all descendants S' of S . Since $\hat{\delta}_{\text{tar}} \geq \delta_{\text{tar}}$ at any point during the execution of Algorithm 3.2, this proves the result. \square

The pruning criterion presented in Proposition 3.4 only applies to patterns $S \in \mathcal{M}$ satisfying $r_S \leq n_a$. While this might seem to be a strong limitation, a large proportion of all candidate patterns S in the search space \mathcal{M} are sufficiently rare for this condition to apply. For any pattern S satisfying this constraint, Proposition 3.4 simply states

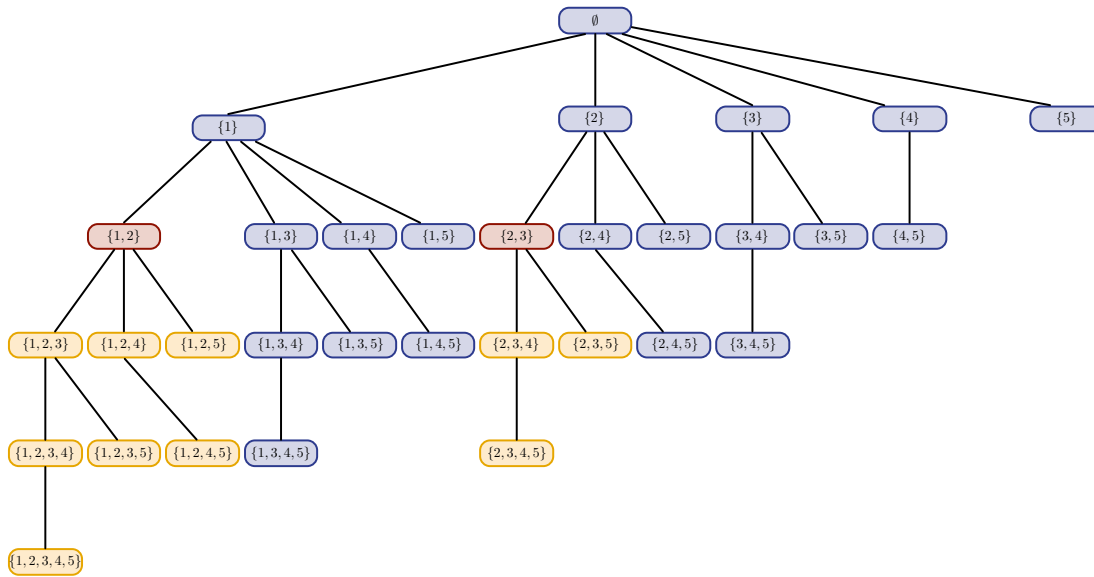


Figure 3.4. – Illustration of the effect of search space pruning in a significant itemset mining problem with $p = 5$ binary features. In this example, it is assumed that patterns \mathcal{S}_1 and \mathcal{S}_2 (highlighted in red) satisfy the conditions of Proposition 3.4, i.e. $r_{\mathcal{S}_1} \leq n_a$, $r_{\mathcal{S}_2} \leq n_a$ and $p_{\mathcal{S}_1, \min} > \hat{\delta}_{\text{tar}}$, $p_{\mathcal{S}_2, \min} > \hat{\delta}_{\text{tar}}$. Then, all their descendants (highlighted in orange) would be pruned from the search space, drastically reducing the number of candidate patterns that need to be enumerated.

that descendants \mathcal{S}' of an untestable pattern \mathcal{S} will also be untestable. As Figure 3.4 illustrates, this can lead to a drastic reduction in computational complexity, allowing to compute δ_{tar} and retrieve the set of testable patterns $\mathcal{M}_{\text{test}}(\delta_{\text{tar}})$ while enumerating only a small subset of all candidate patterns in the search space \mathcal{M} .

3.5 IMPLEMENTATION CONSIDERATIONS

When implementing the state-of-the-art significant pattern mining algorithm described in this chapter, some fundamental design choices, not discussed yet for the sake of clarity, need to be addressed. This section will be devoted to cover each of these considerations in detail.

Constructing and navigating the pattern enumeration tree

One of the aspects that tends to have the greatest impact on computational efficiency is the choice of algorithm to build and traverse the pattern enumeration tree. *A priori*, this might seem a conceptually simple task. However, the vast size of the search space quickly renders naive enumeration approaches computationally unfeasible as the size of the dataset increases, requiring instead the use of sophisticated data structures and enumeration strategies. Fortunately, the design of efficient algorithms to enumerate patterns has been a key subject of research in data mining during decades, leading to a wealth of highly optimised methods that can be readily applied to significant

pattern mining, including both itemset mining (e.g. [31, 68, 69], see [70] for a review) and subgraph mining (e.g. [71, 72], see [73] for a review).

Iterative refinement of the estimate of Tarone’s corrected significance threshold

On every occasion the FWER condition is found to be violated during the execution of Algorithm 3.2, the estimate $\hat{\delta}_{\text{tar}}$ of the corrected significance threshold is decreased in Line 13. A simple strategy to implement this step is via grid search, be it uniform ($\hat{\delta}_{\text{tar}} \leftarrow \hat{\delta}_{\text{tar}} - \Delta$) or logarithmic ($\hat{\delta}_{\text{tar}} \leftarrow 10^{-\Delta} \hat{\delta}_{\text{tar}}$). As long as Δ is sufficiently small, both approaches yield good results in practice. Alternatively, it is possible to exploit the discreteness of the function $p_{\min}(r_S)$ to devise a more efficient strategy. In this case, the sequence of candidate values for $\hat{\delta}_{\text{tar}}$ would be obtained by sorting in descending order the $\lfloor \frac{n}{2} \rfloor + 1$ different values the minimum attainable P-value function $p_{\min}(r_S)$ can take³. Every time $\hat{\delta}_{\text{tar}}$ needs to be decreased, it would be set to the next element of that sequence. This strategy is optimal in the sense that it adaptively selects the minimum step size necessary to decrease the estimate of Tarone’s FWER upper bound $\widetilde{\text{FWER}}(\hat{\delta}_{\text{tar}})$.

Eliminating the need to keep the set of testable patterns in memory

Algorithm 3.3 find_significant_patterns

Input: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, Tarone’s corrected significance threshold δ_{tar}

- 1: **function** find_significant_patterns($\mathcal{D}, \delta_{\text{tar}}$)
- 2: NEXT(\emptyset) ▷ Start pattern enumeration
- 3: **end function**
- 4: **procedure** NEXT(\mathcal{S})
- 5: Compute the minimum attainable P-value $p_{\mathcal{S}, \min}$ ▷ see Section 3.3
- 6: **if** $p_{\mathcal{S}, \min} \leq \delta_{\text{tar}}$ **then** ▷ **if** pattern \mathcal{S} is testable at level δ_{tar} **then**
- 7: Compute the P-value $p_{\mathcal{S}}$
- 8: **if** $p_{\mathcal{S}} \leq \delta_{\text{tar}}$ **then** ▷ **if** pattern \mathcal{S} is significant at level δ_{tar} **then**
- 9: Write \mathcal{S} and $p_{\mathcal{S}}$ to an output file
- 10: **if not** pruning_condition($\mathcal{S}, \delta_{\text{tar}}$) **then** ▷ see Section 3.4
- 11: **for** $\mathcal{S}' \in \text{Children}(\mathcal{S})$ **do**
- 12: NEXT(\mathcal{S}') ▷ Recursively visit nodes in the tree depth-first
- 13: **end procedure**

The estimate $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$ of the set of testable patterns at level $\hat{\delta}_{\text{tar}}$ tends to be large in practice, often containing hundreds of millions or even billions of patterns. Therefore, explicitly storing this set in memory during the execution of the algorithm can be a challenge. Moreover, this naive strategy could cause the execution of Line 14 in Algorithm 3.2 to become a computational bottleneck, as in principle the entire set $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$ would need to be inspected to remove all patterns that become untestable after having decreased $\hat{\delta}_{\text{tar}}$ in the previous step of the algorithm. Nevertheless, in order

³ The symmetry of $p_{\min}(r_S)$ around $n/2$ implies that only $\lfloor \frac{n}{2} \rfloor + 1$ distinct minimum attainable P-values can be obtained for fixed n_1 and n , rather than $n + 1$.

to compute the corrected significance threshold $\hat{\delta}_{\text{tar}}$, Algorithm 3.2 only requires access to the total number of testable patterns $|\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})|$, not the actual patterns themselves. Define $\mathcal{P}_{\text{min}} = \{p_{\text{min}}(r_S) \mid r_S \in \llbracket 0, \lfloor \frac{n}{2} \rrbracket\}$ to be the set of $\lfloor \frac{n}{2} \rfloor + 1$ distinct values that $p_{\text{min}}(r_S)$ can take for a given sample size n . Instead of explicitly storing $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$ in memory, an alternative approach is to maintain only the set of $\lfloor \frac{n}{2} \rfloor + 1$ integers $\{c(p_{S,\text{min}}) \mid p_{S,\text{min}} \in \mathcal{P}_{\text{min}}\}$, where $c(p_{S,\text{min}})$ is the number of patterns enumerated so far that have minimum attainable P-value equal to $p_{S,\text{min}}$. Given any $\hat{\delta}_{\text{tar}}$, the total number $|\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})|$ of testable patterns enumerated so far can be retrieved as:

$$|\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})| = \sum_{p_{S,\text{min}} \in \mathcal{P}_{\text{min}} \mid p_{S,\text{min}} \leq \hat{\delta}_{\text{tar}}} c(p_{S,\text{min}}), \quad (3.15)$$

where the summation includes at most $\lfloor \frac{n}{2} \rfloor + 1$ counts. This approach allows computing δ_{tar} exactly by executing Algorithm 3.2 until its termination without ever needing to store the estimate $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$ of the set of testable patterns in memory. Moreover, it also allows executing Line 14 of Algorithm 3.2 with $O(1)$ complexity, avoiding a potential computational pitfall. However, in order to find the subset of testable patterns which are significantly associated with the class labels, Line 2 of Algorithm 3.1 does require access to not only the corrected significance threshold δ_{tar} but also the actual set of testable patterns $\mathcal{M}_{\text{test}}(\delta_{\text{tar}})$. To this end, the enumeration process can be repeated, starting again at the root of the enumeration tree, but with fixed $\hat{\delta}_{\text{tar}} = \delta_{\text{tar}}$. As patterns are enumerated, the P-values p_S of patterns testable at level δ_{tar} can be computed on the fly, and those which are deemed significantly associated can be written to an output file. This approach is summarised in Algorithm 3.3. While this strategy requires enumerating patterns twice, thus approximately doubling the total runtime, it completely avoids the need to keep the set of testable patterns $\mathcal{M}_{\text{test}}(\delta_{\text{tar}})$ in memory, greatly reducing memory usage. As a consequence, this is the preferred implementation choice in most situations. Finally, it is worth noting that this strategy is also applicable in situations for which the estimate $\hat{\delta}_{\text{tar}}$ of the corrected significance threshold is decreased using grid search. For instance, if a logarithmic grid was used, it would be possible to define $c(i)$ as the number of patterns enumerated so far that have minimum attainable P-value $p_{S,\text{min}} \in [10^{-(i-1)\Delta}, 10^{-i\Delta})$ for each $1 \leq i \leq i_{\text{max}}$, where i_{max} is set so that $10^{-i_{\text{max}}\Delta}$ is small enough (e.g. 10^{-30}).

Caching the minimum attainable P-value function

In some circumstances, a useful strategy to further speed-up the evaluation of the minimum attainable P-value function $p_{\text{min}}(r_S)$ is to precompute its output for all $n + 1$ possible values of r_S in the range $\llbracket 0, n \rrbracket$ and store the mapping as a look-up table. This requires only $O(n)$ additional memory, often a negligible amount in comparison to the size of the entire dataset \mathcal{D} , and will virtually eliminate the contribution of minimum attainable P-value computations to the overall runtime of the algorithm.

Part II
CONTRIBUTIONS

Perhaps the most influential characteristic of significant pattern mining as a machine learning task is the daunting size of the search space of candidate patterns \mathcal{M} , which has profound statistical and computational implications. Most notably, during the first part of this thesis, it was discussed extensively how this causes two fundamental difficulties: (i) the statistical challenge of dealing with an extreme instance of the multiple comparisons problem, and (ii) the computational challenge of exploring this vast search space \mathcal{M} efficiently. In Chapter 3, it was shown how Tarone’s improved Bonferroni correction for discrete data can be combined with classical data mining techniques to design a practical significant pattern mining algorithm that successfully circumvents those difficulties. The resulting method is able to explore all candidate patterns in the search space \mathcal{M} in a computationally efficient manner and exhibits a considerable amount of statistical power despite guaranteeing FWER control. However, this approach is not devoid of limitations, opening the door to the development of novel algorithms to further improve the state of the art in significant pattern mining.

In particular, another defining characteristic of significant pattern mining, completely overlooked by the approach introduced in Chapter 3, is the fact that the search space of candidate patterns \mathcal{M} is not only extremely large, but also harbours non-trivial dependencies between the many patterns \mathcal{M} contains.

In this chapter we present Westfall-Young light [49], a fast and memory-efficient significant pattern mining algorithm that models the statistical dependencies between patterns in the search space, obtaining a more accurate approximation of the FWER than that provided by Tarone’s method and, ultimately, leading to a gain in statistical power.

The rest of this chapter is organised as follows. Section 4.1 details how the dependence between patterns arises in significant pattern mining, as well as the main implications of this phenomenon on the correction for the multiple comparisons problem. Section 4.2 will be devoted to introduce the *Westfall-Young permutation testing procedure*, a resampling-based approach to directly estimate the real value of $\text{FWER}(\delta)$ without the need to make simplifying independence assumptions. Next, Section 4.3 discusses the challenge of developing significant pattern mining algorithms able to make use of permutation testing. In particular, Section 4.3.1 introduces the FastWY algorithm [43], which to the best of our knowledge constitutes the only previously existing attempt to combine permutation testing and significant pattern mining, whereas Section 4.3.2 discusses in detail our novel contribution, the *Westfall-Young light* algorithm. Finally, a thorough experimental assessment of the computational efficiency of FastWY and *Westfall-Young light* is described in Section 4.4.

4.1 INTRODUCTION

The most evident source of dependencies between patterns in \mathcal{M} are subset/superset relationships. If a pattern \mathcal{S} is contained in another pattern \mathcal{S}' , the random variables

$G_S(X)$ and $G_{S'}(X)$ that indicate the occurrence of patterns S and S' in an input sample X satisfy $G_S(X) = 0 \Rightarrow G_{S'}(X) = 0$ or, equivalently, $G_{S'}(X) = 1 \Rightarrow G_S(X) = 1$. Therefore, $G_S(X)$ and $G_{S'}(X)$ are statistically dependent, being mutually redundant to some extent. Alternatively, the apriori property of pattern mining, discussed in Proposition 3.1, can also be used to show that $G_S(X)$ and $G_{S'}(X)$ are statistically dependent for $S \subset S'$. The strength of this dependency itself mainly depends on the probability that the “difference pattern” $S' \setminus S$ occurs in an input sample X . The more likely it is that this pattern occurs, the more frequently S and S' will co-occur, leading to a stronger association between $G_S(X)$ and $G_{S'}(X)$. Figure 4.1 illustrates this phenomenon in a toy significant itemset mining problem with $n = 12$ samples and $p = 10$ features. The illustration depicts feature interaction $S = \{2, 9, 10\}$, highlighted in grey, and feature interaction $S' = S \cup \{7\}$, highlighted in brown. Since S' is formed by adding an additional feature to S , it satisfies $S \subset S'$ by construction. In this example, it can be seen that knowing the value of $g_S(x_i)$ alone is sufficient to also know the value of $g_{S'}(x_i)$ for half of the samples $\{x_i\}_{i=1}^{12}$ in the dataset. In fact, this holds true for any $S' \supseteq S$, regardless of the additional features $S' \setminus S$ being added to the feature interaction. Consequently, $G_S(X)$ is statistically dependent of $G_{S'}(X)$, for any $S' \supseteq S$.

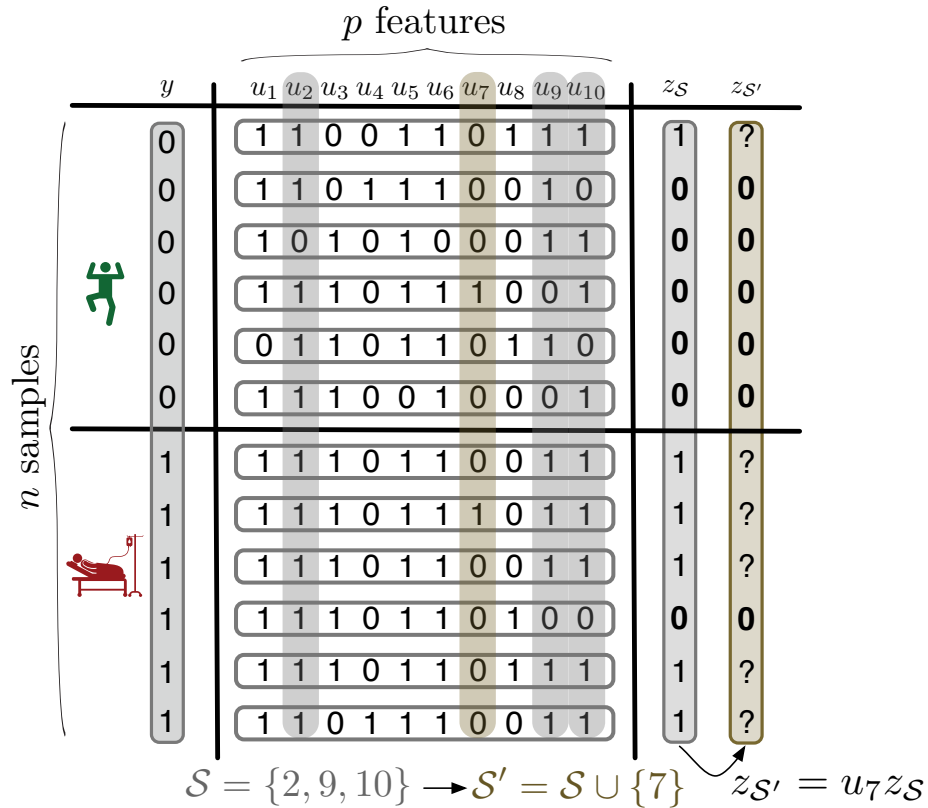


Figure 4.1. – Illustration of how dependence between patterns arises in significant itemset mining due to inclusion relationships $S \subset S'$ between candidate patterns $S, S' \in \mathcal{M}$.

Subset/superset relationships are not the only potential source of dependence between any two candidate patterns $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{M}$. More generally, as long as the two patterns share some sub-structures, i.e. $\mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset$, the random variables $G_{\mathcal{S}_1}(X)$ and $G_{\mathcal{S}_2}(X)$ might be statistically associated. Since $G_{\mathcal{S}_1}(X) = G_{\mathcal{S}_1 \cap \mathcal{S}_2}(X)G_{\mathcal{S}_1 \setminus \mathcal{S}_2}(X)$ and $G_{\mathcal{S}_2}(X) = G_{\mathcal{S}_1 \cap \mathcal{S}_2}(X)G_{\mathcal{S}_2 \setminus \mathcal{S}_1}(X)$ holds for any $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{M}$, the strength of the association between $G_{\mathcal{S}_1}(X)$ and $G_{\mathcal{S}_2}(X)$ depends on the probability that the shared sub-structure $\mathcal{S}_1 \cap \mathcal{S}_2$ occurs in an input sample X , relative to the probability that the pattern-specific sub-structures $\mathcal{S}_1 \setminus \mathcal{S}_2$ and $\mathcal{S}_2 \setminus \mathcal{S}_1$ occur. For example, if $\mathcal{S}_1 \setminus \mathcal{S}_2$ and $\mathcal{S}_2 \setminus \mathcal{S}_1$ are both common sub-structures which are present in almost every input sample X , the probability that \mathcal{S}_1 and \mathcal{S}_2 occur in an input sample X will be dominated by the probability that $\mathcal{S}_1 \cap \mathcal{S}_2$ is present in X . Therefore, in this situation $G_{\mathcal{S}_1}(X)$ and $G_{\mathcal{S}_2}(X)$ will be strongly associated. On the contrary, if $\mathcal{S}_1 \cap \mathcal{S}_2$ is a common sub-structure, the probability that \mathcal{S}_1 occurs in an input sample X will be mostly determined by how frequently $\mathcal{S}_1 \setminus \mathcal{S}_2$ occurs, while the probability that \mathcal{S}_2 occurs in an input sample X will mostly depend on how frequently $\mathcal{S}_2 \setminus \mathcal{S}_1$ occurs. In this case, $G_{\mathcal{S}_1}(X)$ and $G_{\mathcal{S}_2}(X)$ will be approximately independent.

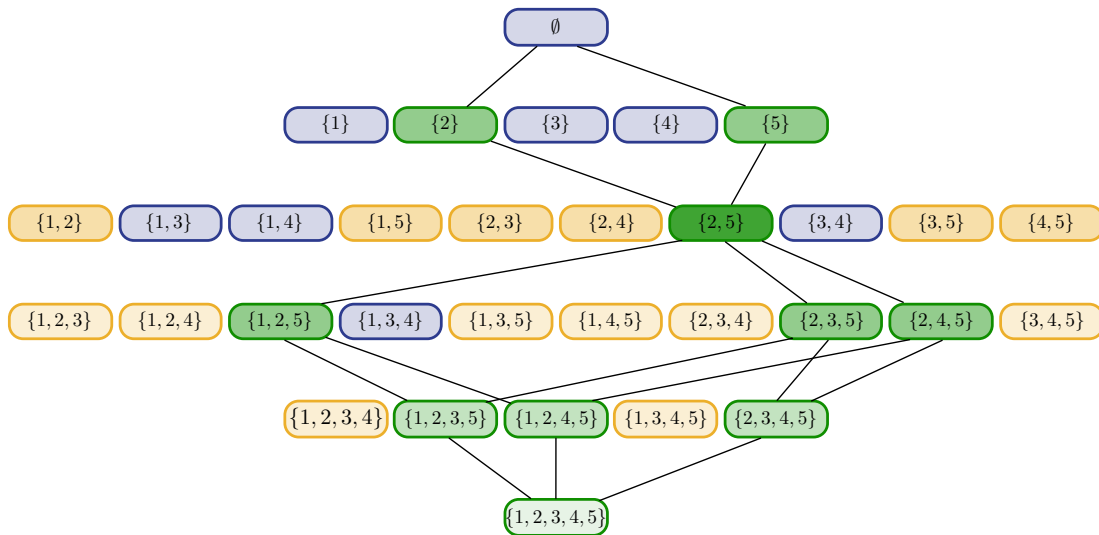


Figure 4.2. – Illustration of how subset/superset relationships between patterns and the sharing of pattern sub-structures can result in a pattern $\mathcal{S} \in \mathcal{M}$ being statistically associated with many other patterns in the search space \mathcal{M} . In this significant itemset mining example, a feature interaction $\mathcal{S} = \{2, 5\}$ (dark green) is related by subset/superset relationships to 9/32 feature interactions (green) and shares features with other 14/32 feature interactions (orange). The intensity of the colour of a node in the pattern enumeration tree is proportional to the relatedness of the corresponding feature interaction \mathcal{S}' with $\mathcal{S} = \{2, 5\}$.

Due to the combinatorial nature of the search space \mathcal{M} , these dependencies which arise as a consequence of subset/superset relationships and the sharing of pattern sub-structures can lead to any given pattern \mathcal{S} being statistically associated with a large proportion of patterns in the search space \mathcal{M} . Figure 4.2 illustrates this effect in a toy significant itemset mining problem with only $p = 5$ features. In this

example, a given feature interaction $\mathcal{S} = \{2, 5\}$ (highlighted in dark green) is directly related by subset/superset relationships to 9 out of 32 patterns (highlighted in green) and shares some sub-structures (in this case, input features) with other 14 out of 32 patterns (highlighted in orange). In summary, in this example, the presence of pattern $\mathcal{S} = \{2, 5\}$ in an input sample X might be statistically associated with occurrences of more than half of all patterns in the entire search space \mathcal{M} .

The existence of this complex web of interdependencies between patterns in the search space \mathcal{M} has profound implications. If the random variables $G_{\mathcal{S}_1}(X)$ and $G_{\mathcal{S}_2}(X)$ are statistically dependent, the corresponding P-values $p_{\mathcal{S}_1}$ and $p_{\mathcal{S}_2}$ quantifying the statistical association of $G_{\mathcal{S}_1}(X)$ and $G_{\mathcal{S}_2}(X)$ with the class labels Y might be statistically dependent as well. Suppose that patterns $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{M}$ are *not* associated with the class labels Y , i.e. $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{M}_0$ as defined in Section 2.3. Then, if the random variables $G_{\mathcal{S}_1}(X)$ and $G_{\mathcal{S}_2}(X)$ are statistically associated, causing $p_{\mathcal{S}_1}$ and $p_{\mathcal{S}_2}$ to be statistically dependent, the occurrence of a false positive for pattern \mathcal{S}_1 , i.e. $[p_{\mathcal{S}_1} \leq \delta]$, and the occurrence of a false positive for pattern \mathcal{S}_2 , i.e. $[p_{\mathcal{S}_2} \leq \delta]$, will be statistically dependent too. The implications of this observation can be traced back to the derivations of the upper bounds on the intractable exact FWER used by the Bonferroni correction and Tarone’s improved Bonferroni correction for discrete data. Both approaches make use of the fact that, as a direct consequence of the axioms of probability, $\Pr([p_{\mathcal{S}_1} \leq \delta] \cup [p_{\mathcal{S}_2} \leq \delta]) = \Pr(p_{\mathcal{S}_1} \leq \delta) + \Pr(p_{\mathcal{S}_2} \leq \delta) - \Pr(p_{\mathcal{S}_1} \leq \delta, p_{\mathcal{S}_2} \leq \delta) \leq \Pr(p_{\mathcal{S}_1} \leq \delta) + \Pr(p_{\mathcal{S}_2} \leq \delta)$. This inequality is exact, i.e. becomes an equality, if and only if the events $[p_{\mathcal{S}_1} \leq \delta]$ and $[p_{\mathcal{S}_2} \leq \delta]$ are mutually exclusive, that is, if and only if $\Pr(p_{\mathcal{S}_1} \leq \delta, p_{\mathcal{S}_2} \leq \delta) = 0$. Under any other circumstances, $\Pr(p_{\mathcal{S}_1} \leq \delta) + \Pr(p_{\mathcal{S}_2} \leq \delta)$ will overestimate the true value of $\Pr([p_{\mathcal{S}_1} \leq \delta] \cup [p_{\mathcal{S}_2} \leq \delta])$. Most importantly, if patterns \mathcal{S}_1 and \mathcal{S}_2 are closely related, leading to a high probability that they co-occur in an input sample X , the random variables $G_{\mathcal{S}_1}(X)$ and $G_{\mathcal{S}_2}(X)$ will be strongly positively correlated¹, causing $[p_{\mathcal{S}_1} \leq \delta]$ and $[p_{\mathcal{S}_2} \leq \delta]$ to be positively correlated as well. In this particular case, which is of special relevance for significant pattern mining, $\Pr(p_{\mathcal{S}_1} \leq \delta, p_{\mathcal{S}_2} \leq \delta) \approx \Pr(p_{\mathcal{S}_1} \leq \delta) \approx \Pr(p_{\mathcal{S}_2} \leq \delta)$, implying that $\Pr([p_{\mathcal{S}_1} \leq \delta] \cup [p_{\mathcal{S}_2} \leq \delta]) \approx \Pr(p_{\mathcal{S}_1} \leq \delta) \approx \Pr(p_{\mathcal{S}_2} \leq \delta)$. The same argument can be generalised to any subset \mathcal{M}' of the search space \mathcal{M} . If the set of P-values $\{p_{\mathcal{S}} \mid \mathcal{S} \in \mathcal{M}'\}$ exhibits extensive positive correlations between subsets of P-values, possibly extending beyond mere pairwise associations between P-values, then $\Pr(\bigcup_{\mathcal{S} \in \mathcal{M}'} [p_{\mathcal{S}} \leq \delta]) \ll \sum_{\mathcal{S} \in \mathcal{M}'} \Pr(p_{\mathcal{S}} \leq \delta)$ will hold. The larger the number of patterns and the strength of the dependencies between patterns in \mathcal{M}' , the bigger the overestimation gap between $\sum_{\mathcal{S} \in \mathcal{M}'} \Pr(p_{\mathcal{S}} \leq \delta)$ and $\Pr(\bigcup_{\mathcal{S} \in \mathcal{M}'} [p_{\mathcal{S}} \leq \delta])$ will be.

The Bonferroni correction approximates the intractable exact FWER, $\text{FWER}(\delta) = \Pr(\bigcup_{\mathcal{S} \in \mathcal{M}} [p_{\mathcal{S}} \leq \delta])$, with an upper bound $\widehat{\text{FWER}}(\delta) = \sum_{\mathcal{S} \in \mathcal{M}} \Pr(p_{\mathcal{S}} \leq \delta) = \delta|\mathcal{M}|$. As discussed in-depth in Section 2.3, this upper bound tends to greatly overestimate $\text{FWER}(\delta)$ in significant pattern mining because the Bonferroni correction implicitly assumes that any pattern in the search space \mathcal{M} can cause a false positive while, in reality, only a much smaller subset $\mathcal{M}_{\text{test}}(\delta) \subseteq \mathcal{M}$ of *testable* patterns can. Tarone’s improved Bonferroni correction for discrete data hinges on that observation, proposing to

1. Two random variables $G_{\mathcal{S}_1}(X)$ and $G_{\mathcal{S}_2}(X)$ are said to be *positively correlated* if $\mathbb{E}(G_{\mathcal{S}_1}(X)G_{\mathcal{S}_2}(X)) > \mathbb{E}(G_{\mathcal{S}_1}(X))\mathbb{E}(G_{\mathcal{S}_2}(X))$. In the particular case that the random variables $G_{\mathcal{S}_1}(X)$ and $G_{\mathcal{S}_2}(X)$ are binary, this is equivalent to $\Pr(G_{\mathcal{S}_1}(X) = 1, G_{\mathcal{S}_2}(X) = 1) > \Pr(G_{\mathcal{S}_1}(X) = 1)\Pr(G_{\mathcal{S}_2}(X) = 1)$.

use instead an upper bound $\widehat{\text{FWER}}(\delta) = \sum_{\mathcal{S} \in \mathcal{M}_{\text{test}}(\delta)} \Pr(p_{\mathcal{S}} \leq \delta) = \delta |\mathcal{M}_{\text{test}}(\delta)|$. Since in practice $|\mathcal{M}_{\text{test}}(\delta)| \ll |\mathcal{M}|$, Tarone’s method drastically reduces the overestimation gap between $\widehat{\text{FWER}}(\delta)$ and $\text{FWER}(\delta)$. Nonetheless, both the Bonferroni correction and Tarone’s improved Bonferroni correction for discrete data rely on the additional simplifying assumptions that the sets of P-values $\{p_{\mathcal{S}} \mid \mathcal{S} \in \mathcal{M}\}$ and $\{p_{\mathcal{S}} \mid \mathcal{S} \in \mathcal{M}_{\text{test}}(\delta)\}$ are statistically independent, respectively. As it was shown above, due to the extensive statistical dependencies which exist between different patterns in the search space, both of these assumptions can seldom be expected to be satisfied in significant pattern mining. Firstly, this observation sheds additional evidence on the inadequacy of the Bonferroni correction for significant pattern mining. Perhaps most importantly, it also implies that while Tarone’s improved Bonferroni correction for discrete data might drastically improve statistical power compared to the Bonferroni correction, the unmet assumption of joint independence of the P-values of all testable patterns renders it over-conservative as well. Consequently, Tarone’s method still overestimates the real FWER, causing the significance threshold δ_{tar} resulting from applying this approach to be still considerably smaller than the significance threshold δ^* one would obtain if the real value of $\text{FWER}(\delta)$ could be computed exactly.

In the next sections we will introduce the Westfall-Young permutation testing procedure, which allows to bypass the need to make independence assumptions by directly obtaining an empirical estimate of $\text{FWER}(\delta)$, and describe how this approach can be leveraged to improve statistical power in significant pattern mining.

4.2 EMPIRICALLY APPROXIMATING THE FWER VIA RANDOM PERMUTATIONS

Both the Bonferroni correction and Tarone’s method bypass the difficulty to exactly evaluate the FWER at a given significance threshold δ by using an easier-to-compute upper bound $\widehat{\text{FWER}}(\delta)$ of the FWER as a surrogate of the intractable quantity $\text{FWER}(\delta)$. As argued in the previous section, while this paradigm enjoys the benefits of simplicity and computational efficiency, it necessarily leads to over-conservative estimations of the FWER. An alternative approach is to instead try to obtain an empirical estimate $\widehat{\text{FWER}}(\delta)$ of the exact value of $\text{FWER}(\delta)$ by using resampling techniques.

One of the most commonly used resampling schemes towards this end consists of applying random permutations to the class labels [74]. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be an input dataset with n observations $x \in \mathcal{X}$ and class labels $y \in \{0, 1\}$. Suppose that $\pi : \llbracket 1, n \rrbracket \rightarrow \llbracket 1, n \rrbracket$ is a random permutation, i.e. a permutation of the set $\llbracket 1, n \rrbracket$ selected uniformly at random from the set of all $n!$ permutations of $\llbracket 1, n \rrbracket$. Define the resampled dataset $\tilde{\mathcal{D}} = \left\{ \left(x_i, y_{\pi(i)} \right) \right\}_{i=1}^n$ in such a way that the i -th observation x_i is paired with the class label of observation $\pi(i)$, for each $i = 1, \dots, n$. The effect of obtaining the class labels in the resampled dataset $\tilde{\mathcal{D}}$ by randomly permuting the original labels in \mathcal{D} is to assign a random class label to each observation while keeping the class ratio n_1/n unchanged. As a consequence, any statistical dependency between patterns and labels which might have existed in the original dataset \mathcal{D} is effectively eliminated by the permutation process. While the ground-truth regarding which patterns $\mathcal{S} \in \mathcal{M}$ are statistically associated with the class labels in the original dataset \mathcal{D} is unknown, in the resampled dataset $\tilde{\mathcal{D}}$, no pattern $\mathcal{S} \in \mathcal{M}$ can possibly be

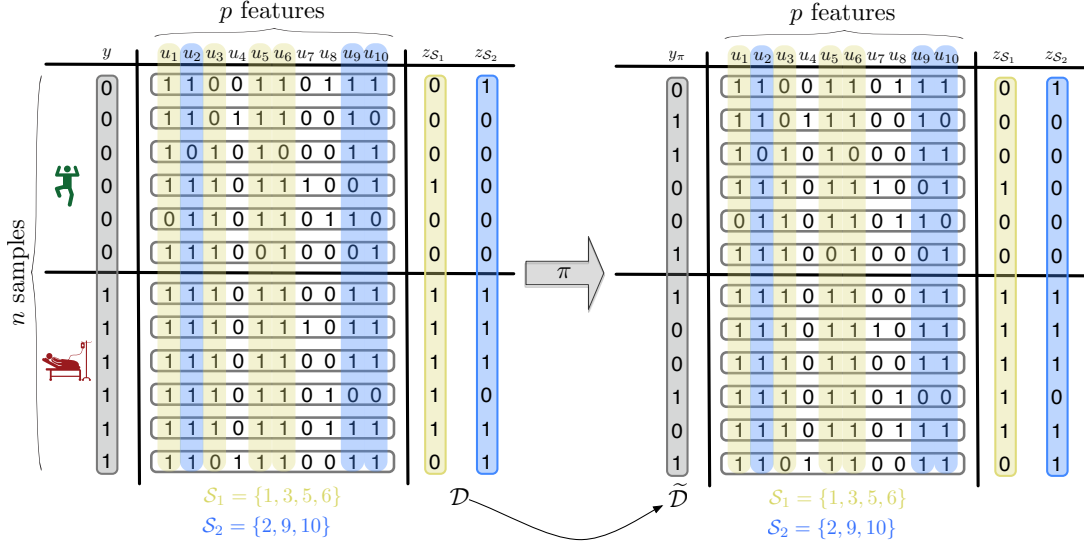


Figure 4.3. – In this significant itemset mining example, an input dataset \mathcal{D} with $p = 10$ features and $n = 12$ samples, shown on the left, has been resampled by applying a random permutation $\pi : \llbracket 1, n \rrbracket \rightarrow \llbracket 1, n \rrbracket$ to the class labels, resulting in a new dataset $\tilde{\mathcal{D}}$, shown on the right. While both datasets share the same observations $\{x_i\}_{i=1}^n$ and class ratio n_1/n , the mapping of class labels and observations is different. A consequence of this is that the occurrence of patterns \mathcal{S}_1 and \mathcal{S}_2 , which is enriched in observations belonging to class $y = 1$ in the original dataset \mathcal{D} , no longer show any association with the class labels in the resampled dataset $\tilde{\mathcal{D}}$.

associated with the class labels. In other words, the *global null hypothesis* $\mathcal{M}_0 = \mathcal{M}$ holds for $\tilde{\mathcal{D}}$. Figure 4.3 illustrates the permutation process on a toy significant itemset mining dataset with $n = 12$ samples and $p = 10$ features.

The fact that a resampled dataset $\tilde{\mathcal{D}}$ obtained in this manner is known to contain no associations can be exploited to obtain an empirical estimate of the FWER under the global null hypothesis that no pattern $\mathcal{S} \in \mathcal{M}$ is associated with the class labels. Suppose that the resampling process described above is repeated a number j_p of times, leading to a set $\{\tilde{\mathcal{D}}^{(k)}\}_{k=1}^{j_p}$ of j_p resampled datasets. Leaving the matter of computational feasibility temporarily aside, suppose that for each dataset $\tilde{\mathcal{D}}^{(k)}$, the P-values $p_{\mathcal{S}}^{(k)}$ for all patterns $\mathcal{S} \in \mathcal{M}$ had been obtained. Define $p_{\text{ms}}^{(k)}$ to be the P-value corresponding to the most significant pattern, i.e. $p_{\text{ms}}^{(k)} = \min \{p_{\mathcal{S}}^{(k)} \mid \mathcal{S} \in \mathcal{M}\}$. By construction, $p_{\text{ms}}^{(k)} \leq p_{\mathcal{S}}^{(k)}$ for all patterns $\mathcal{S} \in \mathcal{M}$. Hence, if $p_{\text{ms}}^{(k)} > \delta$, it follows that no pattern $\mathcal{S} \in \mathcal{M}$ will be deemed significant in the k -th resampled dataset $\tilde{\mathcal{D}}^{(k)}$, leading to no false positives being reported in this resampled dataset, i.e. $\text{FP}^{(k)}(\delta) = 0$. On the contrary, if $p_{\text{ms}}^{(k)} \leq \delta$, there is at least one pattern $\mathcal{S} \in \mathcal{M}$ being deemed significant for

the k -th resampled dataset. Since $\tilde{\mathcal{D}}^{(k)}$ is known to contain no associations, this implies that $\text{FP}^{(k)}(\delta) > 0$. Therefore, an empirical estimator of the FWER can be obtained as:

$$\widehat{\text{FWER}}(\delta) = \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[p_{\text{ms}}^{(k)} \leq \delta \right], \quad (4.1)$$

where $\mathbb{1}[\bullet]$ evaluates to 1 if its input argument is true and to 0 otherwise. Intuitively, the estimator $\widehat{\text{FWER}}(\delta)$ of the FWER at significance threshold δ is simply given by the proportion of the j_p resampled datasets that contain at least one false positive. If the number of permutations j_p is chosen sufficiently large (e.g. $j_p \approx 10,000$), $\widehat{\text{FWER}}(\delta)$ will be a rather accurate estimate of the true value of $\text{FWER}(\delta)$. A corrected significance threshold can then be proposed based on this estimator as:

$$\delta_{\text{wy}} = \max \left\{ \delta \mid \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[p_{\text{ms}}^{(k)} \leq \delta \right] \leq \alpha \right\}. \quad (4.2)$$

In other words, the corrected significance threshold δ_{wy} can be obtained as the α -quantile of the set $\left\{ p_{\text{ms}}^{(k)} \right\}_{k=1}^{j_p}$. It can be shown that δ_{wy} defined in this way guarantees *weak control* of the FWER, i.e. controls the FWER under the global null hypothesis. Moreover, if the *subset pivotality condition* [74] holds, then it can be shown that permutation testing also strongly controls the FWER, i.e. controls the FWER when any subset of hypotheses are allowed to be non-null. Permutation testing has been nonetheless extensively applied to problems for which the subset pivotality condition cannot be proven to hold, as is the case of significant pattern mining, while still leading to successful results.

The empirical estimate of the FWER obtained via permutation testing implicitly accounts for the dependence structure that might exist between patterns in the search space \mathcal{M} . While this can lead to a considerable gain in statistical power with respect to Tarone’s method, naively applying this procedure to significant pattern mining is entirely unfeasible. Firstly, evaluating $p_{\text{ms}}^{(k)}$ for a single resampled dataset $\tilde{\mathcal{D}}^{(k)}$ is a challenging problem on its own. In principle, a naive evaluation of $p_{\text{ms}}^{(k)}$ would require computing the P-value $p_{\mathcal{S}}^{(k)}$ of each pattern $\mathcal{S} \in \mathcal{M}$, which is typically unfeasible due to the size of the search space \mathcal{M} . Moreover, in permutation testing, this operation needs to be repeated between $j_p = 1,000$ and $j_p = 10,000$ times if a sufficiently accurate estimate of the FWER is to be obtained, further exacerbating the computational challenge.

While making use of Westfall-Young permutation testing in significant pattern mining might therefore seem to be a hopeless endeavour, the next section will describe in detail how the concept of the minimum attainable P-value, introduced in Section 2.3 in the context of Tarone’s method, can also be leveraged to design computationally efficient permutation testing-based significant pattern mining algorithms.

4.3 PERMUTATION TESTING IN SIGNIFICANT PATTERN MINING

This section builds upon the framework presented in Chapter 3, describing two novel significant pattern mining algorithms which use the corrected significance

threshold δ_{wy} obtained via permutation testing instead of Tarone’s corrected significance threshold δ_{tar} , thereby implicitly exploiting the dependence between patterns in the search space to improve statistical power. Both novel approaches follow the skeleton of Algorithm 3.1, thus proceeding in two steps. First, a specialised algorithm is designed to efficiently compute the corrected significance threshold δ_{wy} . Then, Algorithm 3.3 can be invoked to retrieve all patterns in the search space which are significantly associated with the class labels at level δ_{wy} . Hence, the transition from using Tarone’s method to permutation testing merely involves substituting Algorithm 3.2 by a different approach able to compute δ_{wy} efficiently. The rest of this section will be devoted to describe two such approaches.

The first method to be introduced is the FastWY algorithm [43], the only pre-existing attempt to use Westfall-Young permutation testing in significant pattern mining. Next, our contribution, the Westfall-Young light algorithm, will be discussed in detail. Both algorithms are identical from a statistical perspective: they both exactly compute the corrected significance threshold δ_{wy} and retrieve all patterns significantly associated with the class labels at that level. Consequently, FastWY and Westfall-Young light are indistinguishable in terms of statistical power and false positive rate. Nonetheless, they radically differ from an algorithmic perspective, showing vast differences in computational efficiency when applied to real-world data.

4.3.1 Related work: the FastWY algorithm

The idea of using permutation testing to improve statistical power in significant pattern mining was pioneered by [43]. In their work, the authors explicitly tackle the problem of efficiently computing $p_{ms}^{(k)}$ for a single resampled dataset $\tilde{D}^{(k)}$. Once $p_{ms}^{(k)}$ has been obtained for each $k = 1, \dots, j_p$, the corrected significance threshold δ_{wy} can be evaluated according to Equation (4.2). In order to avoid computing the P-values $p_S^{(k)}$ for all patterns $S \in \mathcal{M}$ in the search space, FastWY relies on the concept of testability. Suppose that $p_{ms}^{(k)}(\delta) = \min \{ p_S^{(k)} \mid S \in \mathcal{M}_{test}(\delta) \}$ was available for some $\delta \in (0, 1)$. Then, since $p_S^{(k)} \geq p_{S, \min} > \delta$ for all patterns $S \in \mathcal{M} \setminus \mathcal{M}_{test}(\delta)$, it follows that $p_{ms}^{(k)}(\delta) \leq \delta$ implies $p_{ms}^{(k)} = p_{ms}^{(k)}(\delta)$. In other words, if the P-value $p_{ms}^{(k)}(\delta)$ corresponding to the most significant testable pattern at level δ is known and happens to be smaller or equal than δ , then none of the patterns which are untestable at level δ could possibly be more significant. In that case, it would therefore be possible to compute $p_{ms}^{(k)}$ without the need to evaluate the P-values $p_S^{(k)}$ for any pattern S which is untestable at level δ . The FastWY algorithm exploits this observation, as described in Algorithm 4.1.

In order to compute δ_{wy} , FastWY evaluates $\{ p_{ms}^{(k)} \}_{k=1}^{j_p}$ by processing each resampled dataset $\tilde{D}^{(k)}$ independently, as shown in Lines 2-4. For a given resampled dataset $\tilde{D}^{(k)}$, the routine `compute_pmin`, invoked in Line 4, obtains $p_{ms}^{(k)}$ by iteratively computing $p_{ms}^{(k)}(\delta)$ for a sequence of monotonically increasing values of δ . This sequence is initialised at the smallest attainable P-value, $\delta_0 = \min \{ p_{\min}(r_S) \mid r_S \in \llbracket 0, n \rrbracket \}$, where $p_{\min}(r_S)$ is the minimum attainable P-value function of the test statistic of choice.

Algorithm 4.1 FastWY**Input:** Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, target FWER α , number of permutations j_p **Output:** Corrected significance threshold δ_{wy}

```

1: function compute_significance_threshold( $\mathcal{D}, \alpha, j_p$ )
2:   for  $k = 1, 2, \dots, j_p$  do
3:     Obtain a random permutation  $\pi^{(k)} : \llbracket 1, n \rrbracket \rightarrow \llbracket 1, n \rrbracket$ 
4:      $p_{\text{ms}}^{(k)} \leftarrow \text{compute\_pmin}(\mathcal{D}, \pi^{(k)})$ 
5:      $\delta_{\text{wy}} \leftarrow \max \left\{ \delta \mid \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[ p_{\text{ms}}^{(k)} \leq \delta \right] \leq \alpha \right\}$ 
6:     Return  $\delta_{\text{wy}}$ 
7: end function
8: function compute_pmin( $\mathcal{D}, \pi$ )
9:   Initialise global variable  $\delta \leftarrow \min \{ p_{\text{min}}(r_S) \mid r_S \in \llbracket 0, n \rrbracket \}$ 
10:   $p_{\text{ms}}(\delta) \leftarrow \text{compute\_pmin\_delta}(\mathcal{D}, \pi, \delta)$ 
11:  while  $p_{\text{ms}}(\delta) > \delta$  do
12:    Increase  $\delta$ 
13:     $p_{\text{ms}}(\delta) \leftarrow \text{compute\_pmin\_delta}(\mathcal{D}, \pi, \delta)$ 
14:   $p_{\text{ms}} \leftarrow p_{\text{ms}}(\delta)$ 
15:  Return  $p_{\text{ms}}$ 
16: end function
17: function compute_pmin_delta( $\mathcal{D}, \pi, \delta$ )
18:   Initialise global variable  $\hat{p}_{\text{ms}}(\delta) \leftarrow 1$ 
19:   NEXT( $\emptyset$ ) ▷ Start pattern enumeration
20:    $p_{\text{ms}}(\delta) \leftarrow \hat{p}_{\text{ms}}(\delta)$ 
21:   Return  $p_{\text{ms}}(\delta)$ 
22: end function
23: procedure NEXT( $\mathcal{S}$ )
24:   Compute the minimum attainable P-value  $p_{\mathcal{S}, \text{min}}$  ▷ see Section 3.3
25:   if  $p_{\mathcal{S}, \text{min}} \leq \delta$  then ▷ if pattern  $\mathcal{S}$  is testable at level  $\delta$  then
26:     Compute P-value  $p_{\mathcal{S}}$  for resampled dataset  $\tilde{D}$ 
27:      $\hat{p}_{\text{ms}}(\delta) \leftarrow \min(\hat{p}_{\text{ms}}(\delta), p_{\mathcal{S}})$ 
28:   if not pruning_condition( $\mathcal{S}, \delta$ ) then ▷ see Section 3.4
29:     for  $\mathcal{S}' \in \text{Children}(\mathcal{S})$  do
30:       NEXT( $\mathcal{S}'$ ) ▷ Recursively visit nodes in the tree depth-first
31: end procedure

```

This initialisation is justified by the observation that $\mathcal{M}_{\text{test}}(\delta) = \emptyset$ for any $\delta < \delta_0$. As long as $p_{\text{ms}}^{(k)}(\delta) > \delta$, the algorithm continues to increase δ and recompute $p_{\text{ms}}^{(k)}(\delta)$, as shown in Lines 11-13, until the condition $p_{\text{ms}}^{(k)}(\delta) \leq \delta$ is eventually satisfied. At that point, as discussed previously, $p_{\text{ms}}^{(k)} = p_{\text{ms}}^{(k)}(\delta)$ holds and the routine `compute_pmin` can terminate, returning the exact value of $p_{\text{ms}}^{(k)}$. Each computation of $p_{\text{ms}}^{(k)}(\delta)$ required by that process is performed using the routine `compute_pmin_delta` (Lines 17-31). Each time this routine is executed, the search space \mathcal{M} is explored by traversing the pattern enumeration tree depth-first, making use of the search space pruning criterion

introduced in Section 3.4 to efficiently retrieve the set $\mathcal{M}_{\text{test}}(\delta)$ of patterns testable at level δ . Each time a testable pattern is enumerated, the routine computes its P-value $p_S^{(k)}$ in order to iteratively update $p_{\text{ms}}^{(k)}(\delta)$. Once all testable patterns have been visited, the routine terminates, returning the exact value of $p_{\text{ms}}^{(k)}(\delta)$.

As a consequence, the FastWY algorithm requires enumerating patterns from scratch each time the routine `compute_pmin_delta` is invoked to compute $p_{\text{ms}}^{(k)}(\delta)$ for a certain value of δ . This might occur a potentially large number of times for any given resampled dataset $\tilde{D}^{(k)}$ but, perhaps most importantly, this entire process must be repeated for each of the $j_p \approx 10,000$ resampled datasets. As a result, even if FastWY is able to obtain the exact value of $p_{\text{ms}}^{(k)}$ while enumerating only a small subset of patterns in the search space \mathcal{M} , the computational overhead of repeating pattern enumeration such a large number of times limits its applicability to datasets of only small-to-moderate size.

4.3.2 Contribution: the Westfall-Young light algorithm

Our contribution, the Westfall-Young light algorithm, builds upon the work in [43] to provide an alternative approach to apply permutation testing in significant pattern mining. Unlike FastWY, our method processes all resampled datasets simultaneously, requiring to enumerate patterns only a single time. In practice, this leads to a drastic reduction in runtime and memory usage that allows scaling-up permutation testing-based significant pattern mining to considerably larger datasets.

Pseudocode describing Westfall-Young light is shown in Algorithm 4.2. The skeleton of the method closely parallels Algorithm 3.2. The search space \mathcal{M} of all candidate patterns is explored in the same manner: recursively traversing a pattern enumeration tree that satisfies $\mathcal{S}' \in \text{Children}(\mathcal{S}) \Rightarrow \mathcal{S} \subseteq \mathcal{S}'$ depth-first. The algorithm begins by initialising the estimate $\hat{\delta}_{\text{wy}}$ of the corrected significance threshold to 1 (Line 2). Next, in Lines 3-5, for each of the j_p resampled datasets, the algorithm precomputes the random permutation of the class labels and initialises the estimate $\tilde{p}_{\text{ms}}^{(k)}$ of the most significant P-value to 1. After the initialisation phase, the algorithm proceeds to start the pattern enumeration procedure at the root of the tree (Line 6). For each pattern \mathcal{S} visited during the traversal of the enumeration tree, Algorithm 4.2 first computes the minimum attainable P-value $p_{\mathcal{S},\text{min}}$ in Line 11. The algorithm then proceeds differently depending on the testability of pattern \mathcal{S} .

If pattern \mathcal{S} is testable at level $\hat{\delta}_{\text{wy}}$ then, for each of the j_p resampled datasets $\tilde{D}^{(k)}$, the algorithm computes the P-value $p_S^{(k)}$ and updates the estimate $\tilde{p}_{\text{ms}}^{(k)}$ of the most significant P-value for the k -th resampled dataset (Lines 13-15). Next, in Line 16, an estimate $\widetilde{\text{FWER}}(\hat{\delta}_{\text{wy}})$ of the FWER at level $\hat{\delta}_{\text{wy}}$ is obtained using the estimates $\tilde{p}_{\text{ms}}^{(k)}$ of the most significant P-value for each resampled dataset. Since the enumeration process is not yet completed, $\tilde{p}_{\text{ms}}^{(k)} \geq p_{\text{ms}}^{(k)}$ holds. This leads to $\frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[\tilde{p}_{\text{ms}}^{(k)} \leq \hat{\delta}_{\text{wy}} \right] = \widetilde{\text{FWER}}(\hat{\delta}_{\text{wy}})$ being a lower bound of $\widehat{\text{FWER}}(\hat{\delta}_{\text{wy}}) = \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[p_{\text{ms}}^{(k)} \leq \hat{\delta}_{\text{wy}} \right]$. Thus, if $\widetilde{\text{FWER}}(\hat{\delta}_{\text{wy}}) > \alpha$, it follows that $\widehat{\text{FWER}}(\hat{\delta}_{\text{wy}}) > \alpha$ as well, implying that the FWER condition is not satisfied. In Lines 17-19, Algorithm 4.2 checks this condition and, if

Algorithm 4.2 Westfall-Young light**Input:** Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, target FWER α , number of permutations j_p **Output:** Corrected significance threshold δ_{wy}

```

1: function compute_significance_threshold( $\mathcal{D}, \alpha, j_p$ )
2:   Initialise global variable  $\widehat{\delta}_{\text{wy}} \leftarrow 1$ 
3:   for  $k = 1, 2, \dots, j_p$  do
4:     Obtain a random permutation  $\pi^{(k)} : \llbracket 1, n \rrbracket \rightarrow \llbracket 1, n \rrbracket$ 
5:     Initialise global variable  $\widetilde{p}_{\text{ms}}^{(k)} \leftarrow 1$ 
6:     NEXT( $\emptyset$ ) ▷ Start pattern enumeration
7:      $\delta_{\text{wy}} \leftarrow \max \left\{ \delta \in [0, \widehat{\delta}_{\text{wy}}] \mid \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[ \widetilde{p}_{\text{ms}}^{(k)} \leq \delta \right] \leq \alpha \right\}$ 
8:     Return  $\delta_{\text{wy}}$ 
9:   end function
10: procedure NEXT( $\mathcal{S}$ )
11:   Compute the minimum attainable P-value  $p_{\mathcal{S}, \text{min}}$  ▷ see Section 3.3
12:   if  $p_{\mathcal{S}, \text{min}} \leq \widehat{\delta}_{\text{wy}}$  then ▷ if pattern  $\mathcal{S}$  is testable at level  $\widehat{\delta}_{\text{wy}}$  then
13:     for  $k = 1, 2, \dots, j_p$  do
14:       Compute P-value  $p_{\mathcal{S}}^{(k)}$  for resampled dataset  $\widetilde{\mathcal{D}}^{(k)}$ 
15:        $\widetilde{p}_{\text{ms}}^{(k)} \leftarrow \min(\widetilde{p}_{\text{ms}}^{(k)}, p_{\mathcal{S}}^{(k)})$ 
16:        $\widetilde{\text{FWER}}(\widehat{\delta}_{\text{wy}}) \leftarrow \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[ \widetilde{p}_{\text{ms}}^{(k)} \leq \widehat{\delta}_{\text{wy}} \right]$ 
17:       while  $\widetilde{\text{FWER}}(\widehat{\delta}_{\text{wy}}) > \alpha$  do
18:         Decrease  $\widehat{\delta}_{\text{wy}}$ 
19:          $\widetilde{\text{FWER}}(\widehat{\delta}_{\text{wy}}) \leftarrow \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[ \widetilde{p}_{\text{ms}}^{(k)} \leq \widehat{\delta}_{\text{wy}} \right]$ 
20:       if not pruning_condition( $\mathcal{S}, \widehat{\delta}_{\text{wy}}$ ) then ▷ see Section 3.4
21:         for  $\mathcal{S}' \in \text{Children}(\mathcal{S})$  do
22:           NEXT( $\mathcal{S}'$ ) ▷ Recursively visit nodes in the tree depth-first
23:     end procedure

```

found to be violated, decreases the estimate $\widehat{\delta}_{\text{wy}}$ of the corrected significance threshold until the FWER condition is satisfied again.

On the contrary, if pattern \mathcal{S} is untestable at level $\widehat{\delta}_{\text{wy}}$, it will not affect the values of the FWER estimator $\widetilde{\text{FWER}}(\delta) = \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[\widetilde{p}_{\text{ms}}^{(k)} \leq \delta \right]$ for any $\delta \leq \widehat{\delta}_{\text{wy}}$. Consequently, the computation of the P-values $p_{\mathcal{S}}^{(k)}$ for $k = 1, \dots, j_p$ can be skipped, as well as the update of the FWER estimator $\widetilde{\text{FWER}}(\widehat{\delta}_{\text{wy}})$. The fact that untestable patterns cannot modify the value of $\widetilde{\text{FWER}}(\delta)$ follows from the definition of testability. If $p_{\mathcal{S}, \text{min}} > \widehat{\delta}_{\text{wy}}$ and $\widetilde{p}_{\text{ms}}^{(k)} > \widehat{\delta}_{\text{wy}}$, then $\min(\widetilde{p}_{\text{ms}}^{(k)}, p_{\mathcal{S}}^{(k)}) > \widehat{\delta}_{\text{wy}}$. Moreover, since $\widehat{\delta}_{\text{wy}} \geq \delta_{\text{wy}}$ at any point during the execution of the algorithm, it follows that if a pattern \mathcal{S} is untestable at level $\widehat{\delta}_{\text{wy}}$, then $\min(\widetilde{p}_{\text{ms}}^{(k)}, p_{\mathcal{S}}^{(k)}) > \delta_{\text{wy}}$, leading to pattern \mathcal{S} being irrelevant as far as the permutation testing-based FWER estimator is concerned.

Search space pruning is fundamental to the computational feasibility of Algorithm 4.2, as is for other significant pattern mining algorithms. As shown in Section 3.4, when Pearson's χ^2 test or Fisher's exact test are used, descendants \mathcal{S}' of an untestable

pattern \mathcal{S} will also be untestable provided that $r_{\mathcal{S}} \leq \min(n_1, n - n_1)$. Reiterating the argument above, this implies that those descendants cannot affect the value of $\widehat{\text{FWER}}(\delta)$ for any $\delta \leq \widehat{\delta}_{\text{wy}}$ and, thus, can be pruned from the search space. In other words, the pruning condition of Algorithm 4.2 is identical to that of Algorithm 3.2. Finally, Lines 21-22 continue the traversal of the tree recursively, visiting the children of patterns for which the pruning condition does not apply.

As was the case for Algorithm 3.2, $\widehat{\delta}_{\text{wy}}$ progressively decreases as patterns are enumerated, leading to less patterns being testable and the pruning condition becoming more stringent. Eventually, the algorithm ends its execution after all patterns have either been pruned or visited. A fundamental property of the Westfall-Young light algorithm is that, at convergence, the estimate $\widetilde{p}_{\text{ms}}^{(k)}$ of the most significant P-value for the k -th resampled dataset is, in general, not equal to the exact value of $p_{\text{ms}}^{(k)}$. Instead, when Algorithm 4.2 terminates, $\widetilde{p}_{\text{ms}}^{(k)} = \min(p_{\text{ms}}^{(k)}(\widehat{\delta}_{\text{wy}}), q_{\text{ms}}^{(k)})$, where:

- (i) $p_{\text{ms}}^{(k)}(\widehat{\delta}_{\text{wy}}) = \min \left\{ p_{\mathcal{S}}^{(k)} \mid \mathcal{S} \in \mathcal{M}_{\text{test}}(\widehat{\delta}_{\text{wy}}) \right\}$.
- (ii) $q_{\text{ms}}^{(k)} = \min \left\{ p_{\mathcal{S}}^{(k)} \mid \mathcal{Q}(\widehat{\delta}_{\text{wy}}) \right\}$, with $\mathcal{Q}(\widehat{\delta}_{\text{wy}}) \subseteq \mathcal{M} \setminus \mathcal{M}_{\text{test}}(\widehat{\delta}_{\text{wy}})$ being the subset of untestable patterns at level $\widehat{\delta}_{\text{wy}}$ which the algorithm visited during earlier iterations.

In particular, this implies that $\widetilde{p}_{\text{ms}}^{(k)} \leq p_{\text{ms}}^{(k)}(\widehat{\delta}_{\text{wy}})$ and, consequently, $\mathbb{1} \left[\widetilde{p}_{\text{ms}}^{(k)} \leq \delta \right] = \mathbb{1} \left[p_{\text{ms}}^{(k)} \leq \delta \right]$ holds for all $\delta \leq \widehat{\delta}_{\text{wy}}$. This in turn leads to the most remarkable property of the Westfall-Young light algorithm: it is able to exactly evaluate $\widehat{\text{FWER}}(\delta) = \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[p_{\text{ms}}^{(k)} \leq \delta \right]$ as $\widehat{\text{FWER}}(\delta) = \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[\widetilde{p}_{\text{ms}}^{(k)} \leq \delta \right]$ in the range $\delta \in [0, \widehat{\delta}_{\text{wy}}]$ despite not having computed $\left\{ p_{\text{ms}}^{(k)} \right\}_{k=1}^{j_p}$ exactly. Since $\widehat{\delta}_{\text{wy}} \geq \delta_{\text{wy}}$ always holds throughout the execution of the algorithm, including after its termination, it immediately follows that the Westfall-Young light algorithm is able to exactly evaluate $\widehat{\text{FWER}}(\delta)$ within the range of values of δ which are necessary to retrieve δ_{wy} , i.e.

$$\delta_{\text{wy}} = \max \left\{ \delta \in [0, \widehat{\delta}_{\text{wy}}] \mid \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[\widetilde{p}_{\text{ms}}^{(k)} \leq \delta \right] \leq \alpha \right\} \quad (4.3)$$

holds. This result is used in Line 7 of the algorithm to compute δ_{wy} and finally return it in the next line. Once the corrected significance threshold δ_{wy} has been obtained, Algorithm 3.3 can be used to retrieve all patterns \mathcal{S} that are statistically significant at level δ_{wy} .

Implementation considerations

All low-level design choices which were discussed in Section 3.5 in the context of Algorithm 3.2 also apply to Westfall-Young light:

- (i) In order to efficiently navigate the pattern enumeration tree, highly-optimised, problem-specific pattern mining algorithms can be seamlessly integrated as a subcomponent of Westfall-Young light.

- (ii) When decreasing the estimate $\widehat{\delta}_{\text{wy}}$ of the corrected significance threshold in Line 18 of Algorithm 4.2, the $\lfloor \frac{n}{2} \rfloor$ distinct values the minimum attainable P-value function $p_{\min}(r_S)$ takes can be used as the sole candidate values for $\widehat{\delta}_{\text{wy}}$.
- (iii) To compute the corrected significance threshold δ_{wy} , it is unnecessary to keep the set $\mathcal{M}_{\text{test}}(\widehat{\delta}_{\text{wy}})$ of testable patterns in memory. Significant patterns at level δ_{wy} can be retrieved by performing pattern enumeration a second time using Algorithm 3.3.
- (iv) The minimum attainable P-value function $p_{\min}(r_S)$ of the test statistic of choice can be cached in a look-up table to speed-up the execution of the algorithm at the price of a negligible increase in memory usage.

The fact that Westfall-Young light processes simultaneously all j_p resampled datasets allows to introduce an additional low-level optimisation. Given a pattern S with r_S occurrences in the input dataset \mathcal{D} , the number of distinct P-values one can obtain is of the order $O(\min(n_1, r_S))$. For many patterns in the search space \mathcal{M} , $\min(n_1, r_S) \ll j_p$ holds. In these cases, it is advantageous to implement Line 14 in Algorithm 4.2 by pre-computing all attainable P-values and storing them in a look-up table. This would reduce the computational complexity of computing $p_S^{(k)}$ for all j_p resampled datasets from $O(j_p)$ to $O(\min(n_1, r_S))$.

Comparison to FastWY

As previously mentioned, FastWY and Westfall-Young light can be understood as two different approaches to compute δ_{wy} , being thus statistically identical. However, Westfall-Young light introduces many novel elements that render it considerably more computationally efficient. In particular:

- (i) FastWY uses the search strategy postulated by the original implementation of the LAMP algorithm [26]: δ is initialised to the smallest value it could take, being then increased in each iteration of the algorithm until convergence. Nonetheless, subsequent work [65, 66] has shown that the opposite search strategy, namely, initialising δ to 1 and proceeding instead by decreasing it in each iteration, reduces the runtime of the algorithm by several orders of magnitude. Westfall-Young light uses this vastly more efficient search strategy, as does the LAMP 2.0 algorithm [65] described in Chapter 3.
- (ii) In order to compute the corrected significance threshold δ_{wy} , the Westfall-Young light algorithm needs to traverse the pattern enumeration tree only once. In contrast, FastWY needs to repeat this process j_p times. This leads to either a vast increase in runtime, if the enumeration is repeated naively, or a vast increase in memory usage, if all intermediate computations are cached in memory to reduce the runtime overhead.
- (iii) FastWY requires computing the most significant P-values $\left\{ p_{\text{ms}}^{(k)} \right\}_{k=1}^{j_p}$ of all j_p resampled datasets. In particular, resampled datasets for which $p_{\text{ms}}^{(k)}$ ends up being large require exploring a larger proportion of the search space \mathcal{M} , dominating the overall runtime of the algorithm. On the contrary, the Westfall-Young light

algorithm does not need to compute $\left\{p_{\text{ms}}^{(k)}\right\}_{k=1}^{j_p}$ exactly, effectively eliminating this undesirable effect entirely.

As a result of these improvements, *Westfall-Young light* drastically outperforms *FastWY* in terms of runtime and memory usage, as will be shown in Section 4.4.

Comparison to LAMP 2.0

Compared to the *LAMP 2.0* algorithm described in Algorithm 3.2, *Westfall-Young light* will exhibit more statistical power due to the use of permutation testing to obtain a better approximation of the FWER. However, it is a more computationally demanding approach, as j_p P-values need to be computed for each pattern deemed testable during the pattern enumeration procedure. In short, *Westfall-Young light* allows to trade off computational complexity for statistical power, an option that might be desirable in applications where signals are too weak to be detected by *LAMP 2.0*.

4.4 EXPERIMENTS

In this section we compare our contribution, the *Westfall-Young light* algorithm, to the current state of the art for permutation testing-based significant pattern mining, the *FastWY* algorithm. In order to make a comprehensive evaluation of both approaches, a wide range of real-world datasets will be used, including both significant itemset mining and significant subgraph mining experiments.

4.4.1 *Experimental Setup*

Our implementation of *Westfall-Young light* was written in C/C++. While the original implementation of *FastWY* was written in Python by its authors, in this section we compare to our own version of *FastWY*, written in C/C++ to allow for a fair comparison. This new implementation of *FastWY* used as a baseline is about two to three orders of magnitude faster and reduces the amount of memory used by one to two orders of magnitude compared to the original Python version. In order to alleviate the impact of having to repeat pattern enumeration j_p times, the original implementation of *FastWY* stores intermediate computations in memory, sacrificing memory usage to be able to analyse datasets of small-to-moderate size in a feasible amount of time. For the sake of consistency, our C/C++ implementation of *FastWY* resorted to the same strategy.

The significant itemset mining instances of both *Westfall-Young light* and *FastWY* use *LCM* version 3 [69] as the underlying itemset mining algorithm to traverse the enumeration tree. *LCM* is widely considered one of the most efficient itemset mining algorithms, having won the *FIMI'04* frequent itemset mining competition [75]. The code was compiled using Intel C++ compiler version 14.0.1 with `-O3` optimisation and executed on a single thread of a 2.7 GHz Intel Xeon CPU with 256 GB of memory available. Similarly, the significant subgraph mining instances of *Westfall-Young light* and *FastWY* make use of *Gaston* [72], reported to be one of the fastest subgraph

mining algorithms [76]. The code was compiled using gcc 4.8.2 with -O3 optimisation and run on a single thread of a 2.5 GHz Intel Xeon CPU with 256 GB of memory.

Significant Itemset Mining Datasets

Table 4.1. – Characteristics of the significant itemset mining datasets. n and n_1 are the total number of samples and the number of samples in the positive class, respectively; p is the number of features (items) and \bar{p}_{act} is the average number of active features per sample. The ratio n/n_1 is shown only for labeled datasets.

Dataset	n	n/n_1	p	\bar{p}_{act}
TicTacToe	958	2.89	18	6.93
Chess	3,196	–	75	37.00
Inetads	3,279	7.14	1,554	12.00
Mushroom	8,124	2.08	117	22.00
Breast cancer	12,773	11.31	1,129	6.70
Pumsb-star	49,046	–	7,117	50.48
Connect	67,557	–	129	43.00
BmsWebview	77,512	–	3,340	4.62
Retail	88,162	–	16,470	10.31
T10I4D100K	100,000	–	870	10.10
T40I10D100K	100,000	–	942	39.61
Bmspos	515,597	–	1,657	6.53

Our significant itemset mining experiments include four labeled datasets: TicTacToe², Inetads³, Mushroom⁴, and Breast cancer. The first three datasets are widely studied datasets from the UCI repository [77] whereas the Breast cancer dataset is described in [43]. Additionally, our experiments were extended by including eight unlabelled datasets commonly used to benchmark frequent itemset mining algorithms [75]: Bmspos, BmsWebview, Retail, T10I4D100K, T40I10D100K, Chess, Connect and Pumsb-star. In order to use these unlabelled datasets in our experiments, we exploit the observation that Algorithms 4.1 and 4.2 only depend on the class labels via n_1 , the number of samples in the positive class. Two representative scenarios were considered for each unlabelled dataset: (i) a case with balanced classes, $n/n_1 = 2$ and (ii) a case with highly unbalanced classes, $n/n_1 = 10$. The main properties of each dataset are summarised in Table 4.1.

Significant Subgraph Mining Datasets

Our significant subgraph mining experiments include 12 labelled graph datasets: four PTC (Predictive Toxicology Challenge) datasets⁵, four NCI (National Cancer

2. <https://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame>

3. <https://archive.ics.uci.edu/ml/datasets/Internet+Advertisements>

4. <https://archive.ics.uci.edu/ml/datasets/mushroom>

5. <http://www.predictive-toxicology.org/ptc/>

Table 4.2. – Characteristics of the significant subgraph mining datasets, where $|V|$ and $|E|$ denote the number of vertices and edges, respectively.

Dataset	n	n/n_1	$\max V $	$\max E $
PTC (MR)	584	3.23	181	181
PTC (FR)	583	3.74	181	181
PTC (MM)	576	3.18	181	181
PTC (FM)	563	3.15	181	181
MUTAG	188	2.98	28	66
ENZYMES	600	2.00	126	149
D&D	1,178	2.42	5,748	14,267
NCI ₁	4,208	2,00	462	468
NCI ₄₁	27,965	17.23	462	468
NCI ₁₀₉	4,256	2.00	462	468
NCI ₁₆₇	80,581	8.38	482	478
NCI ₂₂₀	900	3.10	239	255

Institute) datasets⁶, MUTAG, ENZYMES and D&D⁷. Graphs in ENZYMES and D&D contain no edge labels and represent proteins whereas in all other datasets they contain both node and edge labels and represent chemical compounds. In the four PTC datasets, we follow the setting of [78] and assign graphs labelled as CE, SE, or P to the positive class while graphs labelled NE or N were assigned to the negative class. The main properties of each dataset are summarised in Table 4.2.

In order to be able to run FastWY until termination, the maximum size of subgraphs in the search space had to be artificially limited. For example, considering subgraphs of up-to ten nodes as candidate patterns, the FastWY algorithm had not finished its execution on the ENZYMES dataset after two weeks of computation. In contrast, Westfall-Young light took only 3.6 hours to complete the same analysis. As a result, the number of nodes in candidate subgraphs was limited to: (i) 15 in NCI₁, NCI₁₀₉, and NCI₂₂₀; (ii) 10 in MUTAG, NCI₄₁, and NCI₁₆₇; and (iii) 8 in ENZYMES. In D&D and the four PTC datasets, limiting the size of candidate subgraphs was unnecessary to allow FastWY to complete its execution.

4.4.2 Runtime and memory usage

The main experimental result in this chapter is an exhaustive comparison of the overall runtime and memory usage of our proposed approach, the Westfall-Young light algorithm, and the baseline method, the FastWY algorithm, for 20 significant itemset mining and 12 significant subgraph mining datasets. In all cases, $j_p = 10,000$ random permutations were used to empirically estimate the FWER and obtain a corrected significance threshold δ_{wy} that upper-bounds the FWER by $\alpha = 0.05$, a standard choice across many scientific disciplines. Figures 4.4 and 4.5 depict the results for significant itemset mining and significant subgraph mining, respectively.

6. <https://pubchem.ncbi.nlm.nih.gov/>

7. The datasets MUTAG, ENZYMES, and D&D were obtained from <http://mlcb.is.tuebingen.mpg.de/Mitarbeiter/Nino/Graphkernels/data.zip>

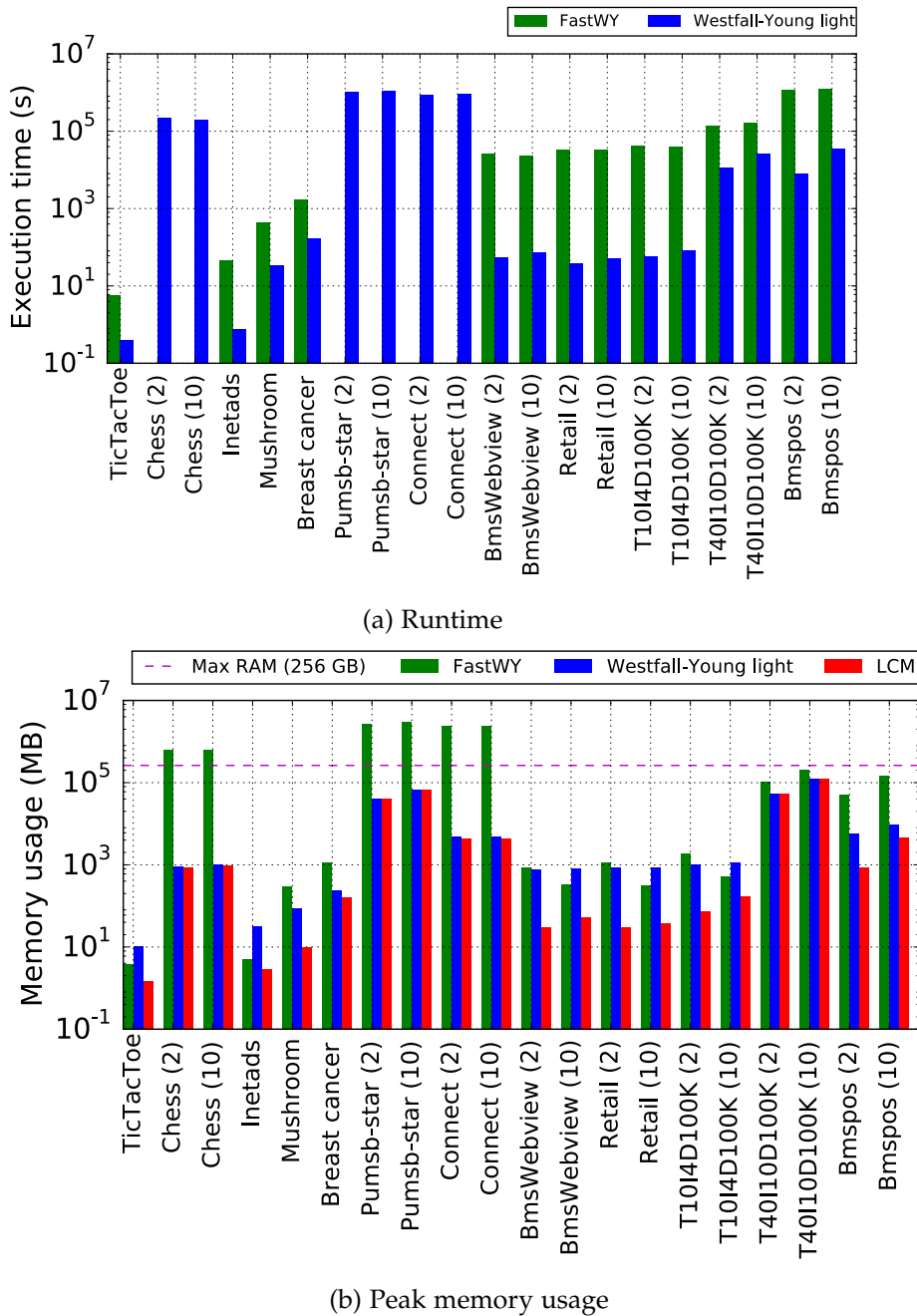
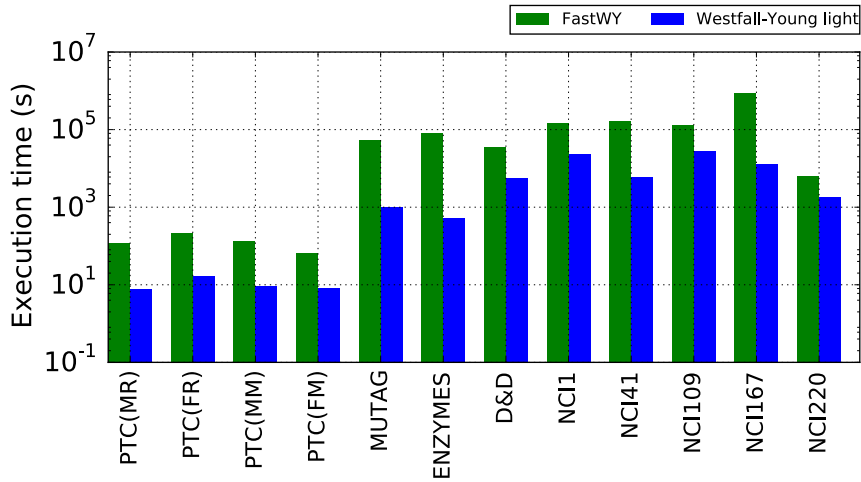
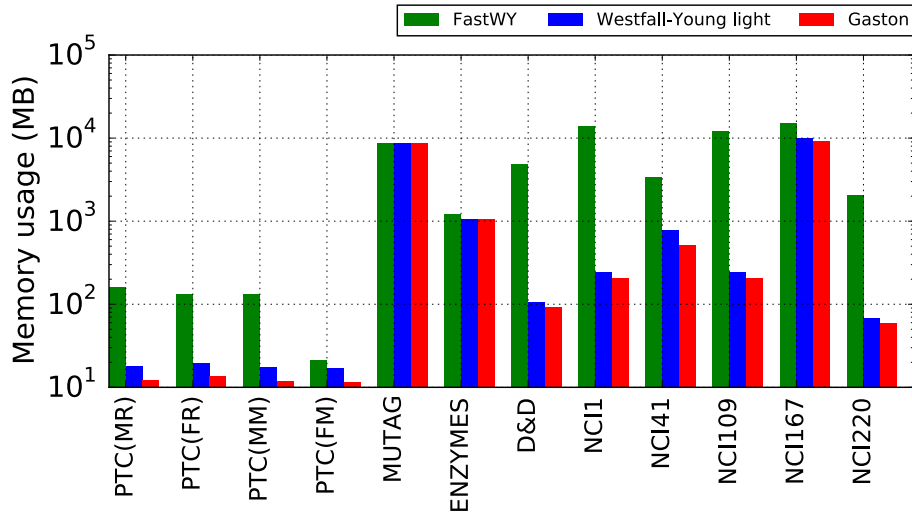


Figure 4.4. – Comparison of runtime and memory usage between Westfall-Young light and FastWY in 20 significant itemset mining experiments, using $j_p = 10,000$ random permutations. Numbers attached to (unlabelled) dataset names denote the ratio n/n_1 .

In terms of runtime, our proposed approach Westfall-Young light can be seen to be two to three orders of magnitude faster than the baseline FastWY across the significant itemset mining experiments and one to two orders of magnitude faster in the significant subgraph mining experiments. The runtime gap between both approaches appears to be heavily dataset-dependent; however, there is a clear trend



(a) Runtime



(b) Peak memory usage

Figure 4.5. – Comparison of runtime and memory usage between Westfall-Young light and FastWY in 12 significant subgraph mining experiments, using $j_p = 10,000$ random permutations.

indicating that the size of this gap increases with the absolute execution time required to analyse the dataset. This strongly suggests that Westfall-Young light scales more gently than FastWY in terms of runtime when analysing real-world datasets.

Figures 4.4 and 4.5 reveal two different qualitative scenarios as far as memory usage is concerned:

1. In 17 out of 32 experiments, Westfall-Young light and FastWY seem to use approximately the same amount of memory. It is worth noting that precisely in these cases for which both approaches exhibit the same memory footprint, simply running the pattern mining algorithm (i.e. LCM or Gaston) also leads to the same memory usage. Therefore, memory usage in these datasets is dominated by the underlying pattern mining algorithm rather than the significant pattern

mining-specific subroutines, leaving little room for improvement and justifying why *Westfall-Young light* and *FastWY* have essentially the same performance.

2. In all other 15 out of 32 experiments, *FastWY* requires a vastly larger amount of memory than *Westfall-Young light*. Perhaps even most importantly, in 6 out of these 15 cases, the memory usage of *FastWY* soars up to the point where the algorithm is unable to complete the analysis. This phenomenon occurred when applying *FastWY* to the datasets Chess, Pumsb-star and Connect. The memory usage of *FastWY* reported in Figure 4.4(b) for these three datasets is a conservative lower bound⁸ on the actual amount; the true memory usage if the analysis had been successfully completed could have been much larger. In contrast, the *Westfall-Young light* algorithm could complete the analysis under the exact same circumstances without further difficulties.

The extreme behaviour of *FastWY* in terms of memory usage is caused by its need to store intermediate computations in memory to alleviate the runtime overhead of repeating pattern enumeration j_p times. In particular, *FastWY* resorts to storing in memory the values of $\{g_S(x_i)\}_{i=1}^n$ for each pattern S that is enumerated by the algorithm, thus avoiding the need to recompute $\{g_S(x_i)\}_{i=1}^n$ the next time the pattern is enumerated in a different resampled dataset. While this design choice allows *FastWY* to complete the analysis of small-sized datasets in a reasonable amount of time, it makes the storage complexity of *FastWY* directly proportional to the number of testable patterns in the dataset, explaining the poor scaling with dataset size observed during the experiments. The *Westfall-Young light* algorithm completely does away with this need, leading to a much more manageable scaling of memory usage with dataset size despite being also considerably faster.

In conclusion, our experiments suggest that the *FastWY* algorithm can only handle successfully small-to-moderate sized problems, exhibiting poor scaling characteristics in larger datasets. Our contribution, the *Westfall-Young light* algorithm, offers a considerable improvement in terms of runtime and memory usage, allowing larger datasets to be analysed. Nevertheless, despite this evident progress, Figures 4.4 and 4.5 clearly indicate that permutation testing-based significant pattern mining is still a highly computationally-demanding task; scaling these algorithms to handle datasets with millions of features remains an open problem.

4.4.3 Final support for pattern mining

The number r_S of samples in a dataset \mathcal{D} for which a pattern S is present is often referred to as the *support* of the pattern in \mathcal{D} . As detailed in Section 3.4, when using Pearson's χ^2 test or Fisher's exact test, if an untestable pattern S has support $r_S \leq \min(n_1, n - n_1)$ then all of its descendants in the pattern enumeration tree must be untestable as well. Exploiting the fact that the minimum attainable P-value

8. This lower bound was computed by estimating the memory overhead incurred by *FastWY* under the assumption that it would enumerate the same number of patterns as *Westfall-Young light*. This is a conservative lower bound for two reasons: (i) *FastWY* enumerates a much larger number of patterns than *Westfall-Young light*, due to its need to compute the most significant P-value of all j_p resampled datasets and (ii) it completely neglects the memory usage of LCM, the underlying pattern mining algorithm, which can itself account for a large proportion of the total memory usage.

function $p_{\min}(r_S)$ corresponding to both test statistics is monotonically decreasing for $r_S \in \llbracket 0, \min(n_1, n - n_1) \rrbracket$, one can compactly rewrite the pruning condition for a pattern S in terms of the support r_S of the pattern: for each $\delta \in [0, 1]$ there exists a *minimum support* $r_{\min}(\delta) \in \llbracket 0, \min(n_1, n - n_1) \rrbracket$ such that the pruning condition of Section 3.4 evaluates to true at level δ if and only if $r_S < r_{\min}(\delta)$. Moreover, the minimum support $r_{\min}(\delta)$ at level δ clearly satisfies that $r_{\min}(\delta') \leq r_{\min}(\delta)$ whenever $\delta \leq \delta'$.

The runtime of significant pattern mining algorithms, including LAMP 2.0 (Algorithm 3.2), FastWY (Algorithm 4.1) and Westfall-Young light (Algorithm 4.2), depends mostly on the total number of patterns enumerated by the algorithm. In particular, this means that the computational complexity incurred by analysing a dataset \mathcal{D} with LAMP 2.0 is approximately proportional to the total number of patterns $S \in \mathcal{M}$ that have a support $r_S \geq r_{\min}(\delta_{\text{tar}})$ in \mathcal{D} . Similarly, using Westfall-Young light instead leads to a computational complexity proportional to the number of patterns $S \in \mathcal{M}$ that have a support $r_S \geq r_{\min}(\delta_{\text{wy}})$ in \mathcal{D} . However, the analysis for FastWY is more complex. Even under the assumption that intermediate computations are stored in memory, thus eliminating the overhead due to repeating pattern enumeration j_p times in exchange for increased memory usage, FastWY still needs to compute $\left\{ p_{\text{ms}}^{(k)} \right\}_{k=1}^{j_p}$ exactly for all j_p resampled datasets. As a consequence, its runtime is loosely proportional to the total number of patterns $S \in \mathcal{M}$ that have a support $r_S \geq r_{\min}(\tilde{\delta}_{\text{fastwy}})$ in \mathcal{D} , where $\tilde{\delta}_{\text{fastwy}} = \max \left\{ p_{\text{ms}}^{(k)} \mid k = 1, \dots, j_p \right\}$. Since $\delta_{\text{wy}} \ll \tilde{\delta}_{\text{fastwy}}$ holds in practice, the *final support* $r_{\min}(\tilde{\delta}_{\text{fastwy}})$ of FastWY is often much smaller than the *final support* $r_{\min}(\delta_{\text{wy}})$ of Westfall-Young light.

This phenomenon is described quantitatively in Figure 4.6, which compares the final support of both approaches when analysing all datasets described in the previous section. The empirical results clearly support the theoretical intuition: the final support of FastWY is often much smaller than that of Westfall-Young light. This has profound implications for the difference in computational efficiency between both approaches. Since most real-world datasets obey a power-law distribution [79], which makes patterns with low supports much more abundant than those with large supports, even just a small decrease in the final support might drastically increase the number of patterns that need to be enumerated by the algorithm and, hence, the overall runtime. Therefore, the results depicted in Figure 4.6 offer at least a partial explanation for the vast difference in runtime between Westfall-Young light and FastWY observed in the previous section.

4.4.4 Statistical power

A useful proxy to compare the statistical power of distinct FWER-controlling approaches is to measure the resulting FWER when applied to a dataset for which the global null hypothesis holds. An optimal method, able to exactly compute $\delta^* = \max \{ \delta \mid \text{FWER}(\delta) \leq \alpha \}$, will attain a FWER virtually identical to the target FWER α . However, suboptimal approaches will in general obtain a corrected significance threshold $\hat{\delta} < \delta^*$, leading to a loss of statistical power as well as to the resulting FWER being strictly smaller than α . Therefore, by measuring how close

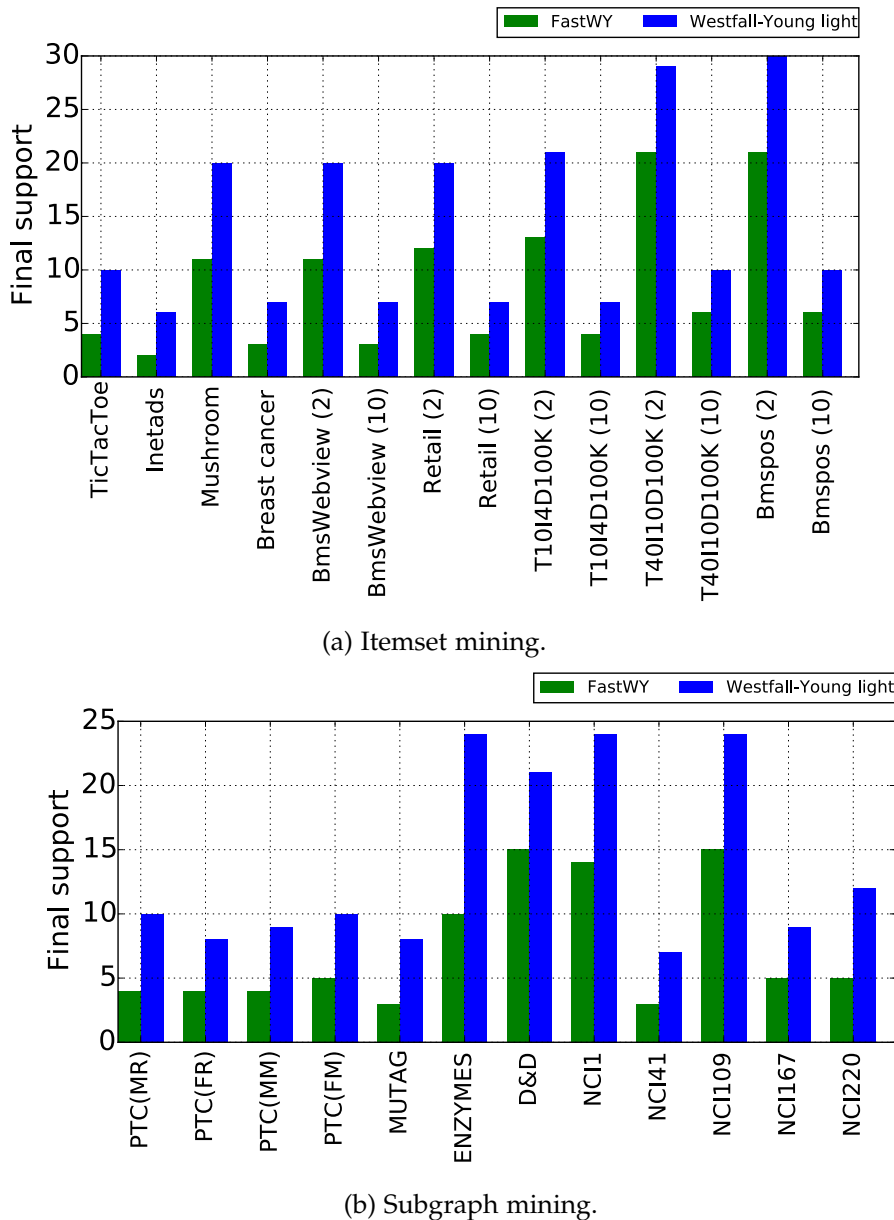


Figure 4.6. – Comparison of the final support corresponding to Westfall-Young light and FastWY. Datasets for which FastWY was unable to complete its execution due to memory limitations were excluded from the figure.

the resulting FWER is to α , one can estimate how close a certain method is to the optimal behaviour among all single-step FWER-controlling approaches in terms of statistical power. In this section, we compare the resulting FWER of permutation testing-based significant pattern mining methods to that of the LAMP algorithm [26], which uses Tarone’s improved Bonferroni correction for discrete data to obtain a corrected significance threshold that controls the FWER.

The parameter of most relevance for this experiment is the number j_p of random permutations used to estimate the FWER. Increasing j_p reduces the variance of the FWER estimator, leading to more stable performance at the expense of increased

runtime and storage complexity. To investigate the effect of varying the number j_p of random permutations, *Westfall-Young light* was executed for 10 different values of j_p between $j_p = 1,000$ and $j_p = 10,000$ in steps of $\Delta j_p = 1,000$. For each pair (dataset, j_p), the experiment was repeated a total of 100 times to obtain the median empirical FWER as a function of j_p , as well as the corresponding 5%–95% confidence interval. In all experiments, the target FWER was set to $\alpha = 0.05$.

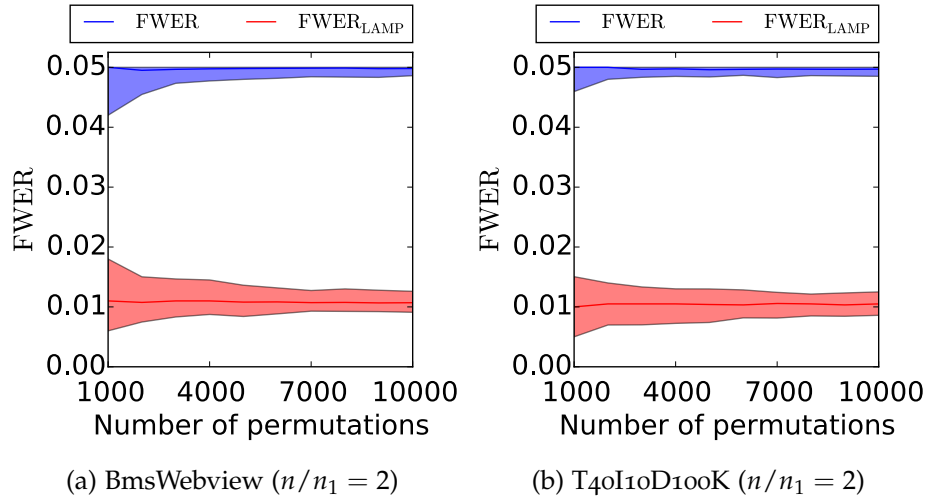


Figure 4.7. – Empirical FWER versus j_p for two representative significant itemset mining datasets.

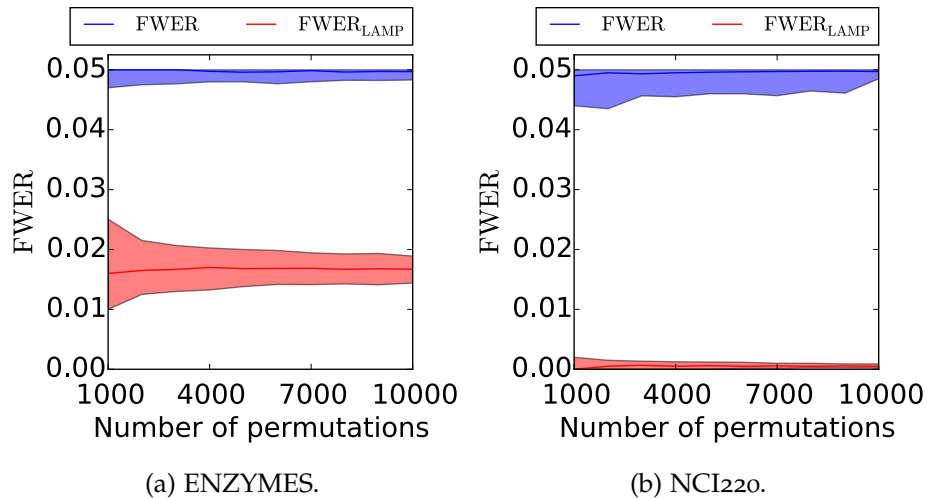


Figure 4.8. – Empirical FWER versus j_p for two representative significant subgraph mining datasets.

Figures 4.7 and 4.8 depict the results for two representative significant itemset mining datasets (BmsWebview and T40I10D100K) and two representative significant subgraph mining datasets (ENZYMES and NCI220), respectively. The most salient feature of these experiments is that the LAMP algorithm, which employs Tarone’s method to control the FWER, is still a considerably over-conservative approach. Its

resulting FWER oscillates between $\alpha/2$ and $\alpha/100$, depending on the dataset. In contrast, `Westfall-Young light` attains a median FWER close to α , regardless of the number j_p of random permutations used, thus being in a sense close to optimal in terms of statistical power among single-step FWER-controlling procedures. The performance of `Westfall-Young light` is particularly robust to the choice of j_p . At the lower end of the range, i.e. $j_p = 1,000$, the resulting FWER of `Westfall-Young light` does exhibit some variability across repetitions of the experiment. Nevertheless, even the worst outcomes can be seen to be considerably closer to α than even the best realisations of LAMP. Moreover, if increasing the computational complexity by a factor of 10 is feasible, using $j_p = 10,000$ yields a very narrow range of variability across repetitions.

In summary, our experimental results confirm the intuition that permutation testing-based significant pattern mining can drastically outperform approaches based on Tarone's method in terms of statistical power. Nonetheless, this enhancement comes at the price of a sharp increase in computational complexity. The optimal choice of approach will therefore be heavily application-dependent.

The need to incorporate into the model covariate factors that might have a confounding effect is an ubiquitous problem in computational biology and clinical data analysis. By neglecting to account for such covariates, an algorithm might discover many spurious patterns whose association with the class labels is entirely mediated by confounding.

In this chapter we present the Fast Automatic Conditional Search (FACS) algorithm [50], a novel significant pattern mining approach that can account for a categorical covariate with an arbitrary number of categories, allowing to drastically reduce spurious false positives due to confounding effects without sacrificing neither computational efficiency nor statistical power.

The remainder of this chapter is organised as follows. Section 5.1 states the problem of correcting for a confounding covariate in significant pattern mining and further elaborates on the motivations to come up with an efficient solution. Section 5.2 extends the background on statistical association testing provided in Section 2.2 by introducing the Cochran-Mantel-Haenszel (CMH) test [46], which generalises Pearson's χ^2 test to conditional association testing. Next, Sections 5.3 and 5.4 present the theoretical foundation of our contribution, the FACS algorithm. In particular, Section 5.3 details the derivation of the minimum attainable P-value for the CMH test whereas Section 5.4 introduces a valid, computationally efficient pruning condition. Section 5.5 discusses low-level implementation considerations, explains how to extend FACS to permutation testing-based significant pattern mining and gives pointers to related work. Finally, the results of an experimental study to assess the computational efficiency, statistical power and false discovery rate of the FACS algorithm are described in Section 5.6.

5.1 INTRODUCTION

Let $G_S(X)$ denote the binary random variable indicating the occurrence of pattern S in an input sample X , Y the binary class label and C be a random variable representing a covariate factor that takes values in a domain \mathcal{C} . We say that the covariate C has a confounding effect on the statistical association between $G_S(X)$ and Y when:

- (i) $G_S(X)$ and Y are *marginally* statistically associated, i.e., $G_S(X) \not\perp Y$. As discussed in Section 2.2, this is the case if and only if $\Pr(G_S(X) = g_S(x), Y = y) \neq \Pr(G_S(X) = g_S(x))\Pr(Y = y)$ for some $(g_S(x), y) \in \{0, 1\}^2$.
- (ii) $G_S(X)$ and Y are *conditionally independent given C*, i.e., $G_S(X) \perp Y \mid C$. This occurs if and only if $\Pr(G_S(X) = g_S(x), Y = y \mid C = c) = \Pr(G_S(X) = g_S(x) \mid C = c)\Pr(Y = y \mid C = c)$ for all $c \in \mathcal{C}$ and $(g_S(x), y) \in \{0, 1\}^2$.

Intuitively, condition (i) above implies that $G_S(X)$ and Y are statistically associated in the absence of information about C . However, condition (ii) also implies that, once the value c taken by the covariate C is known, $G_S(X)$ carries no further information about Y and, therefore, can be discarded. In most applications, patterns $S \in \mathcal{M}$ satisfying

conditions (i) and (ii) are spurious findings that should not be retrieved by a mining algorithm.

These patterns will be of little use to the practitioner, as they provide no additional information about the class membership of an input sample beyond the information that is already contained in the covariate. In many applications, the covariates are quantities that can be measured more easily than the input samples $x \in \mathcal{X}$. For example, while x might represent the genotype of a patient at a set of single nucleotide polymorphisms or describe the measured expression levels of a set of genes, the covariate C often contains simple information such as age, gender, socioeconomic status or genetic ancestry. Thus, from a practical point of view, if a marginally associated pattern $\mathcal{S} \in \mathcal{M}$ is redundant with such a covariate, it might be preferable to simply make use of the covariate when trying to predict the class label Y .

Moreover, not only are patterns satisfying conditions (i) and (ii) of little practical use but, on many occasions, they might represent misleading associations. A particularly common example are spurious associations between genotype and phenotype which arise in genome-wide association studies due to population structure [80]. Often, a phenotype might be strongly associated with the genetic ancestry of an individual, which is obviously itself associated with that individual's genotype. Therefore, if population structure is unaccounted for in a genome-wide association study containing individuals with diverse genetic ancestries, a large number of apparently significant patterns might be retrieved. However, a practitioner might later find that, in fact, most of these patterns simply reflect genotypic motifs that differ between individuals with distinct genetic ancestries and provide no additional insight about the phenotype.

The effect of confounding is illustrated in Figure 5.1, which revisits the significant itemset mining dataset first introduced in Figure 2.1 of Section 2.1. In this example, patterns \mathcal{S}_1 and \mathcal{S}_2 are both marginally associated with the class labels, being enriched among samples of class $Y = 1$. However, Figure 5.1 incorporates a factor not present in the original example: a categorical covariate C with $k = 2$ categories, describing the genetic ancestry of a sample. The inclusion of C changes the interpretation of the associations that patterns \mathcal{S}_1 and \mathcal{S}_2 represent. While the occurrence of pattern \mathcal{S}_1 still carries additional information not contained in the covariate C , pattern \mathcal{S}_2 can be seen to be entirely redundant with C . Thus, following the discussion above, pattern \mathcal{S}_2 might be considered a spurious association that should not be retrieved.

All methods that have been discussed in this thesis so far aim at finding all patterns $\mathcal{S} \in \mathcal{M}$ that are (marginally) statistically associated with the class labels, i.e. they look for the set of patterns $\{\mathcal{S} \in \mathcal{M} \mid G_{\mathcal{S}}(X) \not\perp Y\}$. However, an approach able to correct for the effect of a covariate C would aim to find the set of patterns $\{\mathcal{S} \in \mathcal{M} \mid G_{\mathcal{S}}(X) \not\perp Y \mid C\}$ instead. In particular, a pattern $\mathcal{S} \in \mathcal{M}$ satisfying conditions (i) and (ii) above would *not* be retrieved by the latter formulation yet it would be deemed significant by the former. Consequently, significant pattern mining approaches such as LAMP 2.0 (Algorithm 3.2) and Westfall-Young light (Algorithm 4.2) are prone to discover many spurious patterns due to confounding, severely limiting their applicability in computational biology and clinical data analysis.

The FACS algorithm, which constitutes the main focus of this chapter, can be understood as an extension of LAMP 2.0, described in Section 3.2. However, in order to incorporate the covariate C into the model, FACS replaces Pearson's χ^2 test or

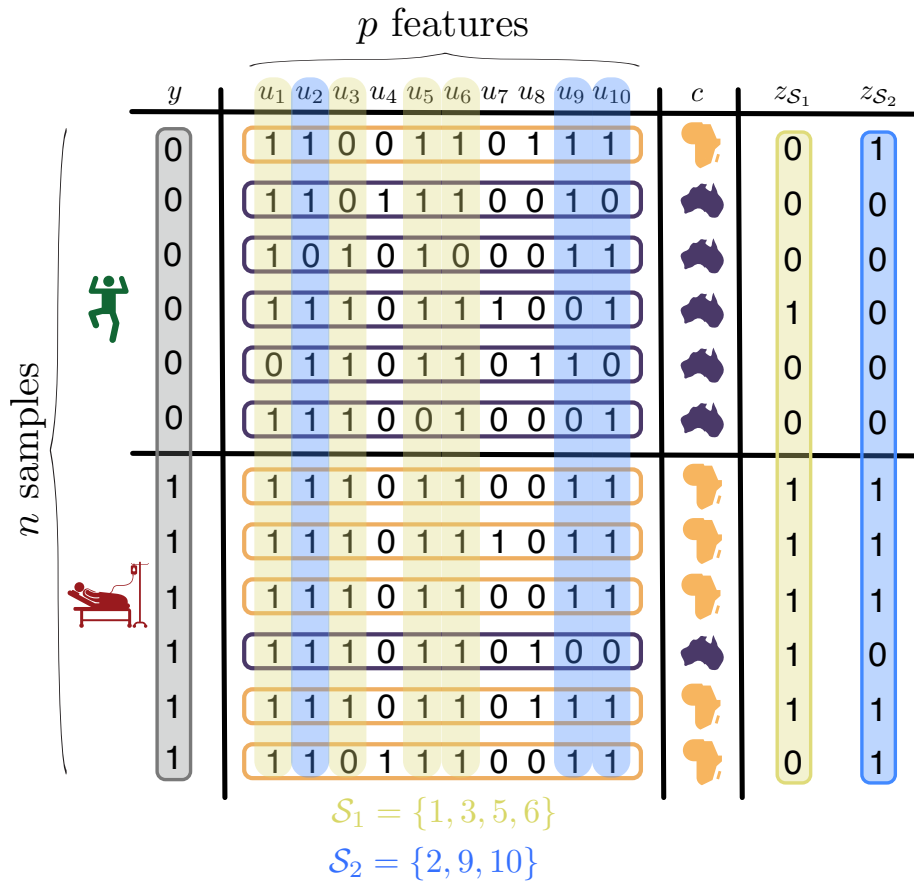


Figure 5.1. – An illustration of the effect of confounding on a toy significant itemset mining problem. A categorical covariate C with $k = 2$ categories (orange and purple) has been introduced. Two patterns, S_1 and S_2 , are marginally associated with the class labels Y . Pattern S_1 remains associated with Y given the covariate C . On the contrary, pattern S_2 carries no information about Y given that the value of C is known, i.e. S_2 is conditionally independent of Y given C .

Fisher’s exact test by the CMH test. Unlike the former two, the CMH test allows assessing the *conditional* association of two binary random variables $G_S(X)$ and Y given a categorical random variable C with k categories, making it an ideal choice for this task. Nevertheless, replacing the test statistic has profound implications for Tarone’s method and its integration into the pattern mining algorithm. In particular, it is necessary to:

- (i) Prove that a minimum attainable P-value exists for the CMH test and devise a tractable expression to evaluate it.
- (ii) Propose a novel search space pruning criterion that applies to the CMH test.

In the next sections we will introduce the CMH test and describe techniques to solve each of these two open problems, culminating in our proposed approach, the FACS algorithm.

5.2 CONDITIONAL ASSOCIATION TESTING IN SIGNIFICANT PATTERN MINING

Section 2.2 introduced Pearson's χ^2 test and Fisher's exact test, two test statistics able to assess the statistical association between two binary random variables. Given an input dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, these can be used to test the (marginal) association between the class labels Y and the binary random variable $G_S(X)$ indicating the occurrence of a pattern $S \in \mathcal{M}$ in an input sample X .

This section however concerns the case in which we are given a dataset $\mathcal{D} = \{(x_i, y_i, c_i)\}_{i=1}^n$, consisting of n observations $x \in \mathcal{X}$ belonging to one of two classes $y \in \{0, 1\}$. Additionally, each of the n observations is now also tagged with a categorical covariate $c \in \{1, 2, \dots, k\}$, where k is the number of distinct categories that an outcome of random variable C can belong to. Unlike in the setting of Section 2.2, the new goal is to test the conditional association between $G_S(X)$ and Y given C . This is precisely what the CMH test was designed for.

Intuitively, the CMH test can be seen as a way to tackle this problem by reducing it to a set of k instances of Pearson's χ^2 test and then combining the k resulting statistics appropriately. By definition, $G_S(X)$ and Y are conditionally independent given C and, therefore, *not* conditionally associated given C , if $\Pr(G_S(X) = g_S(x), Y = y \mid C = c) = \Pr(G_S(X) = g_S(x) \mid C = c)\Pr(Y = y \mid C = c) \forall c \in \{1, 2, \dots, k\}$. For each $c = 1, 2, \dots, k$, let $\mathcal{D}(c) = \{(x_i, y_i) \in \mathcal{D} \mid c_i = c\}$ be the set of samples in \mathcal{D} for which the categorical covariate takes value c . The (unknown) joint distribution $\Pr(G_S(X) = g_S(x), Y = y \mid C = c)$ can be empirically approximated using counts derived from the samples in $\mathcal{D}(c)$:

Variables	$g_S(x) = 1$	$g_S(x) = 0$	Row totals
$y = 1$	$a_{S,c}$	$b_{S,c}$	$n_{1,c}$
$y = 0$	$d_{S,c}$	$c_{S,c}$	$n_{0,c}$
Col. totals	$r_{S,c}$	$q_{S,c}$	n_c

The interpretation of these counts is analogous to the unconditional case described in Section 2.2. For instance, $a_{S,c}$ is the number of samples in $\mathcal{D}(c)$ belonging to class $y = 1$ for which pattern S occurs or, equivalently, the number of samples in \mathcal{D} belonging to class $y = 1$ for which pattern S occurs and the covariate takes value c . Thus, an empirical estimate of $\Pr(G_S(X) = 1, Y = 1 \mid C = c)$ could be obtained as $a_{S,c}/n_c$ for each $c \in \{1, 2, \dots, k\}$. The remaining counts can be described in a similar manner.

As a consequence of Proposition 2.1 in Section 2.2, if $G_S(X)$ is conditionally independent of Y given C , the random variable $A_{S,c}$ given margins $n_{1,c}$ and $r_{S,c}$ and sample size n_c follows a hypergeometric distribution with parameters $n_c, n_{1,c}$ and $r_{S,c}$ for all $c \in \{1, 2, \dots, k\}$. Furthermore, under the assumption that all n samples in \mathcal{D} are obtained as i.i.d. draws, it follows that $A_{S,c}$ is statistically independent of $A_{S,c'}$ for any $c \neq c'$, since $\mathcal{D}(c) \cap \mathcal{D}(c') = \emptyset$. Paralleling the derivation of Pearson's χ^2 test described in Section 2.2, the following Z-score can be proposed as a way to additively aggregate the individual Z-scores of the k distinct 2×2 contingency tables:

$$Z_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S) = \frac{\sum_{c=1}^k a_{S,c} - \mathbb{E}[a_{S,c} \mid R_{S,c} = r_{S,c}, N_{1,c} = n_{1,c}, H_0]}{\sqrt{\sum_{c=1}^k \text{Var}[a_{S,c} \mid R_{S,c} = r_{S,c}, N_{1,c} = n_{1,c}, H_0]}}, \quad (5.1)$$

where the term in the denominator follows from the fact that the variance of a sum of independent random variables equals the sum of the variances of each random variable participating in the sum. To simplify the notation, the vectors $\mathbf{a}_S = (a_{S,1}, \dots, a_{S,k})$, $\mathbf{n} = (n_1, \dots, n_k)$, $\mathbf{n}_1 = (n_{1,1}, \dots, n_{1,k})$ and $\mathbf{r}_S = (r_{S,1}, \dots, r_{S,k})$, which contain the values of $a_{S,c}$, n_c , $n_{1,c}$ and $r_{S,c}$ for all k different 2×2 contingency tables, were introduced. The final expression for the CMH test can be obtained by squaring this Z-score and plugging in the values of $\mathbb{E}[a_{S,c} \mid R_{S,c} = r_{S,c}, N_{1,c} = n_{1,c}, H_0]$ and $\text{Var}[a_{S,c} \mid R_{S,c} = r_{S,c}, N_{1,c} = n_{1,c}, H_0]$ as the mean and variance of a hypergeometric distribution with parameters n_c , $n_{1,c}$ and $r_{S,c}$, resulting in:

$$T_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S) = \frac{\left(\sum_{c=1}^k a_{S,c} - r_{S,c} \frac{n_{1,c}}{n_c}\right)^2}{\sum_{c=1}^k \frac{r_{S,c}}{n_c} \frac{n_c - r_{S,c}}{n_c} \frac{n_c - n_{1,c}}{n_c - 1} n_{1,c}}. \quad (5.2)$$

The CMH test statistic $T_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S)$ aggregates evidence against the null hypothesis H_0 that $G_S(X)$ is conditionally independent of Y given C across all k distinct 2×2 contingency tables. Large values of $T_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S)$ are less likely to occur if the null hypothesis holds.

The null distribution of the CMH test statistic can be approximated in a similar way as the null distribution of the Pearson's χ^2 test statistic. Provided that the sample size n is sufficiently large, $Z_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S)$ will be approximately distributed as a standard normal under the null hypothesis H_0 . Thus, the null distribution of $T_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S)$ can be approximated as a χ_1^2 distribution, with a two-tailed P-value

$$p_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S) = 1 - F_{\chi_1^2}(T_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S)), \quad (5.3)$$

where $F_{\chi_1^2}(\bullet)$ is the cumulative density function of a χ_1^2 distribution.

Figure 5.2 depicts the result of applying the CMH test to assess the statistical significance of pattern \mathcal{S}_2 in the dataset shown previously in Figure 5.1. The 2×2 contingency table built using all samples in the dataset \mathcal{D} , shown at the top of the figure (light blue), suggests a (marginal) association between $G_{\mathcal{S}_2}(X)$ and Y . Indeed, if Pearson's χ^2 test is used to compute a P-value, one obtains $p_{\text{pearson}}(a_{\mathcal{S}_2} \mid n, n_1, r_{\mathcal{S}_2}) = 0.021$, a rather significant result taking into consideration that the sample size is only $n = 12$. However, if this contingency table is split into $k = 2$ distinct tables according to the value of the categorical covariate C , leading to the orange and purple contingency tables shown at the bottom of the figure, this association can be seen to disappear. In particular, those tables are so extreme that only one outcome for $a_{\mathcal{S}_2,c}$ is possible in each case, i.e. $a_{\mathcal{S}_2,c,\min} = a_{\mathcal{S}_2,c,\max}$ holds for both contingency tables. As a consequence, the CMH test leads to an entirely non-significant P-value $p_{\text{cmh}}(\mathbf{a}_{\mathcal{S}_2} \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_{\mathcal{S}_2}) = 1$, successfully eliminating the confounding effect of the covariate. In contrast, it can be readily verified that if this analysis is repeated for pattern \mathcal{S}_1 , which is not affected by confounding in the example of Figure 5.1, the CMH test still returns a rather significant P-value, $p_{\text{cmh}}(\mathbf{a}_{\mathcal{S}_1} \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_{\mathcal{S}_1}) = 0.029$.

5.3 THE MINIMUM ATTAINABLE P-VALUE FOR THE CMH TEST

As Pearson's χ^2 test and Fisher's exact test, the CMH test is based on discrete data and, as a consequence, it can only attain a finite number of distinct values. As

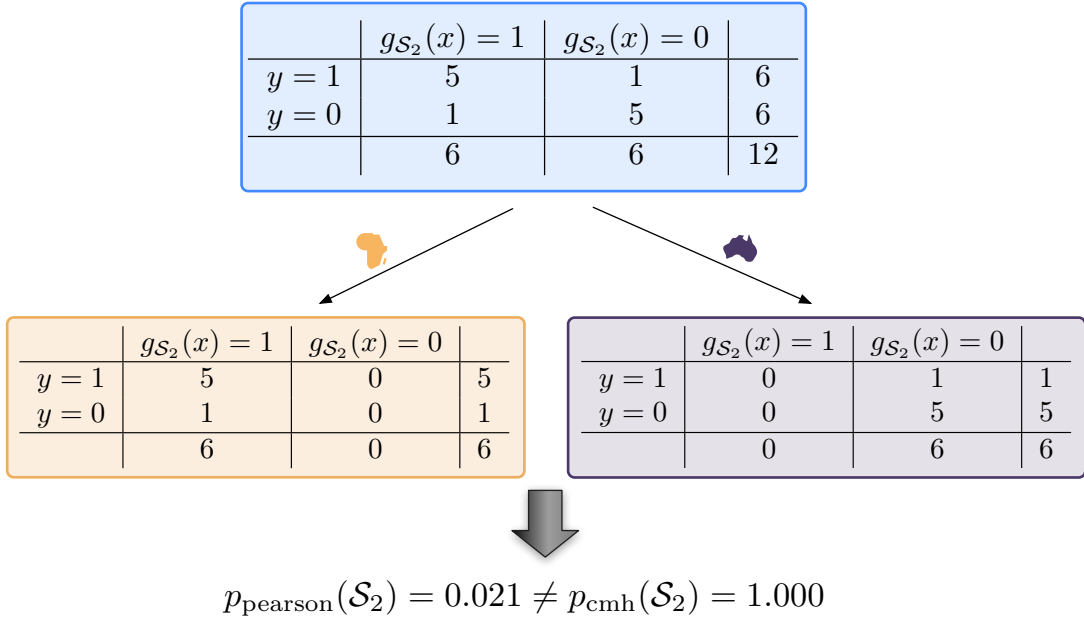


Figure 5.2. – Application of the CMH test to the toy significant pattern mining dataset in Figure 5.1. This example shows an assessment of the statistical association between the class labels Y and the occurrence of pattern S_2 in the input samples, given the categorical covariate C . The contingency table shown at the top of the figure (blue) is constructed using all samples in the dataset \mathcal{D} . In contrast, the two contingency tables shown at the bottom of the figure are obtained from the stratified datasets $\mathcal{D}(1)$, which contains only samples of African ancestry (orange), and $\mathcal{D}(2)$, which comprises only samples of Australian ancestry (purple).

discussed in Section 2.3, this property implies the existence of a minimum attainable P-value strictly larger than zero, allowing one to leverage Tarone's concept of testability. This observation is summarised in the following proposition:

Proposition 5.1 (Minimum attainable P-value function for the CMH test). *Let $\mathbf{a}_{S,\min}$ and $\mathbf{a}_{S,\max}$ be k -dimensional vectors defined as:*

$$\mathbf{a}_{S,\min} = (a_{S,1,\min}, a_{S,2,\min}, \dots, a_{S,k,\min}), \quad (5.4)$$

$$\mathbf{a}_{S,\max} = (a_{S,1,\max}, a_{S,2,\max}, \dots, a_{S,k,\max}), \quad (5.5)$$

where $a_{S,c,\min} = \max(0, r_{S,c} - (n_c - n_{1,c}))$ and $a_{S,c,\max} = \min(n_{1,c}, r_{S,c})$ for each $c = 1, 2, \dots, k$. Then, the minimum attainable P-value function for the CMH test is given by:

$$p_{\min}(\mathbf{r}_S) = 1 - F_{\chi_1^2}(\max(T_{\text{cmh}}(\mathbf{a}_{S,\min} | \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S), T_{\text{cmh}}(\mathbf{a}_{S,\max} | \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S))). \quad (5.6)$$

In particular, this implies that $p_{\min}(\mathbf{r}_S)$ can be evaluated in $O(k)$ time, where k is the number of categories for the categorical covariate C .

Proof. Equation (5.2) can be rewritten as:

$$T_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S) = \frac{\left(a_{S,\text{tot}} - \sum_{c=1}^k r_{S,c} \frac{n_{1,c}}{n_c}\right)^2}{\sum_{c=1}^k \frac{r_{S,c}}{n_c} \frac{n_c - r_{S,c}}{n_c} \frac{n_c - n_{1,c}}{n_c - 1} n_{1,c}}, \quad (5.7)$$

where $a_{S,\text{tot}} = \sum_{c=1}^k a_{S,c}$ has been introduced. As described in Section 2.3, given fixed margins $n_{1,c}$, $r_{S,c}$ and sample size n_c , each count $a_{S,c}$ can only take values in the set $a_{S,c} \in \llbracket a_{S,c,\text{min}}, a_{S,c,\text{max}} \rrbracket$, where $a_{S,c,\text{min}} = \max(0, r_{S,c} - (n_c - n_{1,c}))$ and $a_{S,c,\text{max}} = \min(n_{1,c}, r_{S,c})$. Thus, $a_{S,\text{tot}} \in \llbracket a_{S,\text{tot},\text{min}}, a_{S,\text{tot},\text{max}} \rrbracket$, where $a_{S,\text{tot},\text{min}} = \sum_{c=1}^k a_{S,c,\text{min}}$ and $a_{S,\text{tot},\text{max}} = \sum_{c=1}^k a_{S,c,\text{max}}$. Equation (5.7) clearly shows that $T_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S)$ is a strictly convex function of $a_{S,\text{tot}}$. Therefore, it will be maximised and, hence, $p_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S)$ minimised, either when $a_{S,\text{tot}} = a_{S,\text{tot},\text{min}}$ or $a_{S,\text{tot}} = a_{S,\text{tot},\text{max}}$. Equivalently, this occurs when $\mathbf{a}_S = \mathbf{a}_{S,\text{min}}$ or $\mathbf{a}_S = \mathbf{a}_{S,\text{max}}$, thus proving Equation (5.6). Finally, note that evaluating $T_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S)$ for an arbitrary \mathbf{a}_S requires $O(k)$ operations. Since $p_{\text{min}}(\mathbf{r}_S)$ can be computed by evaluating $T_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S)$ at two distinct values of \mathbf{a}_S , namely $\mathbf{a}_{S,\text{min}}$ and $\mathbf{a}_{S,\text{max}}$, $p_{\text{min}}(\mathbf{r}_S)$ can be computed with $O(k)$ operations as well, thus concluding the proof. \square

Proposition 5.1 above offers a solution to the first of the two challenges mentioned at the beginning of this chapter: providing a computationally tractable expression to evaluate the minimum attainable P-value function $p_{\text{min}}(\mathbf{r}_S)$ for the CMH test. This is, to the best of our knowledge, the first result concerning the use of Tarone's concept of testability in conjunction with the CMH test. From an algorithmic perspective, Proposition 5.1 also suggests the first of the two key modifications that need to be performed to Algorithm 3.2 in order to obtain our novel FACS algorithm: Line 8, which is responsible for evaluating the minimum attainable P-value $p_{S,\text{min}}$ of a pattern S , must now follow Equation (5.6) rather than the formulae described in Section 3.3.

5.4 A SEARCH SPACE PRUNING CONDITION FOR THE CMH TEST

In Section 3.4, a search space pruning condition valid for Pearson's χ^2 test and Fisher's exact test was derived. The fundamental principle on which that pruning criterion relies is that, for fixed n_1 and n , the minimum attainable P-value for these test statistics is a monotonically decreasing function $p_{\text{min}}(r_S)$ of r_S in the range $r_S \in \llbracket 0, \min(n_1, n - n_1) \rrbracket$. This implies that, if a pattern S is untestable at level δ and satisfies $r_S \leq \min(n_1, n - n_1)$, all its descendants S' in the pattern enumeration tree will be untestable at level δ as well and can be pruned from the search space. In this section, an alternative pruning condition which is valid for the CMH test will be proposed.

The first key observation is that, for fixed \mathbf{n} and \mathbf{n}_1 , the minimum attainable P-value $p_{S,\text{min}}$ of a pattern S when using the CMH test can be obtained as a multivariate function of k variables: $r_{S,1}, r_{S,2}, \dots, r_{S,k}$. This will be denoted by $p_{S,\text{min}} = p_{\text{min}}(\mathbf{r}_S)$, where the dependence of $p_{S,\text{min}}$ on \mathbf{n} and \mathbf{n}_1 is kept implicit to avoid cluttering the notation. By applying the apriori property of pattern mining, stated in Proposition 3.1, it can be shown that if S' is a descendant of S in the pattern enumeration tree, then $r_{S',c} \leq r_{S,c}$ will hold for all $c = 1, 2, \dots, k$. Using identical arguments as those exploited

in Section 3.4, a majority of patterns \mathcal{S} in the search space of candidate patterns \mathcal{M} will be relatively rare, satisfying $r_{\mathcal{S},c} \leq \min(n_{1,c}, n_c - n_{1,c})$ for all $c = 1, 2, \dots, k$. This naturally leads to the fundamental question of whether, given that $r_{\mathcal{S},c} \leq \min(n_{1,c}, n_c - n_{1,c})$ for all $c = 1, 2, \dots, k$, the fact that $r_{\mathcal{S}',c} \leq r_{\mathcal{S},c}$ for all $c = 1, 2, \dots, k$ implies that $p_{\min}(\mathbf{r}_{\mathcal{S}'}) \geq p_{\min}(\mathbf{r}_{\mathcal{S}})$ or not. If the answer to this question was affirmative, a pruning condition entirely analogous to the one used for Pearson's χ^2 test and Fisher's exact test would also be valid for the CMH test.

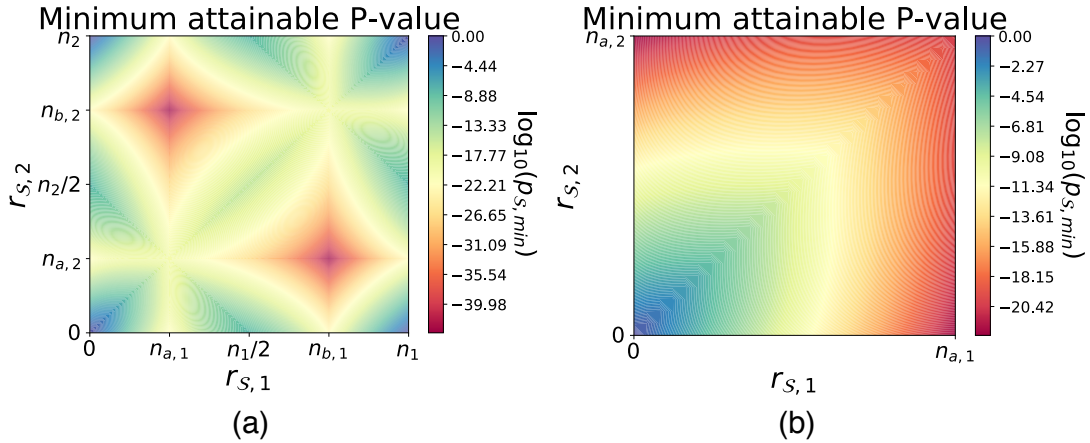


Figure 5.3. – Minimum attainable P-value function $p_{\min}(\mathbf{r}_{\mathcal{S}})$ for the CMH test in a problem with $k = 2$ categories for the covariate. In this example, $n_1 = n_2 = 100$ and $n_{1,1} = 25$, $n_{1,2} = 75$. Thus, category $c = 1$ has a class ratio of $1/4$ while category $c = 2$ has a class ratio of $3/4$. (a) Minimum attainable P-value $p_{\min}(\mathbf{r}_{\mathcal{S}})$ function over the entire domain $\llbracket 0, n_1 \rrbracket \times \llbracket 0, n_2 \rrbracket$. (b) Minimum attainable P-value function $p_{\min}(\mathbf{r}_{\mathcal{S}})$ over the region $\llbracket 0, n_{a,1} \rrbracket \times \llbracket 0, n_{a,2} \rrbracket$, where $n_{a,1} = \min(n_{1,1}, n_1 - n_{1,1})$ and $n_{a,2} = \min(n_{1,2}, n_2 - n_{1,2})$.

Unfortunately, it is easy to come up with non-pathological counterexamples which show that this property does not hold in general. As an example, Figure 5.3 depicts the minimum attainable P-value function $p_{\min}(\mathbf{r}_{\mathcal{S}})$ for the CMH test in a problem with $k = 2$ categories for the covariate. In particular, Figure 5.3(b) illustrates the behaviour of $p_{\mathcal{S},\min}$ as a function of $r_{\mathcal{S},1}$ and $r_{\mathcal{S},2}$ over the region $\llbracket 0, \min(n_{1,1}, n_1 - n_{1,1}) \rrbracket \times \llbracket 0, \min(n_{1,2}, n_2 - n_{1,2}) \rrbracket$. While the function is approximately monotonic when $r_{\mathcal{S},1}$ and $r_{\mathcal{S},2}$ are both sufficiently far from zero, $p_{\min}(\mathbf{r}_{\mathcal{S}})$ is not monotonically decreasing when one of its arguments is small enough, as can be appreciated from the level curves. This has profound implications for the development of a valid pruning criterion, as in principle there is no simple way to make a statement about the minimum attainable P-value $p_{\mathcal{S}',\min}$ of a descendant \mathcal{S}' of a pattern \mathcal{S} based on $p_{\mathcal{S},\min}$ and $\mathbf{r}_{\mathcal{S}}$ alone.

In order to solve this problem, the FACS algorithm uses a monotonically decreasing lower bound of the minimum attainable P-value as a surrogate of $p_{\mathcal{S},\min}$ in its pruning criterion. This surrogate will be referred to as the *lower envelope* of the minimum attainable P-value.

Definition 5.2. Let $\mathcal{S} \in \mathcal{M}$ be a pattern satisfying $r_{\mathcal{S},c} \leq \min(n_{1,c}, n_c - n_{1,c})$ for all $c = 1, 2, \dots, k$. The lower envelope of the minimum attainable P-value $p_{\mathcal{S},\min}$ is defined as:

$$\tilde{p}_{\mathcal{S},\min} = \min_{\mathcal{S}' \supseteq \mathcal{S}} p_{\mathcal{S}',\min}. \quad (5.8)$$

Equivalently, let $\mathcal{B}(\mathbf{r}_{\mathcal{S}}) = \llbracket 0, r_{\mathcal{S},1} \rrbracket \times \llbracket 0, r_{\mathcal{S},2} \rrbracket \times \dots \times \llbracket 0, r_{\mathcal{S},k} \rrbracket$ be the set of all $\mathbf{r}_{\mathcal{S}'}$ satisfying $r_{\mathcal{S}',c} \leq r_{\mathcal{S},c}$ for all $c = 1, 2, \dots, k$. Then, as a consequence of the apriori property of pattern mining, $\tilde{p}_{\mathcal{S},\min}$ can be expressed as the following function $\tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}})$ of $\mathbf{r}_{\mathcal{S}}$:

$$\tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}}) = \min_{\mathbf{r}_{\mathcal{S}' \in \mathcal{B}(\mathbf{r}_{\mathcal{S}})} p_{\min}(\mathbf{r}_{\mathcal{S}'}), \quad (5.9)$$

where the dependence of the minimum attainable P-value $p_{\mathcal{S},\min}$ and its lower envelope $\tilde{p}_{\mathcal{S},\min}$ on $\mathbf{r}_{\mathcal{S}}$ has been made explicit.

Intuitively, $\tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}})$ is defined as the tightest lower bound of the minimum attainable P-value function $p_{\min}(\mathbf{r}_{\mathcal{S}})$, hence the term “lower envelope”, that satisfies $\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}(\mathbf{r}_{\mathcal{S}}) \Rightarrow \tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}'} \geq \tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}})$. This notion is illustrated in Figure 5.4 with a conceptual example.

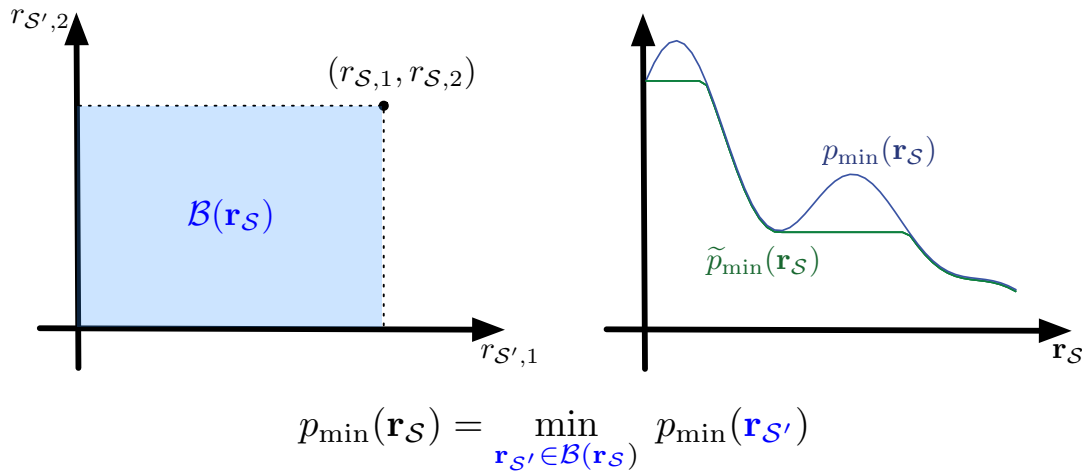


Figure 5.4. – Illustration of the lower envelope $\tilde{p}_{\mathcal{S},\min}$ of the minimum attainable P-value $p_{\mathcal{S},\min}$. To evaluate $\tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}})$, the minimum of $p_{\min}(\mathbf{r}_{\mathcal{S}'})$ over the region $\mathcal{B}(\mathbf{r}_{\mathcal{S}})$ needs to be computed. In this example, this corresponds to minimising $p_{\min}(\mathbf{r}_{\mathcal{S}'})$ over the region shaded in blue on the left of the figure. As a result of this definition, $\tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}})$ is the tightest lower bound of $p_{\min}(\mathbf{r}_{\mathcal{S}})$ that satisfies $\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}(\mathbf{r}_{\mathcal{S}}) \Rightarrow \tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}'} \geq \tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}})$ or, equivalently, that $r_{\mathcal{S}',c} \leq r_{\mathcal{S},c}$ for all $c = 1, 2, \dots, k$ implies that $\tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}'} \geq \tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}})$. This is illustrated, along a one-dimensional slice of $\mathcal{B}(\mathbf{r}_{\mathcal{S}})$, on the right of the figure.

The lower envelope $\tilde{p}_{\mathcal{S},\min}$ is a lower bound of the minimum attainable P-value $p_{\mathcal{S},\min}$ by construction. Also, the fact that $\tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}})$ is monotonically decreasing on $\mathbf{r}_{\mathcal{S}}$, i.e. that $\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}(\mathbf{r}_{\mathcal{S}}) \Rightarrow \tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}'} \geq \tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}})$, is a direct consequence of the way $\tilde{p}_{\mathcal{S},\min}$ is defined. If $\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}(\mathbf{r}_{\mathcal{S}})$, then it follows that $\mathcal{B}(\mathbf{r}_{\mathcal{S}'}) \subseteq \mathcal{B}(\mathbf{r}_{\mathcal{S}})$. Thus,

$\tilde{p}_{\min}(\mathbf{r}_{S'}) = \min_{\mathbf{r}_{S''} \in \mathcal{B}(\mathbf{r}_{S'})} p_{\min}(\mathbf{r}_{S''}) \geq \min_{\mathbf{r}_{S''} \in \mathcal{B}(\mathbf{r}_S)} p_{\min}(\mathbf{r}_{S''}) = \tilde{p}_{\min}(\mathbf{r}_S)$ must hold. These two properties of $\tilde{p}_{S,\min}$ allow proposing the following search space pruning criterion for the CMH test.

Proposition 5.3. [Pruning criterion for the CMH test] *Let $\mathcal{S} \in \mathcal{M}$ be a pattern satisfying:*

- (i) $\tilde{p}_{S,\min} > \hat{\delta}_{\text{tar}}$, i.e. the lower envelope of the minimum attainable P-value is larger than $\hat{\delta}_{\text{tar}}$,
- (ii) $r_{S,c} \leq n_{a,c}$ for all $c = 1, 2, \dots, k$, where $n_{a,c} = \min(n_{1,c}, n_c - n_{1,c})$.

Then, $p_{S',\min} > \hat{\delta}_{\text{tar}} \geq \delta_{\text{tar}}$ for all descendants \mathcal{S}' of \mathcal{S} in the pattern enumeration tree, implying that they can be pruned from the search space. In conclusion, when using the CMH test, `pruning_condition`($\mathcal{S}, \hat{\delta}_{\text{tar}}$) in Line 16 of Algorithm 3.2 evaluates to true if and only if $r_{S,c} \leq n_{a,c}$ for all $c = 1, 2, \dots, k$ and $\tilde{p}_{S,\min} > \hat{\delta}_{\text{tar}}$.

Proof. If \mathcal{S}' is a descendant of \mathcal{S} in the pattern enumeration tree, $\mathbf{r}_{S'} \in \mathcal{B}(\mathbf{r}_S)$ by the apriori property of pattern mining. Hence, as a consequence of the monotonicity of $\tilde{p}_{\min}(\mathbf{r}_S)$, $\tilde{p}_{\min}(\mathbf{r}_{S'}) \geq \tilde{p}_{\min}(\mathbf{r}_S)$. Since the lower envelope $\tilde{p}_{S,\min}$ is a lower bound on the minimum attainable P-value $p_{S,\min}$ and $\tilde{p}_{S,\min} > \hat{\delta}_{\text{tar}}$ by assumption (i), it follows that $p_{S',\min} > \hat{\delta}_{\text{tar}}$. Moreover, since $\hat{\delta}_{\text{tar}} \geq \delta_{\text{tar}}$ holds at any point during the execution of Algorithm 3.2, $p_{S',\min} > \delta_{\text{tar}}$, proving that pattern \mathcal{S}' can be pruned from the search space. \square

The resulting pruning condition for the CMH test is mostly analogous to the pruning criterion for Pearson's χ^2 test and Fisher's exact test described in Proposition 3.4. However, the minimum attainable P-value $p_{S,\min}$ in condition (i), $p_{S,\min} > \hat{\delta}_{\text{tar}}$, is substituted by the lower envelope $\tilde{p}_{S,\min}$. This allows circumventing the difficulties which arise as a consequence of the minimum attainable P-value function $p_{\min}(\mathbf{r}_S)$ not being monotonically decreasing for the CMH test. While the concept of lower envelope of the minimum attainable P-value is only used by FACS in the context of the CMH test, the same principle could be applied to other discrete tests statistics with a non-monotonic minimum attainable P-value function. This might help develop new applications of significant pattern mining that require using domain-specific test statistics.

Nevertheless, an important aspect that remains to be considered is how to efficiently evaluate $\tilde{p}_{S,\min}$. Obtaining the lower envelope of the minimum attainable P-value by naively applying Equation 5.8 would require enumerating and evaluating the minimum attainable P-value $p_{S',\min}$ of all patterns $\mathcal{S}' \supseteq \mathcal{S}$. In other words, in order to verify whether the search space pruning condition applies for a pattern \mathcal{S} according to this naive approach, the very same computations the pruning condition is intended to avoid would have to be performed. Thus, Equation (5.8) is entirely unhelpful from an algorithmic perspective. Alternatively, Equation (5.9) phrases the evaluation of $\tilde{p}_{S,\min}$ as a combinational optimisation problem. Attempting to solve this problem by brute force, i.e. by evaluating $p_{\min}(\mathbf{r}_{S'})$ at each $\mathbf{r}_{S'}$ in $\mathcal{B}(\mathbf{r}_S)$, would result in $\prod_{c=1}^k (r_{S,c} + 1)$ evaluations of the minimum attainable P-value function. Defining $m_S = \left(\prod_{c=1}^k (r_{S,c} + 1) \right)^{\frac{1}{k}}$ to be the geometric mean of $\{r_{S,c} + 1\}_{c=1}^k$, it follows that $p_{\min}(\mathbf{r}_{S'})$ needs to be evaluated $m_S^k = O(n^k)$ times. Since the pruning condition is

assessed for every single pattern $\mathcal{S} \in \mathcal{M}$ that is enumerated along the execution of the algorithm, this would entail an impractical computational overhead. Without a computationally tractable approach to exactly compute the lower envelope of the minimum attainable P-value $\tilde{p}_{\mathcal{S},\min}$, the pruning condition in Proposition 5.3 is nothing more than a theoretical construct, leaving the problem of accounting for covariates in significant pattern mining completely unsolved.

In order to design FACS, one of the main contributions of [50] is an algorithm which solves the optimisation problem defined by Equation (5.9) in only $O(k \log k)$ time, provided that the CMH test is the underlying test statistic¹. Using this novel approach, the computational overhead incurred by evaluating the lower envelope of the minimum attainable P-value $\tilde{p}_{\mathcal{S},\min}$ for the CMH test becomes negligible, rendering Proposition 5.3 computationally tractable. The remainder of this section will be devoted to describe this method in detail and to prove its correctness.

Computing $\tilde{p}_{\mathcal{S},\min}$ in $O(k \log k)$ time

The goal of this section is to propose a computationally efficient approach to assess the pruning condition described in Proposition 5.3. Equivalently, the problem that needs to be solved can be precisely formulated as follows:

Given a k -dimensional vector $\mathbf{r}_{\mathcal{S}}$ satisfying $r_{\mathcal{S},c} \in \llbracket 0, \min(n_{1,c}, n_c - n_{1,c}) \rrbracket$ for all $c = 1, 2, \dots, k$, find a minimiser $\mathbf{r}_{\mathcal{S}'}$ of the minimum attainable P-value function $p_{\min}(\mathbf{r}_{\mathcal{S}'})$ of the CMH test in the set $\mathcal{B}(\mathbf{r}_{\mathcal{S}}) = \llbracket 0, r_{\mathcal{S},1} \rrbracket \times \llbracket 0, r_{\mathcal{S},2} \rrbracket \times \dots \times \llbracket 0, r_{\mathcal{S},k} \rrbracket$.

According to Proposition 5.1, $p_{\min}(\mathbf{r}_{\mathcal{S}'})$ can be written as

$$p_{\min}(\mathbf{r}_{\mathcal{S}'}) = 1 - F_{\chi_1^2}(\max(T_{\max}^l(\mathbf{r}_{\mathcal{S}'}), T_{\max}^u(\mathbf{r}_{\mathcal{S}'}))),$$

where $T_{\max}^l(\mathbf{r}_{\mathcal{S}'}) = T_{\text{cmh}}(\mathbf{a}_{\mathcal{S}',\min} \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_{\mathcal{S}'})$ and $T_{\max}^u(\mathbf{r}_{\mathcal{S}'}) = T_{\text{cmh}}(\mathbf{a}_{\mathcal{S}',\max} \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_{\mathcal{S}'})$ will be used as shorthands throughout this section. Then, exploiting the fact that $1 - F_{\chi_1^2}(\bullet)$ is a monotonically decreasing transformation, the following reformulation of the problem can be obtained:

$$\begin{aligned} \tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}}) &= \min_{\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}(\mathbf{r}_{\mathcal{S}})} p_{\min}(\mathbf{r}_{\mathcal{S}'}) & (5.10) \\ &= \min_{\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}(\mathbf{r}_{\mathcal{S}})} 1 - F_{\chi_1^2}(\max(T_{\max}^l(\mathbf{r}_{\mathcal{S}'}), T_{\max}^u(\mathbf{r}_{\mathcal{S}'}))) \\ &= 1 - F_{\chi_1^2}(\max_{\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}(\mathbf{r}_{\mathcal{S}})} \max(T_{\max}^l(\mathbf{r}_{\mathcal{S}'}), T_{\max}^u(\mathbf{r}_{\mathcal{S}'}))) \\ &= 1 - F_{\chi_1^2}(\max(\max_{\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}(\mathbf{r}_{\mathcal{S}})} T_{\max}^l(\mathbf{r}_{\mathcal{S}'}), \max_{\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}(\mathbf{r}_{\mathcal{S}})} T_{\max}^u(\mathbf{r}_{\mathcal{S}'}))). \end{aligned}$$

As a consequence, if an efficient algorithm to obtain

$$\tilde{T}_{\max}^l(\mathbf{r}_{\mathcal{S}}) = \max_{\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}(\mathbf{r}_{\mathcal{S}})} T_{\max}^l(\mathbf{r}_{\mathcal{S}'}), \quad (5.11)$$

$$\tilde{T}_{\max}^u(\mathbf{r}_{\mathcal{S}}) = \max_{\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}(\mathbf{r}_{\mathcal{S}})} T_{\max}^u(\mathbf{r}_{\mathcal{S}'}), \quad (5.12)$$

1. Deriving a more general procedure to efficiently evaluate $\tilde{p}_{\mathcal{S},\min}$ regardless of the test statistic of choice is still an open problem.

was available, the lower envelope of the minimum attainable P-value for the CMH test could be simply evaluated as:

$$\tilde{p}_{\min}(\mathbf{r}_S) = 1 - F_{\chi_1^2} \left(\max \left(\tilde{T}_{\max}^l(\mathbf{r}_S), \tilde{T}_{\max}^u(\mathbf{r}_S) \right) \right). \quad (5.13)$$

The FACS algorithm adheres to this approach, exploiting specific properties of the functions $T_{\max}^l(\mathbf{r}_{S'})$ and $T_{\max}^u(\mathbf{r}_{S'})$ for the CMH test in order to compute $\tilde{T}_{\max}^l(\mathbf{r}_S)$ and $\tilde{T}_{\max}^u(\mathbf{r}_S)$, thus also $\tilde{p}_{\min}(\mathbf{r}_S)$, in only $O(k \log k)$ time. These properties are highlighted in the following proposition.

Proposition 5.4. *Let \mathbf{r}_S be an arbitrary k -dimensional vector satisfying*

$$r_{S,c} \in \llbracket 0, \min(n_{1,c}, n_c - n_{1,c}) \rrbracket$$

for all $c = 1, 2, \dots, k$. Then, for all $\mathbf{r}_{S'} \in \mathcal{B}(\mathbf{r}_S)$, the functions $T_{\max}^l(\mathbf{r}_{S'})$ and $T_{\max}^u(\mathbf{r}_{S'})$ for the CMH test satisfy the following three properties:

(i) They can be expressed in closed-form as:

$$T_{\max}^l(\mathbf{r}_{S'}) = \frac{\left(\sum_{c=1}^k \gamma_c^l r_{S',c} \right)^2}{\sum_{c=1}^k \frac{n_c}{n_c-1} \gamma_c^l \gamma_c^u r_{S',c} \left(1 - \frac{r_{S',c}}{n_c} \right)}, \quad (5.14)$$

$$T_{\max}^u(\mathbf{r}_{S'}) = \frac{\left(\sum_{c=1}^k \gamma_c^u r_{S',c} \right)^2}{\sum_{c=1}^k \frac{n_c}{n_c-1} \gamma_c^l \gamma_c^u r_{S',c} \left(1 - \frac{r_{S',c}}{n_c} \right)}, \quad (5.15)$$

where $\gamma_c^l = \frac{n_{1,c}}{n_c}$ and $\gamma_c^u = 1 - \frac{n_{1,c}}{n_c}$ for each $c = 1, 2, \dots, k$.

(ii) If their domain is relaxed from $\mathcal{B}(\mathbf{r}_S) = \llbracket 0, r_{S,1} \rrbracket \times \dots \times \llbracket 0, r_{S,k} \rrbracket \subset \mathbb{N}^k$ to $\tilde{\mathcal{B}}(\mathbf{r}_S) = [0, r_{S,1}] \times \dots \times [0, r_{S,k}] \subset \mathbb{R}_+^k$, they are convex in $\mathbf{r}_{S'}$.

(iii) Their maximum value in $\mathcal{B}(\mathbf{r}_S)$ is attained at a vertex, that is,

$$\mathbf{r}_{S'}^{l,*} = \operatorname{argmax}_{\mathbf{r}_{S'} \in \mathcal{B}(\mathbf{r}_S)} T_{\max}^l(\mathbf{r}_{S'}), \quad (5.16)$$

$$\mathbf{r}_{S'}^{u,*} = \operatorname{argmax}_{\mathbf{r}_{S'} \in \mathcal{B}(\mathbf{r}_S)} T_{\max}^u(\mathbf{r}_{S'}), \quad (5.17)$$

satisfy $r_{S',c}^{l,*} \in \{0, r_{S,c}\}$ and $r_{S',c}^{u,*} \in \{0, r_{S,c}\}$ for each $c = 1, 2, \dots, k$.

Proof. By assumption, $r_{S',c} \leq r_{S,c} \leq \min(n_{1,c}, n_c - n_{1,c})$, hence $a_{S',c,\min} = \max(0, r_{S',c} - (n_c - n_{1,c})) = 0$ and $a_{S',c,\max} = \min(n_{1,c}, r_{S',c}) = r_{S',c}$ for each $c = 1, 2, \dots, k$. Therefore, according to Proposition 5.1, we have that:

$$T_{\max}^l(\mathbf{r}_{S'}) = \frac{\left(\sum_{c=1}^k r_{S',c} \frac{n_{1,c}}{n_c} \right)^2}{\sum_{c=1}^k \frac{r_{S,c}}{n_c} \frac{n_c - r_{S,c}}{n_c} \frac{n_c - n_{1,c}}{n_c - 1} n_{1,c}}, \quad (5.18)$$

$$T_{\max}^u(\mathbf{r}_{S'}) = \frac{\left(\sum_{c=1}^k r_{S',c} \left(1 - \frac{n_{1,c}}{n_c} \right) \right)^2}{\sum_{c=1}^k \frac{r_{S,c}}{n_c} \frac{n_c - r_{S,c}}{n_c} \frac{n_c - n_{1,c}}{n_c - 1} n_{1,c}}, \quad (5.19)$$

leading to Equations (5.14) and (5.15).

In order to prove property (ii), it will be shown that a function of the form:

$$T(\mathbf{r}_{S'}) = \frac{\left(\sum_{c=1}^k \gamma_c r_{S',c}\right)^2}{\sum_{c=1}^k \frac{n_c}{n_c-1} \gamma_c (1-\gamma_c) r_{S',c} \left(1 - \frac{r_{S',c}}{n_c}\right)},$$

with domain $\tilde{\mathcal{B}}(\mathbf{r}_S) = [0, r_{S,1}] \times \cdots \times [0, r_{S,k}] \subset \mathbb{R}_+^k$ is convex in $\mathbf{r}_{S'}$ provided that $\gamma_c \in [0, 1]$ for each $c = 1, 2, \dots, k$; an assumption which is satisfied by $T_{\max}^l(\mathbf{r}_{S'})$ and $T_{\max}^u(\mathbf{r}_{S'})$. Let $D(\mathbf{r}_{S'}) := \sum_{c=1}^k \frac{n_c}{n_c-1} \gamma_c (1-\gamma_c) r_{S',c} \left(1 - \frac{r_{S',c}}{n_c}\right)$, $F(\mathbf{r}_{S'}) := \sum_{c=1}^k \gamma_c r_{S',c}$ and $N(\mathbf{r}_{S'}) := F^2(\mathbf{r}_{S'})$. By definition, $T(\mathbf{r}_{S'})$ is convex if $T(\lambda \mathbf{r}_{S'}^{(1)} + (1-\lambda) \mathbf{r}_{S'}^{(2)}) \leq \lambda T(\mathbf{r}_{S'}^{(1)}) + (1-\lambda) T(\mathbf{r}_{S'}^{(2)})$ for any $\mathbf{r}_{S'}^{(1)}, \mathbf{r}_{S'}^{(2)} \in \tilde{\mathcal{B}}(\mathbf{r}_S)$ and $\lambda \in [0, 1]$. Since $T(\mathbf{r}_{S'}) = N(\mathbf{r}_{S'})/D(\mathbf{r}_{S'})$, this condition is equivalent to:

$$\lambda \frac{N(\mathbf{r}_{S'}^{(1)})}{D(\mathbf{r}_{S'}^{(1)})} + (1-\lambda) \frac{N(\mathbf{r}_{S'}^{(2)})}{D(\mathbf{r}_{S'}^{(2)})} \geq \frac{N(\lambda \mathbf{r}_{S'}^{(1)} + (1-\lambda) \mathbf{r}_{S'}^{(2)})}{D(\lambda \mathbf{r}_{S'}^{(1)} + (1-\lambda) \mathbf{r}_{S'}^{(2)})}. \quad (5.20)$$

It is straightforward to show after some algebraic manipulations that:

$$\begin{aligned} N(\lambda \mathbf{r}_{S'}^{(1)} + (1-\lambda) \mathbf{r}_{S'}^{(2)}) &= \lambda N(\mathbf{r}_{S'}^{(1)}) + (1-\lambda) N(\mathbf{r}_{S'}^{(2)}) \\ &\quad - \lambda(1-\lambda) (F(\mathbf{r}_{S'}^{(1)}) - F(\mathbf{r}_{S'}^{(2)}))^2, \end{aligned} \quad (5.21)$$

and

$$\begin{aligned} D(\lambda \mathbf{r}_{S'}^{(1)} + (1-\lambda) \mathbf{r}_{S'}^{(2)}) &= \lambda D(\mathbf{r}_{S'}^{(1)}) + (1-\lambda) D(\mathbf{r}_{S'}^{(2)}) \\ &\quad + \lambda(1-\lambda) \sum_{c=1}^k \frac{\gamma_c (1-\gamma_c)}{n_c-1} (\beta_c^{(2)} - \beta_c^{(1)})^2, \end{aligned} \quad (5.22)$$

hold in general. In particular, this implies that: (i) $N(\lambda \mathbf{r}_{S'}^{(1)} + (1-\lambda) \mathbf{r}_{S'}^{(2)}) \leq \lambda N(\mathbf{r}_{S'}^{(1)}) + (1-\lambda) N(\mathbf{r}_{S'}^{(2)})$, i.e. $N(\mathbf{r}_{S'})$ is convex in $\mathbf{r}_{S'}$, and (ii) $D(\lambda \mathbf{r}_{S'}^{(1)} + (1-\lambda) \mathbf{r}_{S'}^{(2)}) \geq \lambda D(\mathbf{r}_{S'}^{(1)}) + (1-\lambda) D(\mathbf{r}_{S'}^{(2)})$, i.e. $D(\mathbf{r}_{S'})$ is concave in $\mathbf{r}_{S'}$. For the sake of readability, define: (a) $N_1 := N(\mathbf{r}_{S'}^{(1)})$, $N_2 := N(\mathbf{r}_{S'}^{(2)})$, $N_{12} := N(\lambda \mathbf{r}_{S'}^{(1)} + (1-\lambda) \mathbf{r}_{S'}^{(2)})$; (b) $D_1 := D(\mathbf{r}_{S'}^{(1)})$, $D_2 := D(\mathbf{r}_{S'}^{(2)})$, $D_{12} := D(\lambda \mathbf{r}_{S'}^{(1)} + (1-\lambda) \mathbf{r}_{S'}^{(2)})$; and (c) $F_1 := F(\mathbf{r}_{S'}^{(1)})$, $F_2 := F(\mathbf{r}_{S'}^{(2)})$. As $\gamma_c \in [0, 1]$ for all $c = 1, 2, \dots, k$, $D(\mathbf{r}_{S'})$ is positive for any $\mathbf{r}_{S'} \in \tilde{\mathcal{B}}(\mathbf{r}_S)$. Moreover, if $D(\mathbf{r}_{S'}^{(0)}) = 0$ for some $\mathbf{r}_{S'}^{(0)} \in \tilde{\mathcal{B}}(\mathbf{r}_S)$, it follows that $F(\mathbf{r}_{S'}^{(0)}) = 0$, $N(\mathbf{r}_{S'}^{(0)}) = 0$ and $T(\mathbf{r}_{S'}^{(0)}) = 0$ too. As a consequence, if D_1, D_2 or D_{12} are zero, Equation (5.20) is trivially true. Let us consider instead the non-trivial case for which $D_1 > 0$, $D_2 > 0$ and $D_{12} > 0$. Then, multiplying the left-hand and right-hand sides of Equation (5.20) by $D_1 D_2 D_{12}$ leads to the following alternative characterisation of convexity:

$$\underbrace{(\lambda N_1 D_2 + (1-\lambda) N_2 D_1) D_{12}}_L \geq \underbrace{N_{12} D_1 D_2}_R. \quad (5.23)$$

Under the aforementioned assumptions, it is easy to show that this condition indeed holds:

$$\begin{aligned}
L &\geq (\lambda N_1 D_2 + (1 - \lambda) N_2 D_1) (\lambda D_1 + (1 - \lambda) D_2) \\
&= (\lambda^2 N_1 + (1 - \lambda)^2 N_2) D_1 D_2 + \lambda(1 - \lambda) (N_1 D_2^2 + N_2 D_1^2) \\
&= (\lambda N_1 + (1 - \lambda) N_2) D_1 D_2 + \lambda(1 - \lambda) (N_1 D_2^2 + N_2 D_1^2 - (N_1 + N_2) D_1 D_2) \\
&= (\lambda N_1 + (1 - \lambda) N_2 - \lambda(1 - \lambda) (F_1 - F_2)^2) D_1 D_2 + \lambda(1 - \lambda) (F_1 D_2 - F_2 D_1)^2 \\
&= N_{12} D_1 D_2 + \lambda(1 - \lambda) (F_1 D_2 - F_2 D_1)^2 \\
&\geq N_{12} D_1 D_2 = R,
\end{aligned} \tag{5.24}$$

where the first step follows from the fact that $D(\mathbf{r}_{S'})$ is concave, i.e. $D_{12} \geq \lambda D_1 + (1 - \lambda) D_2$, as shown in Equation (5.22), and the penultimate step results from applying Equation (5.21). This establishes the convexity of $T_{\max}^l(\mathbf{r}_{S'})$ and $T_{\max}^u(\mathbf{r}_{S'})$ in $\tilde{\mathcal{B}}(\mathbf{r}_S)$.

Finally, it will be shown that any function $T(\mathbf{r}_{S'})$ which is convex in $\tilde{\mathcal{B}}(\mathbf{r}_S)$ must attain its maximum value in $\tilde{\mathcal{B}}(\mathbf{r}_S)$ at a vertex. Since $\mathcal{B}(\mathbf{r}_S) \subset \tilde{\mathcal{B}}(\mathbf{r}_S)$ and all vertices of $\tilde{\mathcal{B}}(\mathbf{r}_S)$ are also contained in $\mathcal{B}(\mathbf{r}_S)$, this would imply that the maximum value of $T(\mathbf{r}_{S'})$ in $\mathcal{B}(\mathbf{r}_S)$ must also be attained at a vertex, thus proving property (iii). Indeed, suppose that $\tilde{\mathbf{r}}_{S'}^* = \operatorname{argmax}_{\mathbf{r}_{S'} \in \tilde{\mathcal{B}}(\mathbf{r}_S)} T(\mathbf{r}_{S'})$ was not a vertex of $\tilde{\mathcal{B}}(\mathbf{r}_S)$. Then, there would exist

some $c' \in \{1, 2, \dots, k\}$ such that $0 < \tilde{r}_{S',c'}^* < r_{S,c'}$. Let $\mathbf{r}_{S'}^{(1)}, \mathbf{r}_{S'}^{(2)} \in \tilde{\mathcal{B}}(\mathbf{r}_S)$ be defined as: (i) $r_{S',c}^{(1)} = r_{S',c}^{(2)} = \tilde{r}_{S',c}^*$ for all $c \neq c'$, and (ii) $r_{S',c'}^{(1)} = 0, r_{S',c'}^{(2)} = r_{S,c'}$. It can be readily verified that $\tilde{\mathbf{r}}_{S'}^* = \lambda \mathbf{r}_{S'}^{(1)} + (1 - \lambda) \mathbf{r}_{S'}^{(2)}$ if $\lambda = 1 - \frac{r_{S',c'}^*}{r_{S,c'}}$. As $0 < \tilde{r}_{S',c'}^* < r_{S,c'}$ by assumption, it follows that $\lambda \in (0, 1)$ and, as a consequence of $T(\mathbf{r}_{S'})$ being convex, we have that $T(\tilde{\mathbf{r}}_{S'}^*) \leq \lambda T(\mathbf{r}_{S'}^{(1)}) + (1 - \lambda) T(\mathbf{r}_{S'}^{(2)}) \leq \max(T(\mathbf{r}_{S'}^{(1)}), T(\mathbf{r}_{S'}^{(2)}))$. In other words, either $\mathbf{r}_{S'}^{(1)}$ or $\mathbf{r}_{S'}^{(2)}$ must be a maximiser of $T(\mathbf{r}_{S'})$ as well. Inductively applying this argument to all $c' = 1, 2, \dots, k$ shows that there exists $\mathbf{r}_{S'}^* \in \mathcal{B}(\mathbf{r}_S) \subset \tilde{\mathcal{B}}(\mathbf{r}_S)$ with $r_{S',c}^* \in \{0, r_{S,c}\}$ for each $c = 1, 2, \dots, k$ such that $T(\mathbf{r}_{S'}^*) \geq T(\tilde{\mathbf{r}}_{S'}^*)$, thus proving the claim. \square

Proposition 5.4 has direct algorithmic implications. As $T_{\max}^l(\mathbf{r}_{S'})$ and $T_{\max}^u(\mathbf{r}_{S'})$ must attain their maximum values at a vertex of $\mathcal{B}(\mathbf{r}_S)$, the combinatorial optimisation problems defined by Equations (5.11) and (5.12) can be solved with 2^k evaluations of $T_{\max}^l(\mathbf{r}_{S'})$ and $T_{\max}^u(\mathbf{r}_{S'})$, respectively. Consequently, Proposition 5.4 provides an approach to compute $\tilde{p}_{\min}(\mathbf{r}_S)$ with complexity $O(2^k)$ instead of $O(n^k)$. Nevertheless, while this improvement might be sufficient to result in a computationally feasible algorithm if the number k of categories for the covariate is small, e.g. $k = 2$ or $k = 3$, the computational overhead incurred by evaluating $\tilde{p}_{\min}(\mathbf{r}_S)$ still grows exponentially on k , severely hindering the applicability of the pruning condition described in Proposition 5.3. The FACS algorithm is built around a vastly more powerful result, which allows to further reduce the number of evaluations of $T_{\max}^l(\mathbf{r}_{S'})$ and $T_{\max}^u(\mathbf{r}_{S'})$ needed to compute $\tilde{p}_{\min}(\mathbf{r}_S)$ from 2^{k+1} to only $2k$, as described in the following proposition.

Proposition 5.5. *Let \mathbf{r}_S be an arbitrary k -dimensional vector satisfying*

$$r_{S,c} \in \llbracket 0, \min(n_{1,c}, n_c - n_{1,c}) \rrbracket$$

for all $c = 1, 2, \dots, k$ and consider a function

$$T(\mathbf{r}_{S'}) = \frac{\left(\sum_{c=1}^k \gamma_c r_{S',c}\right)^2}{\sum_{c=1}^k \frac{n_c}{n_c-1} \gamma_c (1 - \gamma_c) r_{S',c} \left(1 - \frac{r_{S',c}}{n_c}\right)}, \quad (5.25)$$

with $\gamma_c \in [0, 1]$ for all $c = 1, 2, \dots, k$ and domain $\mathcal{B}(\mathbf{r}_S)$. Furthermore, define

$$\beta_c = \frac{n_c}{n_c - 1} (1 - \gamma_c) \left(1 - \frac{r_{S,c}}{n_c}\right) \quad (5.26)$$

for each $c = 1, 2, \dots, k$ and let $\pi : \llbracket 1, k \rrbracket \rightarrow \llbracket 1, k \rrbracket$ be a permutation of $\llbracket 1, k \rrbracket$ that sorts $\{\beta_c\}_{c=1}^k$ in ascending order, that is, π satisfies $\beta_{\pi(1)} \leq \beta_{\pi(2)} \leq \dots \leq \beta_{\pi(k)}$. Then:

$$T^* = \max_{\mathbf{r}_{S'} \in \mathcal{B}(\mathbf{r}_S)} T(\mathbf{r}_{S'}) = \max_{\kappa \in \llbracket 1, k \rrbracket} O_\kappa, \quad (5.27)$$

where:

$$O_\kappa = \frac{\left(\sum_{c=1}^{\kappa} \gamma_{\pi(c)} r_{S,\pi(c)}\right)^2}{\sum_{c=1}^{\kappa} \frac{n_{\pi(c)}}{n_{\pi(c)}-1} \gamma_{\pi(c)} (1 - \gamma_{\pi(c)}) r_{S,\pi(c)} \left(1 - \frac{r_{S,\pi(c)}}{n_{\pi(c)}}\right)}. \quad (5.28)$$

In particular, this implies that T^* can be obtained with $O(k \log k)$ complexity.

Proof. As shown in the proof of Proposition 5.4, $T(\mathbf{r}_{S'})$ must attain its maximum value in $\mathcal{B}(\mathbf{r}_S)$ at a vertex of $\mathcal{B}(\mathbf{r}_S)$. Consequently:

$$T^* = \max_{\mathbf{r}_{S'} \in \mathcal{B}(\mathbf{r}_S)} T(\mathbf{r}_{S'}) = \max_{\mathbf{q} \in \{0,1\}^k} \frac{\left(\sum_{c=1}^k q_c f_c\right)^2}{\sum_{c=1}^k q_c \beta_c f_c}, \quad (5.29)$$

where $f_c = \gamma_c r_{S,c}$ and $\beta_c = \frac{n_c}{n_c-1} (1 - \gamma_c) \left(1 - \frac{r_{S,c}}{n_c}\right)$ as in Equation (5.26). Let $\pi : \llbracket 1, k \rrbracket \rightarrow \llbracket 1, k \rrbracket$ be a permutation satisfying the assumptions of the proposition, i.e. a permutation such that $\beta_{\pi(1)} \leq \beta_{\pi(2)} \leq \dots \leq \beta_{\pi(k)}$ and define:

$$T_{\pi,j}(\mathbf{q}) = \frac{\left(\sum_{c=1}^j q_{\pi(c)} f_{\pi(c)}\right)^2}{\sum_{c=1}^j q_{\pi(c)} \beta_{\pi(c)} f_{\pi(c)}}, \quad (5.30)$$

for each $j = 1, 2, \dots, k$. Since $T(\mathbf{r}_{S'})$ is invariant to permutations of $\mathbf{r}_{S'}$, it follows that $T_{\pi,k}^* = \max_{\mathbf{q} \in \{0,1\}^k} T_{\pi,k}(\mathbf{q}) = T^*$. In order to prove that $T^* = \max_{\kappa \in \llbracket 1, k \rrbracket} O_\kappa$, with O_κ as defined by Equation (5.28), a strategy reminiscent of dynamic programming will be used. Consider:

$$S_{\pi,j}(\mathbf{q}) = \frac{\left(f_{\pi(j+1)} + \sum_{c=1}^j q_{\pi(c)} f_{\pi(c)}\right)^2}{\beta_{\pi(j+1)} f_{\pi(j+1)} + \sum_{c=1}^j q_{\pi(c)} \beta_{\pi(c)} f_{\pi(c)}}, \quad (5.31)$$

so that:

$$T_{\pi,j}^* = \max_{\mathbf{q} \in \{0,1\}^k} T_{\pi,j}(\mathbf{q}) = \max\left(T_{\pi,j-1}^*, S_{\pi,j-1}^*\right), \quad (5.32)$$

where $S_{\pi,j-1}^* = \max_{\mathbf{q} \in \{0,1\}^k} S_{\pi,j-1}(\mathbf{q})$. Next, it will be shown by induction that $T_{\pi,j}^* = \max_{\kappa \in \llbracket 1,j \rrbracket} O_\kappa$ for all $j = 1, 2, \dots, k$, thus proving the proposition.

The base case $T_{\pi,1}^* = \max(0, O_1) = O_1$ trivially holds. Moreover, supposing that the inductive hypothesis $T_{\pi,j-1}^* = \max_{\kappa \in \llbracket 1,j-1 \rrbracket} O_\kappa$ is satisfied, it follows that:

$$T_{\pi,j}^* = \max \left(T_{\pi,j-1}^*, S_{\pi,j-1}^* \right) = \max \left(\max_{\kappa \in \llbracket 1,j-1 \rrbracket} O_\kappa, S_{\pi,j-1}^* \right). \quad (5.33)$$

The two possible cases (i) $S_{\pi,j-1}^* \leq T_{\pi,j-1}^*$ and (ii) $S_{\pi,j-1}^* > T_{\pi,j-1}^*$ will be considered separately.

- (i) If $S_{\pi,j-1}^* \leq T_{\pi,j-1}^*$ holds, then $T_{\pi,j}^* = T_{\pi,j-1}^* = \max_{\kappa \in \llbracket 1,j-1 \rrbracket} O_\kappa$, where the last step follows by the induction hypothesis. Furthermore, as $S_{\pi,j-1}^* \geq O_j$ is true in general, this would also imply that $T_{\pi,j-1}^* = \max_{\kappa \in \llbracket 1,j-1 \rrbracket} O_\kappa \geq S_{\pi,j-1}^* \geq O_j$, leading to $T_{\pi,j}^* = \max_{\kappa \in \llbracket 1,j-1 \rrbracket} O_\kappa = \max_{\kappa \in \llbracket 1,j \rrbracket} O_\kappa$, which would prove the inductive step.
- (ii) Suppose instead that $S_{\pi,j-1}^* > T_{\pi,j-1}^*$ holds, implying that $T_{\pi,j}^* = S_{\pi,j-1}^*$. The final step of this proof will involve showing that under these circumstances, $S_{\pi,j-1}^* = O_j$, leading to $T_{\pi,j}^* = O_j = \max_{\kappa \in \llbracket 1,j \rrbracket} O_\kappa$ and thus proving the inductive step as well.

The claim will be demonstrated by contradiction. Suppose that $S_{\pi,j-1}^* \neq O_j$. Then, according to Equation (5.31), there exists a strict subset \mathcal{Z} of $\{1, 2, \dots, j-1\}$ such that:

$$S_{\pi,j-1}^* = \frac{\left(f_{\pi(j)} + \sum_{c \in \mathcal{Z}} f_{\pi(c)} \right)^2}{\beta_{\pi(j)} f_{\pi(j)} + \sum_{c \in \mathcal{Z}} \beta_{\pi(c)} f_{\pi(c)}}. \quad (5.34)$$

For the sake of readability, define $F := f_{\pi(j)} + \sum_{c \in \mathcal{Z}} f_{\pi(c)}$ and $D := \beta_{\pi(j)} f_{\pi(j)} + \sum_{c \in \mathcal{Z}} \beta_{\pi(c)} f_{\pi(c)}$, leading to $S_{\pi,j-1}^* = \frac{F^2}{D}$. Additionally, define $F_{\neg} := \sum_{c \in \mathcal{Z}_{\neg}} f_{\pi(c)}$ and $D_{\neg} := \sum_{c \in \mathcal{Z}_{\neg}} \beta_{\pi(c)} f_{\pi(c)}$, where $\mathcal{Z}_{\neg} = \{1, 2, \dots, j-1\} \setminus \mathcal{Z}$ is a non-empty subset of $\{1, 2, \dots, j-1\}$.

Since $S_{\pi,j-1}^* \geq O_j$ holds in general and, by assumption, $S_{\pi,j-1}^* \neq O_j$, it follows that $S_{\pi,j-1}^*$ must be strictly greater than O_j . Using the notation introduced above, this condition can be expressed as:

$$\begin{aligned} S_{\pi,j-1}^* > O_j &\Leftrightarrow \frac{F^2}{D} > \frac{(F + F_{\neg})^2}{D + D_{\neg}} \\ &\Leftrightarrow F^2 (D + D_{\neg}) > (F^2 + F_{\neg}^2 + 2FF_{\neg}) D \\ &\Leftrightarrow F^2 \frac{D_{\neg}}{F_{\neg}} > F_{\neg} D + 2FD \\ &\Rightarrow F^2 \beta_{\pi(j)} > F_{\neg} D + 2FD, \end{aligned} \quad (5.35)$$

where the second step is valid since D and $D + D_{\neg}$ are both strictly larger than zero and the last step follows from:

$$\frac{D_{\neg}}{F_{\neg}} = \sum_{c \in \mathcal{Z}_{\neg}} \beta_{\pi(c)} \frac{f_{\pi(c)}}{F_{\neg}} \leq \sum_{c \in \mathcal{Z}_{\neg}} \beta_{\pi(j)} \frac{f_{\pi(c)}}{F_{\neg}} = \beta_{\pi(j)}, \quad (5.36)$$

which is itself a consequence of the fact that $\beta_{\pi(j)} \geq \beta_{\pi(c)}$ for all $c = 1, 2, \dots, j-1$ due to the way π was defined.

Additionally, it was assumed that $S_{\pi, j-1}^* > T_{\pi, j-1}^*$ holds. As a direct implication of the definition of $T_{\pi, j}(\mathbf{q})$, shown in Equation (5.30), it follows that $T_{\pi, j-1}^* \geq \frac{(F - f_{\pi(j)})^2}{D - \beta_{\pi(j)} f_{\pi(j)}}$. This leads to the following condition:

$$\begin{aligned} S_{\pi, j-1}^* > T_{\pi, j-1}^* &\Rightarrow \frac{F^2}{D} > \frac{(F - f_{\pi(j)})^2}{D - \beta_{\pi(j)} f_{\pi(j)}} \\ &\Leftrightarrow F^2 (D - \beta_{\pi(j)} f_{\pi(j)}) > (F^2 + f_{\pi(j)}^2 - 2F f_{\pi(j)}) D \\ &\Leftrightarrow -F^2 \beta_{\pi(j)} > f_{\pi(j)} D - 2FD, \end{aligned} \quad (5.37)$$

where the second step relies on the fact that $D > 0$ and $D - \beta_{\pi(j)} f_{\pi(j)} = \sum_{c \in \mathcal{Z}} \beta_{\pi(c)} f_{\pi(c)} > 0$. In order to show that the latter holds, it must be proven that $\mathcal{Z} \neq \emptyset$, which follows from Equation (5.35). Indeed, if \mathcal{Z} were empty, F would be equal to $f_{\pi(j)}$ and D to $\beta_{\pi(j)} f_{\pi(j)}$, leading to $\beta_{\pi(j)} f_{\pi(j)} > F - \beta_{\pi(j)} + 2\beta_{\pi(j)} f_{\pi(j)}$, which constitutes a clear contradiction.

Finally, the conditions in Equations (5.35) and (5.37) can be combined into a single inequality, resulting in:

$$(F_{\neg} + f_{\pi(j)}) D < 0. \quad (5.38)$$

This condition is necessarily false. Hence, it follows that under the assumption that $S_{\pi, j-1}^* > T_{\pi, j-1}^*$, $S_{\pi, j-1}^* = O_j \Leftrightarrow \mathcal{Z} = \{1, 2, \dots, j-1\} \Leftrightarrow \mathcal{Z}_{\neg} = \emptyset$ must hold, ultimately implying that $T_{\pi, j}^* = \max_{\kappa \in [1, j]} O_{\kappa}$.

The final statement in the proposition refers to the computational complexity incurred by obtaining $T^* = \max_{\mathbf{r}_{S'} \in \mathcal{B}(\mathbf{r}_S)} T(\mathbf{r}_{S'})$. Since, as it was shown, $T^* = \max_{\kappa \in [1, k]} O_{\kappa}$ holds, in order to compute T^* it is necessary to:

- (i) Compute β_c for all $c = 1, 2, \dots, k$, with complexity $O(k)$.
- (ii) Find an appropriate permutation π by sorting $\{\beta_c\}_{c=1}^k$ in ascending order, with complexity $O(k \log k)$.
- (iii) Compute O_{κ} for all $\kappa = 1, 2, \dots, k$. Since $O_{\kappa} = \frac{F_{\kappa}^2}{D_{\kappa}}$ with $F_{\kappa} := \sum_{c=1}^{\kappa} f_{\pi(c)} = f_{\pi(\kappa)} + F_{\kappa-1}$ and $D_{\kappa} := \sum_{c=1}^{\kappa} \beta_{\pi(c)} f_{\pi(c)} = \beta_{\pi(\kappa)} f_{\pi(\kappa)} + D_{\kappa-1}$, it follows that this step can also be accomplished with complexity $O(k)$.
- (iv) Retrieve the maximum value of O_{κ} for all $\kappa = 1, 2, \dots, k$, with complexity $O(k)$.

Therefore, the overall computational effort is dominated by the sorting step (ii), resulting in $O(k \log k)$ complexity for the entire procedure. \square

Proposition 5.5 culminates the developments described in this section, providing a computationally efficient approach to assess the pruning condition introduced in Proposition 5.3. The resulting method is summarised in Algorithm 5.1, which describes the subroutine of FACS that would be executed each time the pruning condition for the CMH test must be evaluated in Line 16 of Algorithm 3.2.

Algorithm 5.1 pruning_condition_cmh

Input: Pattern \mathcal{S} , estimate of the corrected significance threshold $\hat{\delta}_{\text{tar}}$. For simplicity, the vectors \mathbf{r}_S , \mathbf{n}_1 and \mathbf{n} are assumed to be available within the scope of the function

Output: A Boolean indicating whether the search space pruning condition described in Proposition 5.3 applies to pattern \mathcal{S} or not

```

1: function pruning_condition_cmh( $\mathcal{S}, \hat{\delta}_{\text{tar}}$ )
2:   for  $c = 1, 2, \dots, k$  do
3:     if  $r_{S,c} > \min(n_{1,c}, n_c - n_{1,c})$  then
4:       Return false
5:    $\mathcal{I} \leftarrow \emptyset$   $\triangleright c \in \mathcal{I}$  if and only if  $r_{S,c} > 0$ 
6:   for  $c = 1, 2, \dots, k$  do
7:      $\gamma_c^l \leftarrow \frac{n_{1,c}}{n_c}, \gamma_c^u \leftarrow 1 - \gamma_c^l$   $\triangleright$  see Proposition 5.4
8:     if  $r_{S,c} > 0$  then
9:       Append  $c$  to  $\mathcal{I}$ 
10:     $\tilde{T}_{\max}^l(\mathbf{r}_S) \leftarrow \max_{\mathcal{I}}(\{\gamma_c^l\}_{c=1}^k, \mathcal{I}), \tilde{T}_{\max}^u(\mathbf{r}_S) \leftarrow \max_{\mathcal{I}}(\{\gamma_c^u\}_{c=1}^k, \mathcal{I})$ 
11:     $\tilde{p}_{\min}(\mathbf{r}_S) \leftarrow 1 - F_{\chi_1^2}(\max(\tilde{T}_{\max}^l(\mathbf{r}_S), \tilde{T}_{\max}^u(\mathbf{r}_S)))$ 
12:    if  $\tilde{p}_{\min}(\mathbf{r}_S) > \hat{\delta}_{\text{tar}}$  then
13:      Return true
14:    Return false
15:  end function
16:  procedure max_T( $\{\gamma_c\}_{c=1}^k, \mathcal{I}$ )
17:    for  $c \in \mathcal{I}$  do
18:       $f_c \leftarrow \gamma_c r_{S,c}, d_c \leftarrow \frac{n_c}{n_c - 1} \gamma_c (1 - \gamma_c) r_{S,c} \left(1 - \frac{r_{S,c}}{n_c}\right), \beta_c \leftarrow \frac{d_c}{f_c}$ 
19:    Sort  $\{\beta_c\}_{c \in \mathcal{I}}$  in ascending order to obtain  $\pi : [1, |\mathcal{I}|] \rightarrow \mathcal{I}$  such that  $\beta_{\pi(1)} \leq \beta_{\pi(2)} \leq \dots \leq \beta_{\pi(|\mathcal{I}|)}$ 
20:     $F_0 \leftarrow 0, D_0 \leftarrow 0$ 
21:    for  $\kappa = 1, \dots, |\mathcal{I}|$  do
22:       $F_\kappa \leftarrow F_{\kappa-1} + f_{\pi(\kappa)}, D_\kappa \leftarrow D_{\kappa-1} + \beta_{\pi(\kappa)} f_{\pi(\kappa)}, O_\kappa \leftarrow \frac{F_\kappa^2}{D_\kappa}$ 
23:    Return  $\max_{\kappa \in [1, |\mathcal{I}|]} O_\kappa$ 
24:  end procedure

```

The algorithm begins by verifying whether condition (ii) of Proposition 5.3 is satisfied by pattern \mathcal{S} , i.e. whether $r_{S,c} \leq \min(n_{1,c}, n_c - n_{1,c})$ holds for all $c = 1, 2, \dots, k$, as shown in Lines 2-4. If this was the case, the algorithm would proceed to verify condition (i) next, which requires computing the lower envelope of the minimum attainable P-value $\tilde{p}_{\min}(\mathbf{r}_S)$. As discussed in this section, this will be accomplished by maximising $T_{\max}^l(\mathbf{r}_{S'})$ and $T_{\max}^u(\mathbf{r}_{S'})$ in $\mathcal{B}(\mathbf{r}_S)$, resulting in $\tilde{T}_{\max}^l(\mathbf{r}_S)$ and $\tilde{T}_{\max}^u(\mathbf{r}_S)$, respectively. As shown in Proposition 5.4, the functions $T_{\max}^l(\mathbf{r}_{S'})$ and $T_{\max}^u(\mathbf{r}_{S'})$ have the same functional form, differing only in a set of k parameters $\{\gamma_c\}_{c=1}^k$. The corresponding values of $\{\gamma_c^l\}_{c=1}^k$ and $\{\gamma_c^u\}_{c=1}^k$ are computed in Lines 6-9. Moreover, categories for which $r_{S,c} = 0$ can be ignored without affecting $T_{\max}^l(\mathbf{r}_{S'})$ and $T_{\max}^u(\mathbf{r}_{S'})$, thus allowing to reduce the effective number of categories and, in turn,

making the algorithm more computationally efficient. For this reason, the same block of pseudocode also keeps track of the subset \mathcal{I} of categories for which $r_{S,c} \neq 0$. Next, in Line 10, the subroutine `max_T` is invoked to perform the required maximisations. Its pseudocode, described in Lines 16-23, is a self-explanatory implementation of the procedure described in Proposition 5.5, with the sole difference that categories for which $r_{S,c} = 0$, i.e. categories for which $c \notin \mathcal{I}$, are effectively ignored. Once $\tilde{T}_{\max}^l(\mathbf{r}_S)$ and $\tilde{T}_{\max}^u(\mathbf{r}_S)$ have been obtained, the lower envelope of the minimum attainable P-value $\tilde{p}_{\min}(\mathbf{r}_S)$ is evaluated next in Line 11. Finally, this allows to verify condition (i) of Proposition 5.3 in Line 12 of the algorithm, thus completing the assessment of the pruning condition for pattern S .

5.5 MISCELLANEOUS ASPECTS OF THE FACS ALGORITHM

This section will cover a variety of issues regarding our proposed approach, the FACS algorithm. In particular, following the example of Chapters 3 and 4, the most relevant low-level implementation choices that need to be considered will be discussed first. Next, we will explore the possibility of combining the developments of Chapter 4, which allow to make an efficient use of permutation testing in significant pattern mining, with the techniques proposed in this chapter. Finally, we will refer the reader to an alternative algorithm to account for covariates in significant pattern mining which was developed in parallel to FACS.

Implementation considerations

From an algorithmic point of view, the structure of FACS closely resembles that of LAMP 2.0. As a consequence, most aspects discussed in Section 3.5 in the context of Algorithm 3.2 largely apply to FACS as well. Nevertheless, some subtle differences which arise as a result of the inclusion of covariates must be brought into consideration:

- (i) In previous chapters, we emphasised the practical importance of choosing an appropriate pattern mining algorithm to efficiently generate and traverse the pattern enumeration tree. While LAMP 2.0 and `Westfall-Young light` can readily make use of virtually any frequent pattern mining algorithm “out-of-the-box”, FACS requires the ability to compute the support of each enumerated pattern in k disjoint subsets of the original input dataset \mathcal{D} , that is, it needs to compute $r_{S,c}$ for each $c = 1, 2, \dots, k$ instead of merely computing $r_S = \sum_{c=1}^k r_{S,c}$. While this is by no means an insurmountable obstacle, as most popular pattern mining algorithms can be adapted accordingly, it does imply that implementing the pattern enumeration routines for FACS might be slightly more involved than implementing them for LAMP 2.0 or `Westfall-Young light`.
- (ii) Section 3.5 described a strategy to exploit the fact that the minimum attainable P-value function $p_{\min}(r_S)$ for Pearson’s χ^2 test and Fisher’s exact test can only take $\lfloor \frac{n}{2} \rfloor$ distinct values, leading to an optimal adjustment of the estimate $\hat{\delta}_{\text{tar}}$ of the corrected significance threshold each time Line 13 of Algorithm 3.2 is executed. While, in principle, the same idea could also be applied to FACS, the number of distinct values $p_{\min}(\mathbf{r}_S)$ can take grows as $O(n^k)$ instead of $O(n)$. Consequently,

this strategy might become impractical for sufficiently large n and k , making the simpler alternative based on grid search a more appealing implementation choice for FACS. Analogously, precomputing $p_{\min}(\mathbf{r}_S)$ as a look-up table might be undesirable for very large datasets or covariates with many categories.

- (iii) As LAMP 2.0 and Westfall-Young light, the FACS algorithm relies on the discrete nature of the data. More precisely, FACS is based on the CMH test, implying that it is only able to correct for a single categorical covariate. Nevertheless, since FACS can handle a large number k of categories for the covariate, it is possible to correct for multiple categorical covariates C_1, C_2, \dots, C_d by defining a new covariate C with $k = \prod_{j=1}^d k_j$ categories. Extending FACS to account for real-valued covariates is, however, considerably more challenging. Despite this obvious limitation, our experiments suggest that a heuristic approach based on discretising continuous covariates can lead to satisfactory results. In particular, given a d -dimensional, real-valued covariate $\mathbf{c} \in \mathbb{R}^d$, vector quantisation can be used to map each observed covariate $\mathbf{c}_i \in \mathbb{R}^d$ in an input dataset \mathcal{D} to a single category $\tilde{c}_i \in \{1, 2, \dots, k\}$ out of a set of k codewords. For instance, as will be shown in Section 5.6 and Chapter 6, by simply applying the k-means algorithm [81] to $\{\mathbf{c}_i\}_{i=1}^n$ and defining \tilde{c}_i as the cluster assignment of \mathbf{c}_i , we were able to successfully correct for confounding due to population structure in genome-wide association studies.

Extensions to permutation testing-based significant pattern mining

Chapter 4 showed how permutation testing could be used as an alternative to Tarone’s method in significant pattern mining, effectively providing a practical way to considerably increase statistical power at the expense of computational efficiency. As evidenced throughout this chapter, the FACS algorithm was designed to make use of Tarone’s method, discussed in Section 2.3, as the underlying approach to correct for the multiple comparisons problem. Nevertheless, all derivations described in Sections 5.3 and 5.4 can be directly applied to the Westfall-Young light algorithm introduced in Chapter 4. In particular, a version of Westfall-Young light able to account for a categorical covariate can be obtained by incorporating the following modifications into Algorithm 4.2:

- (i) Since the null hypothesis H_0 is no longer $G_S(X) \perp\!\!\!\perp Y$ but rather $G_S(X) \perp\!\!\!\perp Y \mid C$, the permutation testing procedure described in Section 4.2 needs to be altered slightly. Given an input dataset $\mathcal{D} = \{(x_i, y_i, c_i)\}_{i=1}^n$, define $\mathcal{J}(c) = \{i \in \llbracket 1, n \rrbracket \mid c_i = c\}$ and $\mathcal{D}(c) = \{(x_i, y_i)\}_{i \in \mathcal{J}(c)}$. In order to obtain a resampled dataset $\tilde{\mathcal{D}}$ which obeys the global null hypothesis that $G_S(X) \perp\!\!\!\perp Y \mid C$ for all $S \in \mathcal{M}$, each stratified dataset $\mathcal{D}(c)$ can be first resampled separately, i.e. $\tilde{\mathcal{D}}(c) = \left\{ (x_i, y_{\pi_c(i)}) \right\}_{i \in \mathcal{J}(c)}$ with $\pi_c : \mathcal{J}(c) \rightarrow \mathcal{J}(c)$ being a random permutation of $\mathcal{J}(c)$, followed by the aggregation of all resampled datasets, $\tilde{\mathcal{D}} = \cup_{c=1}^k \tilde{\mathcal{D}}(c)$. Thus, Line 4 of the algorithm needs to be modified accordingly.
- (ii) Line 11 should compute the minimum attainable P-value for the CMH test as described in Section 5.3.

- (iii) Line 20 should implement the pruning condition for the CMH test introduced in Section 5.4 by means of Algorithm 5.1.

The use of the CMH test in place of Pearson’s χ^2 test or Fisher’s exact test will not alter the qualitative trade-offs incurred by the use of permutation testing in significant pattern mining; the approach resulting from modifying Algorithm 4.2 as indicated above will outperform FACS in terms of statistical power but will also be more computationally-demanding.

Related work

An alternative method to account for the effect of a categorical covariate in significant pattern mining was developed concurrently by [82]. The main difference with respect to FACS lies in the choice of test statistic. Rather than making use of the CMH test, the approach in [82] uses a less widely-known particularisation of logistic regression for binary samples [83]. By exploiting the discrete nature of the data, this formulation allows to obtain exact P-values for the logistic regression model, avoiding the need for large sample approximations. For additional details, we refer the reader to the original article [82].

5.6 EXPERIMENTS

In this section, our proposed approach, the FACS algorithm, will be evaluated in terms of computational efficiency, statistical power and ability to correct for confounding covariates by means of simulation experiments with synthetic data. Finally, a proof-of-principle application of FACS to analyse data from two genome-wide association studies in the plant model organism *A. thaliana* will be presented.

5.6.1 *Experimental setup*

While FACS is a general-purpose significant pattern mining algorithm, all experiments discussed in this section will focus exclusively on significant itemset mining. A simplified version of the Eclat algorithm [31, 84], based on an implementation written by the author of [70], was used as the underlying closed² itemset mining algorithm. FACS and all baseline algorithms were written from scratch in C++ and compiled using gcc 4.8.2 with -O3 optimisation. Each experiment was executed on a single thread of a 2.5 GHz Intel Xeon CPU with 64 GB of memory available.

2. A pattern \mathcal{S} is said to be *closed* if $r_{\mathcal{S}} > r_{\mathcal{S}'}$ for all $\mathcal{S}' \supset \mathcal{S}$, where $r_{\mathcal{S}} = \sum_{c=1}^k r_{\mathcal{S},c}$ is the support of \mathcal{S} in the input dataset \mathcal{D} . If a pattern \mathcal{S} is *not* closed, then there exists a closed pattern $\mathcal{S}' \supset \mathcal{S}$ such that $g_{\mathcal{S}}(x_i) = g_{\mathcal{S}'}(x_i)$ for all observations $\{x_i\}_{i=1}^n$ in \mathcal{D} . Thus, enumerating only closed patterns can be understood as a way to account for the most extreme instance of dependence due to subset/superset relationships between patterns, namely, those cases for which the corresponding test statistics would not merely be statistically dependent but would rather be identical.

5.6.2 *Simulation experiments*

The defining characteristic of FACS, which sets it apart from other significant pattern mining algorithms, is its ability to correct for covariates. This naturally leads to two fundamental questions, which will be explored in detail via simulation experiments:

- (i) *Does FACS achieve its goal of reducing the number of false discoveries due to confounding?*
- (ii) *Does incorporating the possibility to account for a categorical covariate affect the performance of the method negatively in other aspects such as statistical power or computational efficiency?*

BASELINES: In order to answer these questions, we will compare FACS with the following baseline algorithms:

LAMP- χ^2 : The LAMP 2.0 algorithm with Pearson’s χ^2 test as the underlying test statistic, which we will denote as LAMP- χ^2 throughout the remainder of this section, is arguably the comparison partner that can bring the most insight in our experiments. LAMP- χ^2 is virtually identical to FACS from both an algorithmic and a statistical perspective, with the sole key difference that FACS is designed to account for a categorical covariate whereas LAMP- χ^2 is not. Thus, comparing FACS to LAMP- χ^2 will allow us to obtain an unbiased, empirical answer to the two questions posed above.

BONF-CMH: With the aim of investigating the effectiveness of Tarone’s method when applied to conditional association testing, an alternative baseline algorithm, which we call Bonf-CMH, will be also introduced. Bonf-CMH consists of applying the CMH test, as described in Section 5.2, in conjunction with a naive Bonferroni correction. Thus, as FACS, the resulting approach will be able to account for a categorical covariate, presumably reducing the number of false discoveries due to confounding. Nevertheless, compared to FACS and LAMP- χ^2 , it should in principle lose the computational and statistical advantages that Tarone’s testability criterion provides.

m^k -FACS AND 2^k -FACS: In order to study the effectiveness of FACS in dealing with a large number k of categories for the covariate, two additional baseline algorithms will be included in the subset of simulation experiments devoted to assess runtime. The first of them, m^k -FACS, is a version of FACS that evaluates the pruning condition for the CMH test by solving the combinatorial optimisation problem in Equation (5.9) by brute force, i.e. by evaluating $p_{\min}(\mathbf{r}_{S'})$ for all $m^k = O(n^k)$ distinct $\mathbf{r}_{S'}$ in $\mathcal{B}(\mathbf{r}_S)$. Finally, 2^k -FACS can be considered an “intermediate” version of FACS that leverages our results in Proposition 5.4 to obtain a solution of Equation (5.9) by only computing $p_{\min}(\mathbf{r}_{S'})$ at the 2^k vertices of $\mathcal{B}(\mathbf{r}_S)$, thus being considerably more efficient than m^k -FACS yet still drastically slower than the full-fledged version of FACS, which solves Equation (5.9) in only $O(k \log k)$ time.

DATA GENERATION: All simulation experiments in this chapter use synthetic item-set mining datasets $\mathcal{D} = \{(x_i, y_i, c_i)\}_{i=1}^n$, consisting of n triplets (x_i, y_i, c_i) sampled

i.i.d. from a distribution $p(x, y, c)$. As described in Section 5.2, $x \in \mathcal{X}$ is an observation, which in itemset mining corresponds to a p -dimensional binary vector $x = (u_1, u_2, \dots, u_p) \in \{0, 1\}^p$, $y \in \{0, 1\}$ is a binary class label and $c \in \{1, 2, \dots, k\}$ a categorical covariate with k categories. The specific choice of data-generating distribution $p(x, y, c)$ will depend on the aim of the experiment.

Since the presence or absence of statistical associations between observations, labels and covariates plays a minor role in determining how computationally demanding mining significant patterns in a certain dataset will be, we use a fully-factorised data-generating distribution $p(x, y, c) = p(x)p(y)p(c)$ in all simulation experiments which aim to assess computational efficiency. In particular, we set $p(y) = \text{Bernoulli}(y \mid p_y)$, $p(c) = \text{Categorical}(c \mid \mathbf{p}_c)$ and, for the sake of simplicity, $p(x)$ is assumed to be a fully-factorised distribution with identically-distributed features, i.e. $p(x) = \prod_{j=1}^p p(u_j)$ with $p(u_j) = \text{Bernoulli}(u_j \mid p_u)$ for each $j = 1, 2, \dots, p$.

In contrast, the data-generating distribution $p(x, y, c)$ for simulation experiments intended to evaluate statistical power and the *false discovery rate (FDR)* is necessarily more complex, as it must incorporate statistical dependencies between x , y and c . For these experiments, our generative model first samples uniformly at random two feature interactions $\mathcal{S}_{\text{true}}, \mathcal{S}_{\text{conf}} \subseteq \{1, 2, \dots, p\}$ satisfying $|\mathcal{S}_{\text{true}}| = l_{\text{true}}$, $|\mathcal{S}_{\text{conf}}| = l_{\text{conf}}$ and $\mathcal{S}_{\text{true}} \cap \mathcal{S}_{\text{conf}} = \emptyset$. As the notation suggests, these two disjoint feature interactions will represent a truly associated pattern and a confounded pattern, respectively. In other words, the generative model will be designed to guarantee that the condition $G_{\mathcal{S}_{\text{true}}}(X) \not\perp\!\!\!\perp Y \mid C$ holds for $\mathcal{S}_{\text{true}}$ whereas the conditions $G_{\mathcal{S}_{\text{conf}}}(X) \not\perp\!\!\!\perp Y$ and $G_{\mathcal{S}_{\text{conf}}}(X) \perp\!\!\!\perp Y \mid C$ hold for $\mathcal{S}_{\text{conf}}$. To accomplish this, we model the joint distribution $p(g_{\mathcal{S}_{\text{true}}}(x), g_{\mathcal{S}_{\text{conf}}}(x), y, c)$ as

$$p(g_{\mathcal{S}_{\text{true}}}(x), g_{\mathcal{S}_{\text{conf}}}(x), y, c) = p(g_{\mathcal{S}_{\text{true}}}(x), y, c) p(g_{\mathcal{S}_{\text{conf}}}(x) \mid c),$$

where:

- (i) Under the simplifying assumption that the covariate c is binary, i.e. that $c \in \{0, 1\}$ or, equivalently, that $k = 2$, we make use of the approach described in [85] to sample from a multivariate Bernoulli distribution $p(g_{\mathcal{S}_{\text{true}}}(x), y, c)$ with specified first and second-order moments. In particular, we set:
 - (a) $\mathbb{E}(G_{\mathcal{S}_{\text{true}}}(X)) = p_{\text{true}}$, $\mathbb{E}(Y) = p_y$ and $\mathbb{E}(C) = p_c$.
 - (b) $\text{Corr}(Y, G_{\mathcal{S}_{\text{true}}}(X)) = \rho_{\text{true}}$ and $\text{Corr}(Y, C) = \rho_{\text{conf}}$ whereas $G_{\mathcal{S}_{\text{true}}}(X)$ and C are uncorrelated, that is, $\text{Corr}(G_{\mathcal{S}_{\text{true}}}(X), C) = 0$.
- (ii) The conditional distribution $p(g_{\mathcal{S}_{\text{conf}}}(x) \mid c)$ relating the occurrences of the confounded pattern $\mathcal{S}_{\text{conf}}$ in an observation x to the value taken by the confounding covariate c is defined as a binary symmetric channel [86] with a small error rate ϵ . That is, $G_{\mathcal{S}_{\text{conf}}}(X) = C$ with probability $1 - \epsilon$ and $G_{\mathcal{S}_{\text{conf}}}(X) = 1 - C$ with probability ϵ .

After having generated n i.i.d. samples $\{(g_{\mathcal{S}_{\text{true}}}(x_i), g_{\mathcal{S}_{\text{conf}}}(x_i), y_i, c_i)\}_{i=1}^n$ in this manner, the final step in our generative model is to obtain n i.i.d. observations $\{x_i\}_{i=1}^n$ which are consistent with the values of $\{g_{\mathcal{S}_{\text{true}}}(x_i)\}_{i=1}^n$ and $\{g_{\mathcal{S}_{\text{conf}}}(x_i)\}_{i=1}^n$. In order to do so, the way in which the p binary features u_1, u_2, \dots, u_p in x will be sampled depends on whether j is in $\mathcal{S}_{\text{true}} \cup \mathcal{S}_{\text{conf}}$ or not:

- (i) All “background” features $\{u_j \mid j \in \{1, 2, \dots, p\} \setminus (\mathcal{S}_{\text{true}} \cup \mathcal{S}_{\text{conf}})\}$ are obtained as i.i.d. draws from a Bernoulli distribution, i.e. $p(u_j) = \text{Bernoulli}(u_j \mid p_u)$.
- (ii) The features $\{u_j \mid j \in \mathcal{S}_{\text{true}}\}$ which form the truly associated pattern $\mathcal{S}_{\text{true}}$ must satisfy $u_{i,j} = 1$ for all samples $i = 1, 2, \dots, n$ for which $g_{\mathcal{S}_{\text{true}}}(x_i) = 1$, since $g_{\mathcal{S}_{\text{true}}}(x_i)$ is defined as $g_{\mathcal{S}_{\text{true}}}(x_i) = \prod_{j \in \mathcal{S}_{\text{true}}} u_{i,j}$. Similarly, for all samples $i = 1, 2, \dots, n$ for which $g_{\mathcal{S}_{\text{true}}}(x_i) = 0$, there must be at least one $j^*(i) \in \mathcal{S}_{\text{true}}$ such that $u_{i,j^*} = 0$. Thus, for each $i = 1, 2, \dots, n$ satisfying $g_{\mathcal{S}_{\text{true}}}(x_i) = 0$, we sample a feature index $j^*(i)$ uniformly at random from $\mathcal{S}_{\text{true}}$ and set $u_{i,j^*} = 0$. The remaining features $\{u_{i,j} \mid j \in \mathcal{S}_{\text{true}} \setminus \{j^*(i)\}\}$ have no effect on $g_{\mathcal{S}_{\text{true}}}(x_i)$ and can therefore take any value. We choose to set $u_{i,j} = 1$ for all $j \in \mathcal{S}_{\text{true}} \setminus \{j^*(i)\}$, as this minimises the univariate statistical association of each feature in $\mathcal{S}_{\text{true}}$ with the class labels, making the discovery of $\mathcal{S}_{\text{true}}$ more challenging.
- (iii) The features $\{u_j \mid j \in \mathcal{S}_{\text{conf}}\}$ which form the confounded pattern $\mathcal{S}_{\text{conf}}$ are sampled analogously as in (ii).

Throughout all simulation experiments, the truly associated and confounded patterns $\mathcal{S}_{\text{true}}$ and $\mathcal{S}_{\text{conf}}$ have the same number of interacting features, i.e. $l_{\text{true}} = l_{\text{conf}} = l$. Moreover, the correlations ρ_{true} and ρ_{conf} are also set to be identical, that is, $\rho_{\text{true}} = \rho_{\text{conf}} = \frac{\rho}{2}$. In this case, $\rho \in [0, 1]$ can be interpreted as the proportion of variance of the class labels which can be jointly explained by the truly associated pattern and the covariate, i.e. it is the overall signal strength in the dataset.

METRICS: Runtime will be used as the objective criterion to benchmark the computational efficiency of FACS against all baseline algorithms³. Absolute runtime values must always be interpreted with care, as they strongly depend on non-algorithmic factors such as the hardware of the system used to execute the algorithms, the programming language the algorithms were written in or the specific choice of pattern mining algorithm used to traverse the enumeration tree. However, all methods included in these simulation experiments have been executed on the same system, programmed in the same language and use our own implementation of Eclat to enumerate closed itemsets. Thus, we can reliably interpret relative differences in runtime to provide a fair assessment of the computational efficiency of the distinct significant pattern mining algorithms under consideration.

Evaluating statistical power and the FDR is, nevertheless, considerably more involved. Owing to the dependence between patterns in the search space, an aspect of significant pattern mining which was discussed extensively in Chapter 4, the null hypothesis $G_{\mathcal{S}}(X) \perp\!\!\!\perp Y \mid C$ will not only be violated by $\mathcal{S}_{\text{true}}$, contrary to what our data generation model might suggest. In practice, there might be many patterns \mathcal{S} which are sufficiently related to $\mathcal{S}_{\text{true}}$ to be statistically associated as well. Typical examples of such patterns include subsets and supersets of $\mathcal{S}_{\text{true}}$, as well as patterns \mathcal{S} for which the intersection between $\mathcal{S}_{\text{true}}$ and \mathcal{S} is large relative to the size of the patterns, i.e. patterns \mathcal{S} with a large Jaccard similarity with $\mathcal{S}_{\text{true}}$. As a consequence, defining which patterns should be considered as true positives and which patterns should be considered as false positives becomes non-trivial. In this chapter, we follow

³. For all practical purposes, the performance of FACS and all baseline algorithms is almost identical in terms of memory usage.

a pragmatic approach and introduce heuristic definitions that nonetheless allow us to compare the relative performance of the approaches under consideration. Let $\mathcal{M}_{\text{sig}} \subseteq \mathcal{M}$ denote the subset of patterns deemed significantly associated by a certain significant pattern mining algorithm. We will consider a pattern $\mathcal{S} \in \mathcal{M}_{\text{sig}}$ to be a true positive if $|\mathcal{S} \cap \mathcal{S}_{\text{true}}| > \frac{|\mathcal{S}|}{2}$, i.e. if strictly more than half of the features indexed by \mathcal{S} are part of the truly associated feature interaction $\mathcal{S}_{\text{true}}$. Similarly, $\mathcal{S} \in \mathcal{M}_{\text{sig}}$ will be considered a false positive if $|\mathcal{S} \cap \mathcal{S}_{\text{true}}| < \frac{|\mathcal{S}|}{2}$. If $|\mathcal{S} \cap \mathcal{S}_{\text{true}}|$ were equal to $\frac{|\mathcal{S}|}{2}$, we would increment the total number of true positives TP and the total number of false positives FP by 0.5 each. Using these heuristic definitions of true and false positives, the FDR will be obtained as usual, i.e.

$$\text{FDR} = \mathbb{E} \left(\frac{\text{FP}}{\max(\text{TP} + \text{FP}, 1)} \right).$$

Finally, in order to define a measure of statistical power for our simulation experiments, a more stringent notion of true positive will be adopted. In particular, we will define statistical power as the probability that $\mathcal{S}_{\text{true}}$ is deemed significantly associated, i.e. as the probability that $\mathcal{S}_{\text{true}}$ is part of \mathcal{M}_{sig} . While these heuristic definitions of FDR and statistical power provide substantial insight into the performance of different significant pattern mining algorithms, developing novel precision and recall metrics that account for the dependence between patterns in a principled manner remains a fundamental open problem in significant pattern mining.

FDR and statistical power

To investigate the ability of FACS to correct for a confounding covariate and the impact this has on its statistical power, we carried out a set of simulation experiments to evaluate the FDR and resulting statistical power of FACS and two baseline algorithms, Bonf-CMH and LAMP- χ^2 , as the overall signal strength ρ varies in the range $[0, 1]$. For each value of ρ , we estimated the FDR and the resulting statistical power by averaging across the results of 100 synthetic datasets, generated according to the model described above. Each dataset consisted of $n = 200$ samples, $p = 5,000$ features and $k = 2$ categories for the covariate. Both the truly significant and the confounded patterns contained $l = 5$ interacting features. For the sake of simplicity, both class labels and both categories for the covariate were equiprobable *a priori*, i.e. $p_y = 0.5$ and $p_c = 0.5$. The probability that the truly associated pattern occurs in an input sample x was also set to $p_{\text{true}} = 0.5$ while, as discussed in the previous section, the presence or absence of the confounded pattern in each input sample was determined by the value taken by categorical covariate, with $g_{\mathcal{S}_{\text{conf}}}(x_i) = c_i$ occurring with probability $1 - \epsilon = 0.95$ and $g_{\mathcal{S}_{\text{conf}}}(x_i) = 1 - c_i$ occurring with probability $\epsilon = 0.05$. Finally, the probability p_u of a “background” feature being active, which largely controls the sparsity of the resulting dataset, was set to 0.1.

Figure 5.5 shows the resulting FDR for all three methods under consideration. The most striking feature of these results is the confirmation that the two approaches which use the CMH test to correct for the categorical covariate, FACS and Bonf-CMH, are able to drastically reduce the FDR compared to LAMP- χ^2 . In particular, it can be seen that both FACS and Bonf-CMH control the FDR at level $\alpha = 0.05$, which is a direct

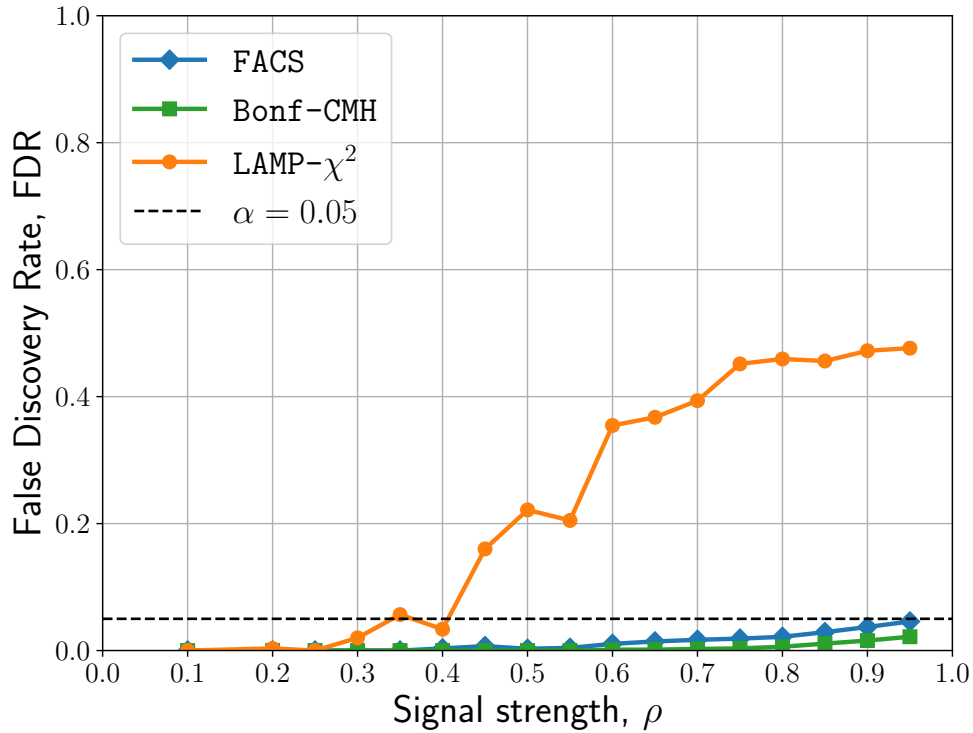


Figure 5.5. – FDR as a function of the signal strength ρ for our proposed approach FACS and two baseline algorithms: Bonf-CMH and LAMP- χ^2 .

consequence⁴ of the fact that both approaches strongly control the FWER at level $\alpha = 0.05$. In contrast, as the signal strength ρ approaches 1, half of all discoveries made by LAMP- χ^2 are false positives, in this particular case owing to confounding. Figure 5.6 depicts the resulting statistical power in these simulation experiments, providing a complementary view of the statistical performance of FACS, LAMP- χ^2 and Bonf-CMH. Firstly, it can be readily seen that FACS and LAMP- χ^2 , which use Tarone’s testability criterion to correct for multiple comparisons, vastly outperform Bonf-CMH, which uses a naive Bonferroni correction instead. This effect is particularly pronounced for moderate values of the signal strength ρ and holds true despite the fact that, in these simulation experiments, the search space \mathcal{M} consists only of closed itemsets, thereby resulting in a much smaller number of candidate feature interactions than 2^p . If that were not the case, the resulting statistical power of Bonf-CMH would have been trivially 0 regardless of ρ . Perhaps most importantly, the results in Figure 5.6 strongly suggest that using FACS instead of LAMP- χ^2 does *not* lead to a loss of statistical power, making FACS an appealing choice to correct for a categorical covariate in significant pattern mining.

4. Since $\frac{FP}{\max(TP+FP,1)} \leq \mathbb{1}[FP > 0]$, it follows that the FDR is always bounded from above by the FWER. Hence, any procedure which controls the FWER at level α also controls the FDR at the same level.

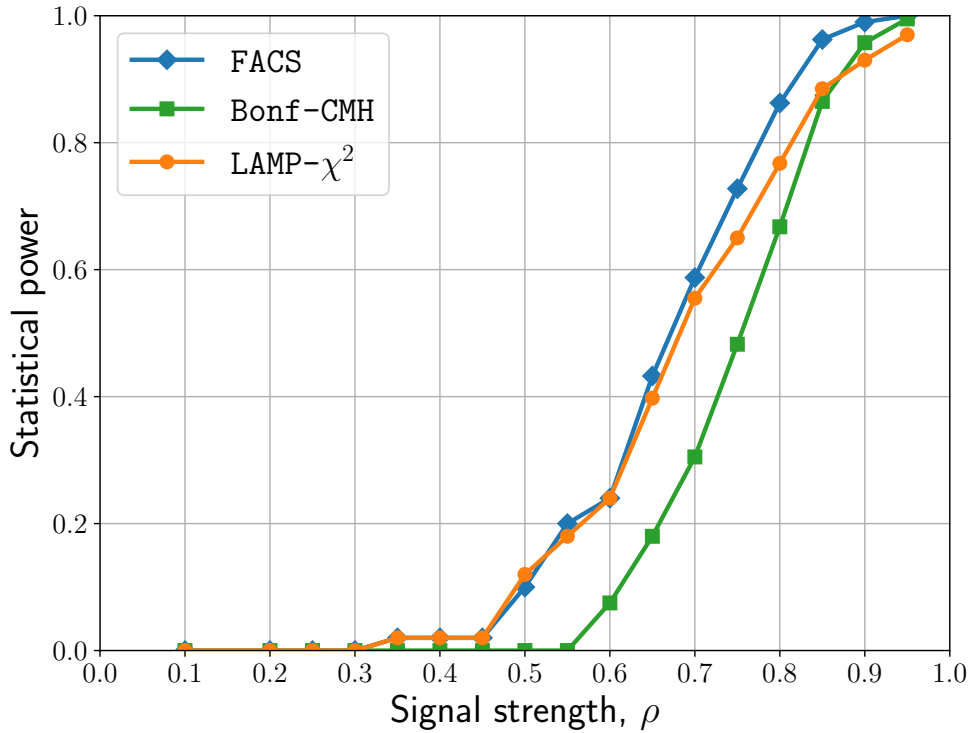


Figure 5.6. – Statistical power at different values of the signal strength ρ for our proposed approach FACS and two baseline algorithms: Bonf-CMH and LAMP- χ^2 .

Runtime

With the aim of benchmarking the computational efficiency of FACS against all baseline algorithms under consideration, we studied how the runtime of each method scales with respect to two key parameters of the input dataset: the number of features p and the number of categories for the covariate k . In a first set of experiments, we generated synthetic data with $n = 500$ samples and $k = 2$ categories for the covariate while varying the number of features p . Next, in a second set of experiments, the number of features was fixed to $p = 5,000$ while the number of categories for the covariate k varied between 2 and 28. The sample size was unchanged with respect to the first set of experiments, i.e. it was kept at $n = 500$. In all of these simulation experiments, both class labels and all categories for the covariate were equiprobable *a priori*, that is, $p_y = 0.5$ and $\mathbf{p}_c = \frac{1}{k}\mathbf{1}_k$. Finally, the probability p_u of a “background” feature being active was 0.1, as in the experiments devoted to evaluate statistical power and the FDR.

Figure 5.7 shows the resulting runtime as a function of the number of features p for FACS and all four baseline algorithms, 2^k -FACS, m^k -FACS, Bonf-CMH and LAMP- χ^2 . Most notably, these results highlight the computational virtues of Tarone’s concept of testability. Relative to all Tarone-based methods, the runtime of Bonf-CMH scales very poorly with the number of features p , rendering it unsuitable to analyse large datasets.

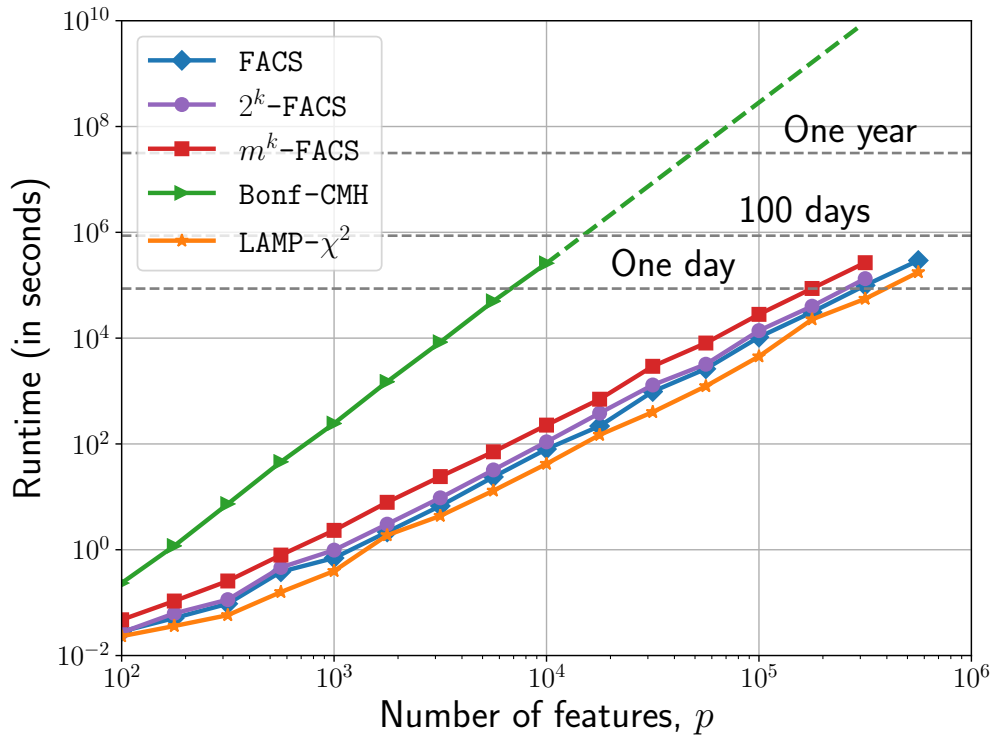


Figure 5.7. – Runtime as a function of the number of features p for our proposed approach FACS and four baseline algorithms: 2^k -FACS, m^k -FACS, Bonf-CMH and LAMP- χ^2 . The discontinuous trace shown for the baseline Bonf-CMH corresponds to forecasts made using a trend model of the form $\log_{10} T = b + \eta \log_{10} p$ rather than values actually measured via experiments.

In contrast, the runtimes of LAMP- χ^2 and FACS, as well as of the simplified versions of FACS, 2^k -FACS and m^k -FACS, can be seen to increase at the same rate as the number of features p grows. More precisely, the runtimes of all of these approaches differ only by a constant amount, which can be attributed to: (i) the overhead of evaluating the CMH test, with complexity $O(k)$, instead of evaluating Pearson’s χ^2 test, with constant complexity $O(1)$; and (ii) the overhead of evaluating the pruning condition for the CMH test, with complexity $O(k \log k)$ for FACS, $O(2^k)$ for 2^k -FACS and $O(n^k)$ for m^k -FACS, instead of evaluating the pruning condition for Pearson’s χ^2 test, again with constant complexity $O(1)$. Compared to the runtime gap between Bonf-CMH and all Tarone-based methods, the runtime overhead of FACS over LAMP- χ^2 which can be observed in Figure 5.7 is virtually negligible. Nevertheless, these simulations were carried out using only $k = 2$ categories for the covariate. Figure 5.8 complements this analysis by showing how the runtime of each algorithm increases with the number of categories for the covariate k . The results depicted in Figure 5.8 emphasise the key role that Algorithm 5.1 plays in making accounting for a categorical covariate in significant pattern mining computationally feasible. Indeed, without our efficient approach to evaluate the pruning condition for the CMH test, m^k -FACS actually becomes slower

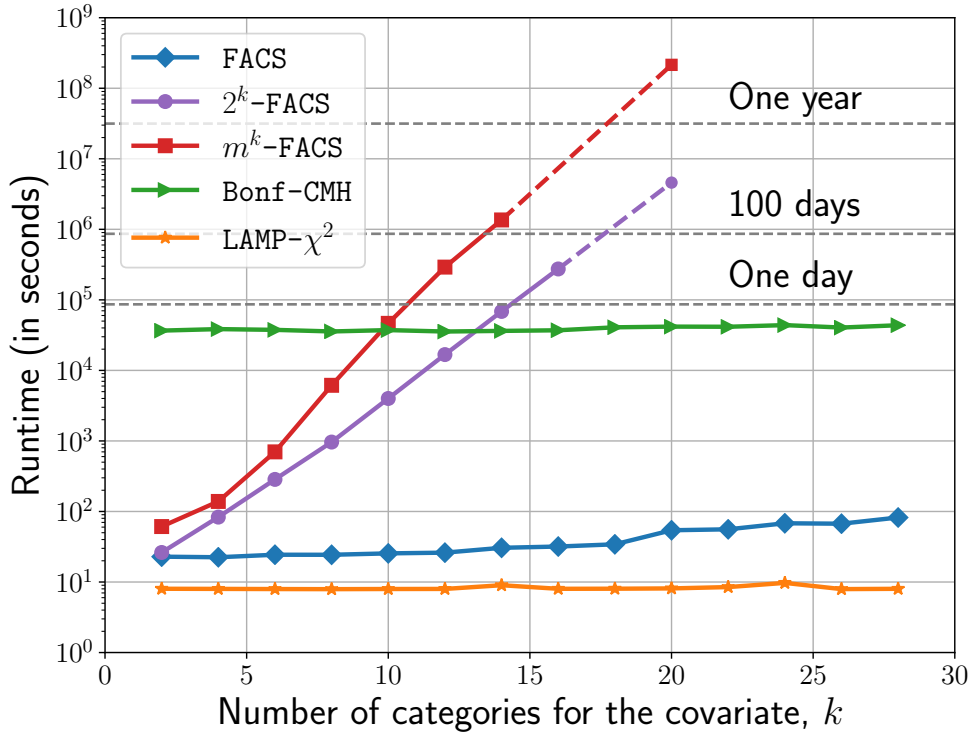


Figure 5.8. – Runtime as a function of the number of categories for the covariate k for our proposed approach FACS and four baseline algorithms: 2^k -FACS, m^k -FACS, Bonf-CMH and LAMP- χ^2 . The discontinuous traces shown for the baselines 2^k -FACS and m^k -FACS correspond to forecasts made using two different trend models of the form $\log_{10} T = b + \eta k$ rather than values actually measured via experiments.

than Bonf-CMH with as few as $k = 10$ categories for the covariate. In this particular case, the overhead of evaluating the pruning condition for the CMH test by solving Equation (5.9) using brute force turns out to be so large that simply applying no pruning at all is a more efficient strategy. Moreover, while 2^k -FACS is definitely faster than m^k -FACS, its scaling with respect to k is still unacceptable. In contrast, the runtime of the FACS algorithm scales very gently with k , confirming that it is able to account for a categorical covariate with a large number of categories without being drastically slower than LAMP- χ^2 .

5.6.3 Applications to genome-wide association studies

To conclude this chapter, a proof-of-concept application of FACS to discover significant high-order feature interactions in genome-wide association studies will be described. In particular, we will contrast the results obtained by FACS with those of two baseline algorithms, LAMP- χ^2 and Bonf-CMH, when analysing two datasets from the well-known collection of *A. thaliana* genome-wide association studies in [87]. Among

all binary phenotypes available in the collection, we chose LY (yellowing leaves) and *avrB* (hypersensitive-response traits) for our experiments. These two phenotypes have been shown to suffer from confounding due to population structure, making them a good test-bed for FACS. The original datasets, which we downloaded from the easyGWAS online resource [88], have sample size $n = 95$ and $n = 87$, respectively. Each of these samples can be represented as a p -dimensional binary vector, where the number of features is $p = 214,051$ for the phenotype LY and $p = 214,032$ for the phenotype *avrB*. Each of these binary features encodes a different single-nucleotide polymorphism, a certain position in the genome of the organism (loci) for which two different values (alleles) are frequently observed in the population. We followed the convention of representing the minor allele, i.e. the allele which is less frequently observed in the population, with a value of 1, whereas the major allele, i.e. the most common allele in the population, was encoded as 0.

DATA PREPROCESSING: Despite the increase in computing power available and recent progress in pattern mining, a direct application of significant itemset mining to genome-wide association studies is still computationally unfeasible. Besides the high dimensionality of the data, with the number of features p being in the order of hundreds of thousands or even millions, itemset mining datasets arising from genome-wide association studies tend to be very dense. For instance, the two datasets we will be working with in this section have approximately 25% of all entries being active (i.e. taking value 1). Nevertheless, itemset mining algorithms are typically applied to considerably sparser data. As a reference, the two most high-dimensional datasets considered in Chapter 4, Retail ($p = 16,470$) and Pumsb-star ($p = 7,117$), have only 0.06% and 0.71% of all entries active, respectively. While recent innovations such as parallel significant pattern mining algorithms [89] have successfully achieved considerable speed-ups, being able to quickly analyse dense, extremely high-dimensional datasets such as those originating from genome-wide association studies remains at large an open problem for significant pattern mining⁵. We will revisit the use of techniques from significant pattern mining to analyse genome-wide association studies in the next chapter, where we introduce a new instance of significant pattern mining, inspired precisely by this target application, which scales to considerably larger and denser datasets than significant itemset mining. However, our aim in this section is not to make new discoveries regarding the genetic architecture of the traits LY and *avrB* in *A. thaliana* but, rather, to elucidate the ability of FACS to correct for confounding also in real-world data. Therefore, we followed a pragmatic approach to bypass the current computational limitations of significant itemset mining by dividing the two original datasets into multiple smaller datasets, which we then analysed separately. As a first simplification, we did not consider interactions between single-nucleotide polymorphisms located in different chromosomes as candidate feature interactions in our search space. Since *A. thaliana* has five chromosomes, this allowed splitting each of the original datasets into five disjoint datasets. Moreover, we exploited linkage disequilibrium, the phenomenon that single-nucleotide polymorphisms often exhibit

5. For example, in spite of the efficiency of the parallelisation scheme proposed in [89], which reduces the runtime by a factor almost identical to the number of cores used, the resulting approach still mainly handles relatively sparse data, with 1% to at most 10% of all entries being active.

strong spatial correlations along the genomic sequence, to further divide each of the resulting datasets into even smaller datasets by downsampling the feature space by a factor of 20 using 20 different offsets. As a result of this process, each of the two original datasets was split into $5 \times 20 = 100$ smaller datasets, with each feature (single-nucleotide polymorphism) in the original datasets being part of exactly one of these 100 smaller datasets. These preprocessing steps brought down the number of features per dataset to a value between $p = 1,423$ and $p = 2,661$, thus considerably reducing the total runtime required to carry out our experiments.

METRICS: Directly assessing statistical power or the FDR in real-world data is challenging, as we most often lack the ground truth knowledge about which features, or interactions thereof, are truly associated with the class labels. In order to evaluate to which extent FACS can correct for confounding due to population structure, we will use the *genomic inflation factor* [90], a heuristic measure of confounding which is nonetheless exceedingly popular in statistical genetics. The genomic inflation factor, typically denoted as λ , is based on the assumption that the null hypothesis holds for the majority of association tests being carried out, i.e. that $|\mathcal{M}_0| \approx |\mathcal{M}|$, a condition satisfied by most genome-wide association studies. If that were indeed the case, the median value of all observed test statistics, i.e. the median of $\{t_{\mathcal{S}} \mid \mathcal{S} \in \mathcal{M}\}$, would be dominated by the set $\{t_{\mathcal{S}} \mid \mathcal{S} \in \mathcal{M}_0\}$ of observed test statistics for null patterns $\mathcal{S} \in \mathcal{M}_0$, thus being close to the median of the theoretical null distribution $\Pr(T = t \mid H_0)$. This motivates defining the genomic inflation factor as the ratio

$$\lambda = \frac{\text{median}(\{t_{\mathcal{S}} \mid \mathcal{S} \in \mathcal{M}\})}{\text{median}(\Pr(T = t \mid H_0))}$$

where $\text{median}(\{t_{\mathcal{S}} \mid \mathcal{S} \in \mathcal{M}\})$ is the empirical median of all observed test statistics and $\text{median}(\Pr(T = t \mid H_0))$ is the median of the theoretical null distribution of the test statistic of choice. Values of λ which differ substantially from one suggest a mismatch between the empirical distribution of the test statistics and the theoretical null distribution. Supposing that larger values of T correspond to stronger associations, a genomic inflation factor λ considerably larger than one indicates that a large fraction of all test statistics yielded more significant outcomes than expected under the null. Assuming that $|\mathcal{M}_0| \approx |\mathcal{M}|$, a possible explanation for this apparent contradiction could be that the observed test statistic values are affected by an uncorrected confounding covariate. While there are alternative reasons why the genomic inflation factor could deviate from one, such as the theoretical null distribution $\Pr(T = t \mid H_0)$ not being completely correct due to unmet assumptions made during its derivation or, simply, the condition $|\mathcal{M}_0| \approx |\mathcal{M}|$ not being true, a large genomic inflation factor suggests at the very least that the possibility that confounding played a role in shaping the results should be carefully studied.

Our experiments will contrast the genomic inflation factors that result from analysing each dataset using FACS, which corrects for population structure by means of a categorical covariate, and LAMP- χ^2 , which can only carry out unconditional association tests. As both approaches use a test statistic that follows a χ_1^2 distribution under the null, their genomic inflation factors are directly comparable. Moreover, to prevent the presumably large amount of untestable feature interactions from artificially decreasing

the genomic inflation factor, λ will be evaluated considering only testable feature interactions, i.e. the numerator of λ will be defined as $\text{median}(\{t_S \mid S \in \mathcal{M}_{\text{test}}(\delta_{\text{tar}})\})$. Apart from the average genomic inflation factor $\bar{\lambda}$ across all 100 datasets each of the two original datasets was divided into, we will also report the total number of “hits” obtained by each method, i.e. the total number of high-order feature interactions deemed significantly associated by FACS, LAMP- χ^2 and Bonf-CMH. Since the emphasis in these experiments is on comparing the results of the different approaches rather than on claiming new discoveries, we made the simplifying choice of only guaranteeing FWER-control at level $\alpha = 0.05$ for each dataset individually, as opposed to jointly controlling the FWER for the entire collection of 100 datasets. Note however that, if necessary, the latter criterion could have been satisfied by setting the significance threshold to be two orders of magnitude more stringent for each individual dataset.

CATEGORICAL COVARIATE: Population structure is a common source of confounding in genome-wide association studies. Confounding due to population structure tends to occur whenever a study includes individuals with different genetic ancestries and the proportion of individuals with a particular ancestry differs between cases and controls. Often the ground truth genetic ancestry of each sample is unknown, thus making population structure a latent (i.e. unobserved) covariate. In other cases, the ethnicity of each individual in the study might be known, but a finer-grained representation of genetic ancestry than provided by ethnicity alone might yield a better correction for confounding. As a result, most approaches to correct for population structure in genome-wide association studies infer the confounding covariate from the data rather than assuming it will be given as an external input. One of the most popular methods to learn a low-dimensional representation of genetic ancestry in a data-driven manner is EIGENSTRAT [91], which is loosely based on principal component analysis. Let $\{x_i\}_{i=1}^n$ be a set of observations, with $x_i = (u_{i,1}, u_{i,2}, \dots, u_{i,p})$ representing the genotype of the i -th individual at p single-nucleotide polymorphisms. Furthermore, for each $j = 1, 2, \dots, p$, define $\hat{\mu}_j = \frac{1}{n} \sum_{i=1}^n u_{i,j}$ and $\hat{\sigma}_j^2 = \frac{1}{n-1} \sum_{i=1}^n (u_{i,j} - \hat{\mu}_j)^2$ as the empirical mean and variance of the j -th feature. The *kinship matrix* $K \in \mathbb{R}^{n \times n}$ is proportional to the Gram matrix of the set $\{\tilde{x}_i\}_{i=1}^n$ of standardised vectors $\tilde{x}_i = (\tilde{u}_{i,1}, \tilde{u}_{i,2}, \dots, \tilde{u}_{i,p})$, that is, the genetic similarity between individuals i and i' can be quantified as

$$K_{i,i'} = \frac{1}{p} \langle \tilde{x}_i, \tilde{x}_{i'} \rangle = \frac{1}{p} \sum_{j=1}^p \tilde{u}_{i,j} \tilde{u}_{i',j},$$

where each standardised feature $\tilde{u}_{i,j} = \frac{u_{i,j} - \hat{\mu}_j}{\hat{\sigma}_j}$ has been mean-centered and scaled to have standard deviation equal to one. EIGENSTRAT represents the genetic ancestry of each individual in terms of the eigenvectors of the kinship matrix K . In particular, since K is self-adjoint, it can be diagonalised as $K = V \Sigma V^T$, with $V \in \mathbb{R}^{n \times n}$ being a unitary matrix whose columns are eigenvectors of K and Σ being a diagonal matrix containing the corresponding eigenvalues. EIGENSTRAT defines a d -dimensional, real-valued covariate $\mathbf{c}_i \in \mathbb{R}^d$ for each sample $i = 1, 2, \dots, n$ as the i -th row of $V_d \in \mathbb{R}^{n \times d}$, a matrix whose columns are the d eigenvectors in V associated with the d largest eigenvalues. Intuitively, the resulting set of covariates $\{\mathbf{c}_i\}_{i=1}^n$ provides the best rank- d approximation of the kinship matrix, thus constituting a set of d -dimensional

embeddings which capture the genetic similarity between individuals. In practice, the number d of eigenvectors used to represent genetic ancestry can be empirically determined using the genomic inflation factor λ as a proxy to quantify confounding; the smallest value of d that provides a satisfactory (i.e. sufficiently close to one) inflation factor is selected for the final analysis.

As discussed in Section 5.5, the FACS algorithm was designed to correct for a single categorical covariate, making it challenging to use the set of covariates $\{\mathbf{c}_i\}_{i=1}^n$ provided by EIGENSTRAT to represent genetic ancestry. To circumvent this limitation, we clustered the set of d -dimensional vectors $\{\mathbf{c}_i\}_{i=1}^n$ using the k -means algorithm, resulting in a set of $\{\tilde{c}_i\}_{i=1}^n$ of cluster assignments, with $\tilde{c}_i \in \{1, 2, \dots, k\}$ for each $i = 1, 2, \dots, n$. These cluster assignments can be understood as a data-driven discretisation of the original covariates into k distinct categories (the cluster centroids), motivating their use as the categorical covariate that FACS will correct for in order to account for population structure. In our experiments, we used the $d = 5$ eigenvectors corresponding to the five largest eigenvalues of the kinship matrix and selected the value of k in the range $\llbracket 2, 8 \rrbracket$ that yielded the lowest average genomic inflation ratio $\bar{\lambda}$, resulting in $k^* = 5$ for the phenotype LY and $k^* = 3$ for the phenotype *avrB*.

Table 5.1. – Total number of feature interactions (hits) deemed significantly associated by LAMP- χ^2 , FACS and Bonf-CMH and average genomic inflation factor $\bar{\lambda}$. The value of $\bar{\lambda}$ for Bonf-CMH is similar to FACS since both methods make use of the CMH test to correct for the same categorical covariate.

Dataset	FACS		LAMP- χ^2		Bonf-CMH
	hits	$\bar{\lambda}$	hits	$\bar{\lambda}$	hits
LY	433	1.17	100,883	3.18	19
<i>avrB</i>	43	1.21	546	2.38	1

RESULTS: The outcome of our experiments on *A. thaliana* is concisely described in Table 5.1. The most remarkable aspect of these results is the sharp decrease in genomic inflation that our proposed approach FACS achieves compared to LAMP- χ^2 , confirming the ability of FACS to correct for the effect of a confounding categorical covariate also in real-world data. The collection $\{t_{\mathcal{S}} \mid \mathcal{S} \in \mathcal{M}_{\text{test}}(\delta_{\text{tar}})\}$ of test statistics for testable feature interactions $\mathcal{S} \in \mathcal{M}_{\text{test}}(\delta_{\text{tar}})$ obtained by LAMP- χ^2 exhibits severe genomic inflation, with the empirical median being on average more than twice as large as expected under the null for the phenotype *avrB* ($\bar{\lambda} = 2.38$) and more than three times as large as expected for the phenotype LY ($\bar{\lambda} = 3.18$). In contrast, FACS successfully eliminates most of the inflation, resulting in average genomic inflation factors of only $\bar{\lambda} = 1.21$ for *avrB* and $\bar{\lambda} = 1.17$ for LY. To further investigate the impact that our discretisation heuristic has on the correction for population structure, we carried out three univariate analyses using logistic regression in combination with the Likelihood Ratio Test (LRT) [92]. P-values for each of the p original features were obtained for both datasets using (i) an uncorrected null model, (ii) a null model that uses the real-valued, d -dimensional covariates that EIGENSTRAT provides and (iii) a null model that uses the same categorical covariates as FACS. The results, summarised in Table 5.2, suggest that discretising the representation of genetic ancestry inferred by EIGENSTRAT does not

Table 5.2. – Genomic inflation factor λ for different *univariate* analyses of the two genome-wide association studies under consideration. Logistic regression (LogReg) P-values are obtained using the Likelihood Ratio Test (LRT) with three different null models: (i) only base rate (uncorrected), (ii) real-valued covariates as obtained by EIGENSTRAT and (iii) the same categorical covariate used by FACS, encoded using dummy variables. For comparison, the genomic inflation resulting from applying the CMH test only to individual features is also included.

Dataset	LogReg			CMH
	Uncorrected	EIGENSTRAT	Categorical	
LY	2.54	1.63	1.54	1.50
<i>avrB</i>	1.64	1.26	1.20	1.17

negatively affect the resulting correction for confounding due to population structure, if anything, it leads to slightly better results on these two datasets. Consequently, the faint inflation which remains in our results is not a byproduct of the need to discretise the covariate, but seems to originate from EIGENSTRAT instead. Presumably, these inflation values could be further reduced if the sample size was increased, allowing to infer a more accurate representation of genetic ancestry than possible with less than 100 samples per dataset. Another highlight of the results described in Table 5.1 is the fact that the decrease in genomic inflation is accompanied by a proportional decrease in the number of feature interactions deemed significantly associated (“hits”), hinting towards the possibility that a large majority of the discoveries made by LAMP- χ^2 in these experiments might be merely a product of confounding. Finally, the number of “hits” shown in Table 5.1 also hints towards Bonf-CMH lacking statistical power in real-world data, in line with the simulation results depicted in Figure 5.6. Despite using the same test statistic and correcting for the same categorical covariate as our method FACS, the number of discoveries made by Bonf-CMH is substantially smaller, most likely owing to the naive Bonferroni correction used by Bonf-CMH being too over-conservative compared to Tarone’s method.

GENOME-WIDE ASSOCIATION STUDIES AT A REGION LEVEL

During the last decade, *genome-wide association studies* (GWASs) have been used to systematically detect associations between individual genomic variants, most often *single-nucleotide polymorphisms* (SNPs), and a vast array of phenotypic traits (e.g. [93–95]). As of May 2018, GWASs have led to more than 61,000 novel SNP-trait associations being discovered [16, 17], substantially enriching our understanding of the biological mechanisms underlying disease [18, 19, 96] as well as other traits of fundamental importance (e.g. [97, 98]).

Standard GWASs typically compute correlations between individual SNPs and the phenotype of interest. As a result, the smaller the sample size is and the weaker the effect sizes of the associated variants are, the lower the resulting statistical power will be, thereby causing the detection of novel associations to become increasingly more challenging. A useful approach to improve the statistical power to detect a certain kind of weak associations in GWASs is to exploit *genetic heterogeneity* [99], the phenomenon that multiple genomic markers might have a similar effect on the phenotype. For genomic regions for which the assumption of genetic heterogeneity applies, the aggregation of multiple neighbouring variants into a joint *meta-marker* will provide a stronger, easier to detect signal. This leads naturally to the problem of finding genomic regions which are statistically associated with a phenotypic trait of interest, as opposed to focusing on individual markers. Nevertheless, given a dataset with p markers, there are $\frac{p(p+1)}{2} = O(p^2)$ distinct sets of contiguous markers one could test for association with the phenotype. For instance, in a standard GWAS dataset with one million SNPs there are more than 500 billion genomic regions one could potentially inspect. Owing to the substantial computational and statistical difficulties that performing such a vast number of association tests would entail, most existing approaches are limited to consider only a relatively small number of genomic regions chosen *a priori*. Some methods define the regions to be tested as certain functional units (e.g. exons, known regulatory regions or entire genes) whereas others aim to be less reliant on prior knowledge by dividing the genome into (possibly overlapping) fixed-length windows (e.g. [48, 100]). Nonetheless, regardless of the manner in which the candidate genomic regions are preselected, these will only cover a small subset of all possible genomic regions. As a consequence, the ability of these methods to detect novel associations will be considerably impaired if the set of candidate genomic regions was misspecified. For example, this would occur if the genome was split into regions which are not similar in length to the (unknown) truly associated regions or if the phenotypic variance was partly explained by variants not contained in the functional units under consideration. For existing approaches, introducing these *a priori* assumptions was a necessary sacrifice in order to obtain a manageable number of association tests. However, the flourishing of significant pattern mining has provided tools to directly deal with statistical association testing in enormous search spaces,

thus offering an opportunity to change the way in which genomic regions are assayed for association with phenotypic traits.

In this chapter we present the Fast Automatic Interval Search (FAIS) [51] and FastCMH [52] algorithms, two novel methods based on significant pattern mining which allow testing all genomic regions for association with a binary phenotype of interest. By eliminating the need to limit the analysis to regions of a certain length or located at certain positions, the resulting approaches are insensitive to misspecification of the set of candidate genomic regions to be tested, ultimately leading to a gain in statistical power.

The remainder of this chapter is organised as follows. Section 6.1 introduces our model of genetic heterogeneity and formalises the problem of testing genomic regions for association with a phenotypic trait as an instance of significant pattern mining. Next, Section 6.2 discusses the algorithmic aspects of the resulting formulation, describing how the methods presented in Chapters 3, 4 and 5 can be adapted to this specific application. In particular, Section 6.2 will introduce FAIS, our first contribution to the problem of region-wise GWAS, and FastCMH, an extension of FAIS which allows to correct for a categorical covariate. Finally, the chapter concludes with an exhaustive experimental evaluation of our novel approaches on simulated data, several *A. thaliana* datasets and a study of Chronic Obstructive Pulmonary Disease (COPD), which will be presented in Section 6.3.

6.1 INTRODUCTION

In this chapter we will consider a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ consisting of n observations $x \in \mathcal{X}$ and their corresponding class labels $y \in \{0, 1\}$. We will also consider the setting discussed in Chapter 5, where each observation-label pair is additionally tagged by a categorical covariate c with k categories, resulting in a dataset $\mathcal{D} = \{(x_i, y_i, c_i)\}_{i=1}^n$ with $c_i \in \{1, 2, \dots, k\}$ for each sample $i = 1, 2, \dots, n$. In the previous chapters of this thesis the specific nature of observations, class labels and categorical covariates (if available) was deliberately unspecified for the sake of generality. However, all methods that will be introduced in this chapter aim specifically to analyse data originating from GWASs. Therefore, we will first provide an in-depth description of such data, which will then serve as guidance to propose *significant region mining*, a novel instance of significant pattern mining designed to carry out GWAS at a region level. Finally, we will contrast the resulting problem statement with that of *burden tests*, a widely used family of approaches for region-wise GWAS, which nevertheless only consider fixed-length genomic windows as candidate regions.

Data description

Throughout this chapter each observation x will be a representation of the genotype of a certain individual or specimen based on a set of p genomic markers measured at different positions (loci) in its genome. However, unlike in significant itemset mining, these features are not exchangeable but rather have an intrinsic order depending on the position each marker has along the genome. To emphasise this conceptual difference, we will modify the notation and explicitly denote x as a sequence instead

of as a p -dimensional vector. In this way, $x_i(j)$ will stand for the value which the j -th genomic variant takes in the i -th individual. Since GWAS datasets are the main focus of this chapter, we will consider that the p genomic markers which make up each observation x are SNPs, specific locations in the genome of an organism where two different configurations (alleles) are commonly observed in the population under study. For any given SNP, the least common allele in the population is often denoted the *minor allele* whereas the most common allele will be referred to as the *major allele*. Since many organisms contain multiple sets of chromosomes, including humans, it is possible that an individual has the minor allele for some sets of chromosomes and the major allele for the others. In other words, if an organism had q sets of chromosomes, each SNP could in principle attain 2^q different configurations. In practice, *SNP arrays*, the technology that has been used to obtain most existing GWAS datasets, only allow to measure directly the number of sets of chromosomes which have the minor allele for each of the p SNPs in the array. As a result, SNP arrays can only distinguish between $q + 1$ disjoint sets of configurations for each SNP in the array. For instance, humans have $q = 2$ sets of chromosomes, implying that each SNP assayed by a SNP array can take one of three possible values: (i) two major alleles, typically represented as 0; (ii) one minor allele and one major allele, denoted as 1; and (iii) two minor alleles, encoded as 2. In other words, for human GWAS datasets, $x_i(j) \in \llbracket 0, 2 \rrbracket$ for each variant $j = 1, 2, \dots, p$ and individual $i = 1, 2, \dots, n$. More generally, $x_i(j)$ could be considered to be a count variable with range $\llbracket 0, q \rrbracket$ that indicates the number of copies of the minor allele that the i -th individual has at the j -th SNP. Each observation x will be accompanied by a binary class label y , used to represent the case/control status of an individual when studying disease phenotypes or, more generally, the presence or absence of a certain phenotypic trait of interest. Finally, in the context of GWAS, the categorical covariate (if any) will be utilised to correct for the potentially confounding effect of factors such as population structure, age or gender, among others.

Significant Region Mining

The goal of this section is to propose significant region mining, an instance of significant pattern mining specifically designed to test genomic regions for association with a phenotype of interest under a model of genetic heterogeneity. Following the example of Section 2.1, which formally introduced significant itemset mining and significant subgraph mining, there are three more elements needed to characterise a certain formulation of significant pattern mining, apart from the type of dataset \mathcal{D} under consideration: (i) the notion of pattern \mathcal{S} ; (ii) the search space \mathcal{M} consisting of all candidate patterns \mathcal{S} to be studied; and (iii) the concept of occurrence of a pattern \mathcal{S} in an observation x .

In significant region mining, each pattern \mathcal{S} will correspond to a different genomic region. Since a genomic region is simply a set of contiguous markers, each pattern \mathcal{S} can be uniquely represented by the starting and ending positions of the corresponding genomic region, j_s and j_e , which must satisfy $1 \leq j_s \leq j_e \leq p$. In this way, each pattern $\mathcal{S} = \llbracket j_s, j_e \rrbracket$ indexes $|\mathcal{S}| = j_e - j_s + 1$ contiguous genomic markers, a quantity we will refer to as the *length* of the pattern \mathcal{S} . The search space \mathcal{M} comprises the set of all possible genomic regions, regardless of starting position and length,

i.e. $\mathcal{M} = \{\llbracket j_s, j_e \rrbracket \mid 1 \leq j_s \leq j_e \leq p\}$. The resulting search space \mathcal{M} thus contains $\frac{p(p+1)}{2}$ candidate regions, a much larger number than considered by any competing approaches for region-wise GWAS. Nevertheless, compared to alternative formulations of significant pattern mining discussed in this thesis, the size of the search space increases relatively gently with the number of features p . For example, in significant region mining the number of patterns only grows quadratically in p , i.e., $|\mathcal{M}| = O(p^2)$. In contrast, the size of the search space in significant itemset mining grows exponentially in p , that is, $|\mathcal{M}| = O(2^p)$. Consequently, significant region mining can be applied to considerably larger and denser datasets than significant itemset mining, allowing to circumvent the scalability limitations that hinder the usefulness of the latter framework in GWASs.













	y	c	p ordered genomic markers	g_{S_1}	g_{S_2}
n_1 cases 	1		1 0 0 0 0 0 2 1 0 0 1 0 0 0 0 0 0 2 0 1 x_1	0	1
	1		1 0 0 2 0 0 0 2 1 0 0 0 0 2 0 1 0 1 0 1 x_2	1	1
	1		2 0 0 0 0 1 0 1 1 0 0 0 0 0 1 0 0 1 0 1 x_3	1	1
	1		2 0 0 0 1 0 0 2 1 0 0 0 0 0 0 2 0 1 0 2 \vdots	1	1
	1		1 0 2 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 \vdots	1	0
$n - n_1$ controls 	0		1 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 2 0 1 \vdots	0	1
	0		2 0 0 0 0 0 0 2 1 0 0 0 0 2 2 0 0 1 0 1 \vdots	0	1
	0		1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 x_{n-2}	0	0
	0		1 0 2 0 0 1 0 2 1 0 0 0 0 0 0 0 0 1 0 1 x_{n-1}	1	0
	0		2 0 0 0 0 0 0 2 1 0 0 0 0 0 0 0 0 1 0 2 x_n	0	0
			$S_1 = \llbracket 3, 6 \rrbracket$	$S_2 = \llbracket 12, 16 \rrbracket$	meta-markers

Figure 6.1. – An illustration of how genetic heterogeneity can be exploited to gain statistical power in GWASs. Two genomic regions, $S_1 = \llbracket 3, 6 \rrbracket$ (green) and $S_2 = \llbracket 12, 16 \rrbracket$ (red), contain markers which are weakly associated with the phenotype. However, by combining these markers into meta-markers $g_{S_1}(x)$ and $g_{S_2}(x)$ that indicate whether the region contains minor alleles or not, stronger signals emerge. In this particular example, $g_{S_1}(x)$ and $g_{S_2}(x)$ are both marginally associated with the phenotype, while only $g_{S_1}(x)$ remains associated once the categorical covariate c is taken into account.

The notion of occurrence of a pattern \mathcal{S} in an observation x will be defined in such a way that, if the assumption of genetic heterogeneity holds for the genomic region corresponding to \mathcal{S} , the resulting pattern occurrence indicator $g_{\mathcal{S}}(x)$ would aggregate the (possibly weak) effects of all markers indexed by \mathcal{S} . Under a model of genetic heterogeneity, multiple genomic markers in close proximity might have evolved to

affect the phenotype of interest in a similar way. For example, a certain genomic region might be particularly sensitive to perturbations, leading to any mutation within the region causing a disruption of its functionality. Motivated by this interpretation, we will say that pattern \mathcal{S} is present in an observation x if *any* of the markers indexed by \mathcal{S} have *one or more* copies of the minor allele. More precisely, let $b_{\mathcal{S}}(x) = \sum_{j \in \mathcal{S}} x(j)$ be the *burden count*, that is, the total number of minor alleles contained in the genomic region. Then, the pattern occurrence indicator will be given by $g_{\mathcal{S}}(x) = \mathbb{1}[b_{\mathcal{S}}(x) > 0]$. If the assumption of genetic heterogeneity holds for the markers indexed by \mathcal{S} and the effect signs are homogeneous within the region, the meta-marker $g_{\mathcal{S}}(x)$ will have a stronger association with the phenotype than any of the individual markers within the genomic region. This idea is illustrated in Figure 6.1, which depicts a toy GWAS dataset with $n = 10$ individuals ($n_1 = 5$ cases and $n - n_1 = 5$ controls) genotyped at $p = 20$ SNPs. In this example, two genomic regions contain markers which are weakly associated with the phenotype, $\mathcal{S}_1 = \llbracket 3, 6 \rrbracket$ (green) and $\mathcal{S}_2 = \llbracket 12, 16 \rrbracket$ (red). However, the meta-markers $g_{\mathcal{S}_1}(x)$ and $g_{\mathcal{S}_2}(x)$ that indicate the presence or absence of minor alleles within the regions exhibit a much stronger association. Moreover, the example in Figure 6.1 also shows that confounding affects significant region mining as much as any other instance of significant pattern mining. In particular, while both meta-markers $g_{\mathcal{S}_1}(x)$ and $g_{\mathcal{S}_2}(x)$ are marginally associated with the phenotype, only $g_{\mathcal{S}_1}(x)$ remains associated after correcting for the categorical covariate c that represents the genetic ancestry of each sample.

In summary, significant region mining aims to find *all* genomic regions, regardless of starting position or length, for which the presence or absence of any number of minor alleles within the region is significantly associated (either marginally or conditioned on a categorical covariate) with a binary phenotypic trait of interest.

Related work: burden tests

Among all existing approaches to carry out GWASs at a region level, burden tests [48] are the closest to significant region mining. Burden tests also exploit genetic heterogeneity, proposing a variety of ways to obtain meta-markers for each genomic region under consideration. In particular, the Cohort Allelic Sums Test (CAST) [101] aggregates the markers in each region using exactly the same procedure as significant region mining, i.e. as a binary variable that indicates the presence or absence of minor alleles in the region. Thus, for a single genomic region, the burden test CAST and significant region mining are statistically indistinguishable. Nevertheless, what sets burden tests and our newly proposed framework apart is the way in which the set of candidate genomic regions to be tested is defined. Burden tests bypass the difficulties involved in testing all possible genomic regions by requiring the user to specify *a priori* a small subset of candidate regions to be tested. For example, gene-based burden tests make use of domain knowledge, defining candidate regions either as entire genes or as entire exons (coding regions of genes). A more popular alternative, which allows to also include in the analysis variants that do not lie in or nearby genes, is to partition the genome into windows of fixed length, which can be either non-overlapping or sliding. More precisely, given a dataset with p genomic markers and a desired window length w , a burden test with non-overlapping windows would consider as candidate

genomic regions the set $\mathcal{M} = \{\llbracket (i-1)w + 1, \min(iw, p) \rrbracket \mid 1 \leq i \leq \lceil \frac{p}{w} \rceil\}$. In contrast, a burden test with sliding windows would extend the set of genomic regions under consideration to all genomic windows of length w , regardless of starting position, i.e. $\mathcal{M} = \{\llbracket j_s, j_s + w - 1 \rrbracket \mid 1 \leq j_s \leq p - w + 1\}$.

Hence, while significant region mining tests all $\frac{p(p+1)}{2}$ possible genomic regions, burden tests introduce *a priori* assumptions to reduce the number of candidate genomic regions to $\lceil \frac{p}{w} \rceil$ if only non-overlapping windows are considered or $p - w + 1$ for the case of sliding windows. As a result, both types of approaches can exhibit drastically different behaviours, being complementary to some extent. Since burden tests perform a much smaller number of association tests, the resulting multiple comparisons correction will be less stringent. Therefore, burden tests will outperform significant region mining if the truly associated genomic regions satisfy the assumptions made by the burden tests, e.g. if those regions have a length close to w . However, on most occasions, the length of the truly associated regions is unknown and could even vary considerably from one truly associated region to another, leading to difficulties in choosing a value for the hyperparameter w . Moreover, attempts to select w in a data-driven manner, for example, by inspecting the results corresponding to different choices for w and keeping the most promising, would require an additional correction for multiple comparisons, an observation that might remain unnoticed by practitioners without a background in statistics. Thus, the need to set the hyperparameter w can lead to unintentional data dredging, potentially compromising the reproducibility of the results. In contrast, significant region mining exploits advances in significant pattern mining to test all genomic regions in a computationally and statistically efficient manner. Significant region mining approaches are therefore robust to misspecification of the set of regions to be tested, leading to improved statistical power in situations where the optimal choice for w is unknown *a priori*. Besides, by eliminating a key hyperparameter from the model, the potential to misuse the resulting methods is considerably reduced.

6.2 METHOD

In the previous section we introduced significant region mining as a novel instance of significant pattern mining, motivated by the problem of testing genomic regions for association with a binary phenotype under a model of genetic heterogeneity. We postulated that, by testing *all* genomic regions instead of only regions of a prespecified length, significant region mining has the potential to outperform burden tests in terms of statistical power whenever the length of the truly associated regions is not known *a priori* or might vary considerably throughout the genome. The remainder of this section will be devoted to show how significant region mining can be formalised as a restricted instance of significant itemset mining, allowing us to utilise the algorithmic machinery described in previous chapters of this thesis also in this novel setting.

Significant region mining as a restricted instance of significant itemset mining

We will consider two variants of significant region mining, depending on whether we look for *marginal* associations between genomic regions and the phenotype or for *conditional* associations that account for the effect of a categorical covariate. Formally, let Y be a binary random variable representing a phenotype under study and X a random sequence with p genomic markers. For each pattern $\mathcal{S} = \llbracket j_s, j_e \rrbracket$ in the search space $\mathcal{M} = \{\llbracket j_s, j_e \rrbracket \mid 1 \leq j_s \leq j_e \leq p\}$, define $G_{\mathcal{S}}(X) = \mathbb{1} \left[\sum_{j \in \mathcal{S}} X(j) > 0 \right]$ as a binary random variable that indicates whether X contains a non-zero number of copies of a minor allele in the genomic region corresponding to \mathcal{S} . In the first variant of significant region mining, we aim to find all $\mathcal{S} \in \mathcal{M}$ for which $G_{\mathcal{S}}(X) \not\perp Y$. In contrast, in the second variant, the goal is to discover all $\mathcal{S} \in \mathcal{M}$ which satisfy $G_{\mathcal{S}}(X) \not\perp Y \mid C$, where C is a categorical random variable with k categories. Each of these two problem statements has already been studied in this thesis in the context of general-purpose significant pattern mining algorithms. In particular, this includes significant itemset mining, arguably the “classical” instance of significant pattern mining which most closely resembles significant region mining. The fundamental differences between these two significant pattern mining paradigms are:

- (i) In both cases, a pattern \mathcal{S} can be understood as a subset of features, that is, $\mathcal{S} \subseteq \{1, 2, \dots, p\}$. However, the collection of patterns (feature subsets) which make up the search space sets both significant pattern mining formulations apart. The search space in significant region mining contains only patterns \mathcal{S} such that all features in \mathcal{S} have consecutive indices, i.e. $\mathcal{S} = \llbracket j_s, j_e \rrbracket$ for some $1 \leq j_s \leq j_e \leq p$. In contrast, the search space in significant itemset mining contains *all* possible feature subsets, regardless of whether the features in \mathcal{S} are contiguous or not.
- (ii) At first glance, the notion of occurrence of a pattern \mathcal{S} in a sample x is completely different between both paradigms. In significant itemset mining, all features in $x = (u_1, u_2, \dots, u_p)$ are assumed to be binary, and a pattern \mathcal{S} is said to occur in x if and only if all features indexed by \mathcal{S} have value one, i.e. $g_{\mathcal{S}}(x) = \prod_{j \in \mathcal{S}} u_j$. In contrast, in significant region mining the features in x are integers in $\llbracket 0, q \rrbracket$, and a pattern \mathcal{S} is said to occur in x if and only if the sum $\sum_{j \in \mathcal{S}} x(j)$ is non-zero, i.e. $g_{\mathcal{S}}(x) = \mathbb{1} \left[\sum_{j \in \mathcal{S}} x(j) > 0 \right]$.

The implications of observation (i) above are mainly algorithmic. The search space in significant itemset mining is enormous, comprising 2^p candidate patterns. Moreover, the number of patterns containing $|\mathcal{S}|$ features is $\binom{p}{|\mathcal{S}|}$. Intuitively, the width of the pattern enumeration tree thus grows very rapidly as we descend levels in the tree. Breadth-first traversal strategies in this setting would require too much memory, offering a justification as of why all algorithms presented in this thesis so far use depth-first traversal strategies instead. Nevertheless, this restriction does not apply to significant region mining. In this instance of significant pattern mining, there are only $\frac{p(p+1)}{2}$ patterns in the search space. Moreover, the widest level of the pattern enumeration tree is the first level, which contains p patterns (regions) of length $|\mathcal{S}| = 1$. More generally, there are $p - |\mathcal{S}| + 1$ patterns (regions) containing $|\mathcal{S}|$ contiguous features, implying that the pattern enumeration tree gets narrower as we descend

levels in the tree. Breadth-first traversal of the pattern enumeration tree is hence not only feasible in significant region mining, but also advantageous compared to a depth-first traversal strategy. For each pattern $\mathcal{S} = \llbracket j_s, j_e \rrbracket$ of length greater than one, there are exactly two subset patterns in the preceding level, $\mathcal{S}_l = \llbracket j_s + 1, j_e \rrbracket$ and $\mathcal{S}_r = \llbracket j_s, j_e - 1 \rrbracket$. Since \mathcal{S} is a superset of both \mathcal{S}_l and \mathcal{S}_r , it can be pruned from the search space if the appropriate pruning condition (see Sections 3.4 and 5.4) applies to either \mathcal{S}_l or \mathcal{S}_r . Breadth-first traversal, which would allow to verify the pruning condition for both \mathcal{S}_l and \mathcal{S}_r before visiting pattern \mathcal{S} , can therefore prune the search space more effectively than depth-first traversal, which would only allow to verify the pruning condition for either \mathcal{S}_l or \mathcal{S}_r prior to visiting \mathcal{S} . Owing to this observation, the significant region mining algorithms which will be presented in this chapter traverse the pattern enumeration tree in a breadth-first manner.

Observation (ii) may initially seem to be the biggest hurdle towards adapting existing significant itemset mining algorithms to perform significant region mining instead. Indeed, it can be readily seen that patterns in significant region mining do not obey the Apriori property of pattern mining (Proposition 3.1), which plays a fundamental role in the search space pruning conditions used by all significant pattern mining algorithms covered in this thesis. Instead, the support of patterns in significant region mining is nondecreasing as we descend levels in the pattern enumeration tree. That is, in significant region mining, $\mathcal{S}' \supset \mathcal{S}$ implies that $r_{\mathcal{S}'} \geq r_{\mathcal{S}}$ instead of $r_{\mathcal{S}'} \leq r_{\mathcal{S}}$. This discrepancy can however be reconciled in at least two different yet ultimately equivalent ways. One possibility is to exploit the fact that, as shown in Section 3.3, the minimum attainable P-value function corresponding to Pearson's χ^2 test and Fisher's exact test is symmetric around $\frac{n}{2}$, i.e. $p_{\min}(r_{\mathcal{S}}) = p_{\min}(n - r_{\mathcal{S}})$. This allows to prove that a valid pruning condition for significant region mining can be obtained by simply replacing condition (ii) in Proposition 3.4 by (ii)': $r_{\mathcal{S}} \geq n_b$, with $n_b = \max(n_1, n - n_1)$. This derivation, which was used in our first publication on GWASs at a region level [51], can also be applied to the CMH test by noting that its minimum attainable P-value function satisfies $p_{\min}(\mathbf{r}_{\mathcal{S}}) = p_{\min}(\mathbf{n} - \mathbf{r}_{\mathcal{S}})$. Due to the reversal of the Apriori property of pattern mining in significant region mining, the lower envelope of the minimum attainable P-value (Definition 5.2) is given by $\tilde{p}_{\min}(\mathbf{r}_{\mathcal{S}}) = \min_{\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}_c(\mathbf{r}_{\mathcal{S}})} p_{\min}(\mathbf{r}_{\mathcal{S}'})$, where $\mathcal{B}_c(\mathbf{r}_{\mathcal{S}}) = \llbracket r_{\mathcal{S},1}, n_1 \rrbracket \times \llbracket r_{\mathcal{S},2}, n_2 \rrbracket \times \cdots \times \llbracket r_{\mathcal{S},k}, n_k \rrbracket$ is the set of all $\mathbf{r}_{\mathcal{S}'}$ satisfying $r_{\mathcal{S}',c} \geq r_{\mathcal{S},c}$ for all $c = 1, 2, \dots, k$. However, due to the symmetry of $p_{\min}(\mathbf{r}_{\mathcal{S}})$, we have that $\min_{\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}_c(\mathbf{r}_{\mathcal{S}})} p_{\min}(\mathbf{r}_{\mathcal{S}'}) = \min_{\mathbf{r}_{\mathcal{S}'} \in \mathcal{B}(\mathbf{n} - \mathbf{r}_{\mathcal{S}})} p_{\min}(\mathbf{r}_{\mathcal{S}'})$, where the right-hand side equals $\tilde{p}_{\min}(\mathbf{n} - \mathbf{r}_{\mathcal{S}})$ as defined in Equation (5.9). As a result, all developments presented in Chapter 5, including Algorithm 5.1, which allows to evaluate the lower envelope of the minimum attainable P-value with complexity $O(k \log k)$, can be readily applied to significant region mining as well. A second possibility to reconcile both significant pattern mining paradigms is to exploit the fact that two binary random variables $G_{\mathcal{S}}(X)$ and Y are statistically associated if and only if $1 - G_{\mathcal{S}}(X)$ and Y are statistically associated as well. This holds true for both marginal and conditional associations. This phenomenon can also be inferred directly from symmetries exhibited by all test statistics discussed in this thesis. Indeed, the P-value functions for Pearson's χ^2 test (Equation (2.6)) and Fisher's exact test (Equation (2.8)) satisfy $p_{\text{pearson}}(a_{\mathcal{S}} \mid n, n_1, r_{\mathcal{S}}) = p_{\text{pearson}}(n_1 - a_{\mathcal{S}} \mid n, n_1, n - r_{\mathcal{S}})$ and $p_{\text{fisher}}(a_{\mathcal{S}} \mid n, n_1, r_{\mathcal{S}}) = p_{\text{fisher}}(n_1 - a_{\mathcal{S}} \mid$

$n, n_1, n - r_S$). Analogously, the P-value function of the CMH test (Equation (5.2)) satisfies $p_{\text{cmh}}(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S) = p_{\text{cmh}}(\mathbf{n}_1 - \mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{n} - \mathbf{r}_S)$. In summary, redefining the notion of occurrence of a pattern \mathcal{S} in a sample x by negating the pattern occurrence indicator $g_S(x)$ does *not* affect the assessment of the significance of \mathcal{S} . Applying this negation to significant region mining restores the Apriori property of pattern mining, allowing us to use all derivations and algorithms presented in this thesis unchanged. This alternative derivation of significant region mining best illustrates how closely related this formulation of significant pattern mining is to significant itemset mining. Applying De Morgan’s laws, we have that $1 - g_S(x) = \prod_{j \in \mathcal{S}} \mathbb{1}[x(j) = 0]$. Therefore, significant region mining is entirely equivalent to significant itemset mining provided that (i) the search space \mathcal{M} is restricted to contain only feature subsets for which all features are contiguous and (ii) the original, integer-valued genomic markers $\{x(j)\}_{j=1}^p$ are represented by binary features $\{u_j\}_{j=1}^p$ defined as $u_j = 1$ if $x(j) = 0$ and $u_j = 0$ otherwise.

The possibility to view significant region mining as a restricted instance of significant itemset mining allows us to design algorithms for region-wise GWASs borrowing from methods already introduced in this thesis. We will use this insight to present a novel framework for significant region mining using Tarone’s method to correct for the multiple comparisons problem. This framework encompasses two different yet closely related approaches, FAIS and FastCMH. FAIS, our first contribution on this topic, predates the development of techniques to correct for confounding covariates introduced in Chapter 5 of this thesis. As a result, it was designed to look for genomic regions marginally associated with the phenotype of interest. FastCMH, a more recent contribution, combines FAIS with the techniques described in Chapter 5, allowing to correct for the potentially confounding effect of covariate factors ubiquitous in GWASs such as population structure, gender or age, among others. However, from an algorithmic perspective, both approaches are virtually indistinguishable. The only aspects which differ between both methods are the computation of the minimum attainable P-value (Sections 3.3 and 5.3), the assessment of the pruning condition (Sections 3.4 and 5.4) and the calculation of the P-values of testable regions (Sections 2.2 and 5.2), all of which can be treated as “black boxes” in this chapter. Therefore, for the sake of clarity, FAIS and FastCMH will be presented next as a unified approach for significant region mining. The section will then conclude discussing two extensions of FAIS and FastCMH. First, we will describe how to make use of the developments introduced in Chapter 4, allowing us to utilise permutation testing to exploit the dependence between the test statistics of overlapping genomic regions, thereby improving statistical power. Finally, we will consider a possible generalisation of our model of genetic heterogeneity that can nonetheless still be tackled with the same algorithms.

A general algorithm for significant region mining

Given an input dataset \mathcal{D} as described in Section 6.1 above and a desired FWER α , the goal of our significant region mining algorithms is to find a set $\mathcal{M}_{\text{sig, filt}}$ of non-overlapping genomic regions which are significantly associated with the phenotype

of interest. If the dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ does not contain a covariate, a situation which corresponds to FAIS, genomic regions will be surveyed for marginal association with the phenotype. In contrast, if the dataset $\mathcal{D} = \{(x_i, y_i, c_i)\}_{i=1}^n$ also contains a categorical covariate, the association between genomic regions and phenotype will be assessed conditioned on the covariate, a setting corresponding to FastCMH.

Algorithm 6.1 Significant Region Mining

Input: Dataset \mathcal{D} , target FWER α

Output: Set of non-overlapping clusters of overlapping significantly associated genomic regions $\mathcal{M}_{\text{sig,clustered}}$

- 1: $\delta \leftarrow \text{compute_significance_threshold}(\mathcal{D}, \alpha)$
 - 2: $\mathcal{M}_{\text{sig,raw}} \leftarrow \text{retrieve_significant_regions}(\mathcal{D}, \delta)$
 - 3: $\mathcal{M}_{\text{sig,clustered}} \leftarrow \text{cluster_overlapping_regions}(\mathcal{M}_{\text{sig,raw}})$
 - 4: Return $\mathcal{M}_{\text{sig,clustered}}$
-

Regardless of the presence or absence of a covariate, our significant region mining algorithms all proceed in three main steps, as illustrated by Algorithm 6.1. The first step is to compute the corrected significance threshold δ using Tarone’s method, with the goal of guaranteeing FWER control at level α . As described in Chapter 3, obtaining δ requires traversing the pattern enumeration tree while iteratively adjusting an estimate $\hat{\delta}$ of δ . This procedure is carried out by the routine `compute_significance_threshold`, invoked in Line 1 and described in Algorithm 6.2 below. Once the corrected significance threshold δ has been obtained, the next step is to retrieve the set $\mathcal{M}_{\text{sig,raw}}$ of significantly associated genomic regions. According to the discussion in Section 3.5, we chose to implement this by traversing the pattern enumeration tree a second time, thus avoiding the need to keep all testable regions in memory at any point during the execution of Algorithm 6.1. This procedure is performed by the routine `retrieve_significant_regions`, invoked in Line 2 and described in Algorithm 6.3 below. While these two first steps fully parallel the general-purpose significant pattern mining algorithm of Chapter 3, our significant region mining algorithms include an additional step, aimed at enhancing the interpretability of the results. As extensively discussed in Chapter 4, related patterns tend to have strongly correlated test statistics. In particular, if a certain genomic region \mathcal{S} were truly associated with the phenotype, it would be plausible that some genomic regions which overlap with \mathcal{S} show a significant association as well. Conceptually, this means that the set $\mathcal{M}_{\text{sig,raw}}$ of significantly associated genomic regions tends to be composed of disjoint clusters, each comprising multiple overlapping genomic regions. To eliminate this inherent redundancy and facilitate the interpretation of the results provided by our significant region mining algorithms, the routine `cluster_overlapping_regions` invoked in Line 3 agglutinates overlapping, significantly associated genomic regions into a set $\mathcal{M}_{\text{sig,clustered}}$ of disjoint clusters, as described in Algorithm 6.4.

Out of these three steps, obtaining Tarone’s corrected significance threshold δ is in practice the most computationally demanding. The routine summarised in Algorithm 6.2, `compute_significance_threshold`, performs this task in a virtually identical manner as Algorithm 3.2. The key difference between both approaches lies in the way the pattern enumeration tree is traversed. As previously discussed,

Algorithm 6.2 `compute_significance_threshold`**Input:** Dataset \mathcal{D} , target FWER α **Output:** Corrected significance threshold δ

```

1: function compute_significance_threshold( $\mathcal{D}, \alpha$ )
2:   Initialise global variable  $\hat{\delta} \leftarrow 1$ 
3:   init_specific()
4:   Create empty queue region_queue
5:   for  $j_s = 1, 2, \dots, p$  do region_queue.enqueue( $\llbracket j_s, j_s \rrbracket$ )
6:   while region_queue is not empty do
7:      $\mathcal{S} \leftarrow$  region_queue.dequeue()  $\triangleright \mathcal{S} = \llbracket j_s, j_e \rrbracket$ 
8:     Compute the minimum attainable P-value  $p_{\mathcal{S}, \min}$   $\triangleright$  see Sections 3.3 and 5.3
9:     if  $p_{\mathcal{S}, \min} \leq \hat{\delta}$  then process_region( $\mathcal{S}$ )
10:    cond  $\leftarrow$  pruning_condition( $\mathcal{S}, \hat{\delta}$ )  $\triangleright$  see Sections 3.4 and 5.4
11:    if  $j_s \neq 1$  and  $j_s - j_{s,p} = 1$  and not (cond_p or cond) then
12:      region_queue.enqueue( $\llbracket j_s - 1, j_e \rrbracket$ )
13:     $j_{s,p} \leftarrow j_s, \text{cond}_p \leftarrow \text{cond}$ 
14:  Return  $\hat{\delta}$ 
15: end function
16: procedure init_specific()
17:   Initialise global variable  $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}) \leftarrow \emptyset$ 
18: end procedure
19: procedure process_region( $\mathcal{S}$ )
20:   Append  $\mathcal{S}$  to  $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta})$ 
21:    $\widetilde{\text{FWER}}(\hat{\delta}) \leftarrow \hat{\delta} |\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta})|$ 
22:   while  $\widetilde{\text{FWER}}(\hat{\delta}) > \alpha$  do
23:     Decrease  $\hat{\delta}$ 
24:      $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}) \leftarrow \{ \mathcal{S}' \in \widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}) \mid p_{\mathcal{S}', \min} \leq \hat{\delta} \}$ 
25:      $\widetilde{\text{FWER}}(\hat{\delta}) \leftarrow \hat{\delta} |\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta})|$ 
26: end procedure

```

our significant region mining algorithms use a breadth-first traversal strategy, as opposed to the depth-first traversal strategy used in Algorithm 3.2. The first part of Algorithm 6.2, its initialisation phase, comprises three actions:

- (i) In Line 2, the estimate $\hat{\delta}$ of the corrected significance threshold is set to 1, the largest value it could take.
- (ii) In Line 3, the routine `init_specific()` is called to initialise the variables specifically related to Tarone's method. In this case, this includes only the data structure used to represent the set $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta})$ of testable genomic regions at level $\hat{\delta}$. We kindly refer the reader to Section 3.5 for a description of this data structure and how it can be used to keep the minimal amount of information that Algorithm 6.2 needs without having to explicitly store all testable regions in memory.

- (iii) Finally, a “first in, first out” queue that will contain candidate genomic regions pending to be visited, denoted `region_queue`, is created in Line 4. After Line 5 is executed, `region_queue` will be populated by all genomic regions of length one, that is, by the set $\{\llbracket j_s, j_s \rrbracket \mid 1 \leq j_s \leq p\}$ of genomic regions which correspond to single markers. Moreover, these are appended to the queue sorted by their position j_s in ascending order. Together, Lines 4 and 5 initialise the breadth-first pattern enumeration procedure.

Once the initialisation phase is completed, Algorithm 6.2 begins its iterative, pattern enumeration phase. As shown in Line 6, Algorithm 6.2 continues its execution while `region_queue` contains genomic regions pending to be visited. Due to the way `region_queue` is initialised, containing all genomic regions of length one sorted by their starting position in increasing order, and the manner in which new candidate genomic regions will be recursively appended to the queue, Algorithm 6.2 guarantees that genomic regions will be enumerated firstly in ascending order of length and then of starting position j_s . Genomic regions are processed one at a time, always picking the region currently at the head of `region_queue` (Line 7). Given an arbitrary region $\mathcal{S} = \llbracket j_s, j_e \rrbracket$ currently being processed, the first step is to compute its minimum attainable P-value $p_{\mathcal{S}, \min}$ according to the test statistic of choice, as described in Sections 3.3 (Pearson’s χ^2 test and Fisher’s exact test) and 5.3 (CMH test). If region \mathcal{S} is testable at level $\hat{\delta}$, then the routine `process_region` is invoked in Line 9 to carry out all actions specifically related to the correction for multiple comparisons. In this particular case, Algorithm 6.2 utilises Tarone’s improved Bonferroni correction and, as a result, the steps taken by `process_region` are entirely identical to Lines 10-15 of Algorithm 3.2. The last four lines of Algorithm 6.2 are responsible for appending new candidate genomic regions to `region_queue`. In Line 10, the search space pruning condition is assessed for the genomic region currently being processed. The specific implementation of the pruning condition depends on the corresponding test statistic of choice. FAIS, which uses Pearson’s χ^2 test or Fisher’s exact test, employs the pruning condition described in Section 3.4 whereas FastCMH, which makes use of the CMH test, evaluates the pruning condition using Algorithm 5.1. A genomic region $\llbracket j_s, j_e \rrbracket$ of length $j_e - j_s + 1$ is a subset of exactly two genomic regions of length $j_e - j_s + 2$, namely, $\llbracket j_s - 1, j_e \rrbracket$ and $\llbracket j_s, j_e + 1 \rrbracket$. The only exceptions are genomic regions for which $j_s = 1$, which are only contained in $\llbracket j_s, j_e + 1 \rrbracket$, and genomic regions for which $j_e = p$, which are only contained in $\llbracket j_s - 1, j_e \rrbracket$. As a result, if the pruning condition evaluates to true for region $\llbracket j_s, j_e \rrbracket$, there are either one or two genomic regions in the next level of the pattern enumeration tree which can be pruned from the search space. Our significant region mining algorithms implement this as part of the breadth-first enumeration strategy as follows. Since genomic regions are enumerated in increasing order of length and starting position, when processing genomic region $\llbracket j_s, j_e \rrbracket$, the outcome of the pruning condition for the preceding region $\llbracket j_s - 1, j_e - 1 \rrbracket$ is already known. Thus, all information needed to decide whether region $\llbracket j_s - 1, j_e \rrbracket$ in the next level of the pattern enumeration tree should be added to `region_queue` as a candidate region or rather be pruned from the search space along all its descendants is readily available. In particular, three possible cases can be distinguished:

- (i) If $j_s = 1$, there is simply no region $\llbracket j_s - 1, j_e \rrbracket$ to be considered.

- (ii) If the preceding region $\llbracket j_s - 1, j_e - 1 \rrbracket$ was never enumerated, the pruning condition must have evaluated to true for some region $S' \subset \llbracket j_s - 1, j_e - 1 \rrbracket$ located in a former level of the pattern enumeration tree. Therefore, the pruning condition must evaluate to true for region $\llbracket j_s - 1, j_e - 1 \rrbracket$ as well and, consequently, $\llbracket j_s - 1, j_e \rrbracket$ and all its descendants can be pruned from the search space. We detect this particular situation by keeping track of the starting position $j_{s,p}$ of the genomic region processed immediately before the region currently being processed. If $j_s - j_{s,p} \neq 1$, then we know that region $\llbracket j_s - 1, j_e - 1 \rrbracket$ was never enumerated and, thus, $\llbracket j_s - 1, j_e \rrbracket$ should *not* be appended to `region_queue`.
- (iii) If the preceding region $\llbracket j_s - 1, j_e - 1 \rrbracket$ was enumerated, then region $\llbracket j_s - 1, j_e \rrbracket$ can be pruned from the search space if either the pruning condition for region $\llbracket j_s - 1, j_e - 1 \rrbracket$ evaluated to true, as indicated by the variable `cond_p`, or the pruning condition for the region $\llbracket j_s, j_e \rrbracket$ currently being processed evaluated to true, as indicated by the variable `cond`.

All these conditions are checked in Line 11¹, leading to region $\llbracket j_s - 1, j_e \rrbracket$ being appended to `region_queue` if necessary in Line 12. Finally, the variables $j_{s,p}$ and `cond_p` are updated in Line 13 to be made available for the genomic region being processed in the next iteration, if any. Owing to condition (ii) above, if a genomic region is not appended to `region_queue`, none of its descendants will possibly be appended either. Thus, the number of genomic regions contained in `region_queue` progressively decreases during the execution of Algorithm 6.2 until the queue becomes empty, indicating that all genomic regions in the search space either have been already visited or have been pruned from the search space. At that point, the algorithm has converged and the estimate $\hat{\delta}$ of Tarone's corrected significance threshold equals its exact value. Hence, Algorithm 6.2 terminates by returning $\hat{\delta}$ in Line 14.

Once the corrected significance threshold δ has been obtained, the next step in our significant region mining algorithms is to retrieve all genomic regions which are significantly associated at level δ . This task is accomplished by the routine `retrieve_significant_regions`, summarised in Algorithm 6.3. Following the discussion in Section 3.5, `retrieve_significant_regions` traverses the pattern enumeration tree a second time from scratch. The way in which Algorithm 6.3 enumerates genomic regions is virtually identical to that of Algorithm 6.2. Nevertheless, Algorithm 6.3 uses the exact value δ of the corrected significance threshold from the start of the execution of the algorithm rather than a progressively decreasing estimate $\hat{\delta}$. Compared to Algorithm 6.2, this renders search space pruning more effective during early iterations of the pattern enumeration process. The other relevant distinction between Algorithm 6.3 and Algorithm 6.2 lies in the actions taken when an enumerated region is found to be testable. Algorithm 6.2 aims to compute the corrected significance threshold δ and, as a consequence, utilises the newly found testable region to adjust the current estimate $\hat{\delta}$ of δ , if necessary. In contrast, Algorithm 6.3 aims to retrieve significantly associated regions. Thus, in Line 9 of the algorithm, the P-value p_S of the (testable) region currently being processed is computed according to the test statistic of choice. If p_S is smaller than δ , the region is significantly associated and is

1. Lazy evaluation was assumed while writing the pseudocode to avoid the need to initialise $j_{s,p}$ and `cond_p` before the first genomic region is processed by the algorithm.

Algorithm 6.3 retrieve_significant_regions

Input: Dataset \mathcal{D} , corrected significance threshold δ
Output: Set of significantly associated genomic regions $\mathcal{M}_{\text{sig,raw}} = \{\mathcal{S} \in \mathcal{M} \mid p_{\mathcal{S}} \leq \delta\}$

```

1: function retrieve_significant_regions( $\mathcal{D}, \alpha$ )
2:   Initialise  $\mathcal{M}_{\text{sig,raw}} \leftarrow \emptyset$ 
3:   Create empty queue region_queue
4:   for  $j_s = 1, 2, \dots, p$  do region_queue.enqueue ( $\llbracket j_s, j_s \rrbracket$ )
5:   while region_queue is not empty do
6:      $\mathcal{S} \leftarrow$  region_queue.dequeue()  $\triangleright \mathcal{S} = \llbracket j_s, j_e \rrbracket$ 
7:     Compute the minimum attainable P-value  $p_{\mathcal{S},\min}$   $\triangleright$  see Sections 3.3 and 5.3
8:     if  $p_{\mathcal{S},\min} \leq \delta$  then
9:       Compute the P-value  $p_{\mathcal{S}}$   $\triangleright$  see Sections 2.2 and 5.2
10:      if  $p_{\mathcal{S}} \leq \delta$  then append  $\mathcal{S}$  to  $\mathcal{M}_{\text{sig,raw}}$ 
11:       $\text{cond} \leftarrow$  pruning_condition( $\mathcal{S}, \delta$ )  $\triangleright$  see Sections 3.4 and 5.4
12:      if  $j_s \neq 1$  and  $j_s - j_{s,p} = 1$  and not ( $\text{cond}_p$  or  $\text{cond}$ ) then
13:        region_queue.enqueue ( $\llbracket j_s - 1, j_e \rrbracket$ )
14:       $j_{s,p} \leftarrow j_s, \text{cond}_p \leftarrow \text{cond}$ 
15:   Return  $\mathcal{M}_{\text{sig,raw}}$ 
16: end function

```

appended to the set $\mathcal{M}_{\text{sig,raw}}$ of significantly associated regions at level δ in Line 10. Once the enumeration process is completed, the algorithm ends by returning $\mathcal{M}_{\text{sig,raw}}$ in Line 15.

The final step in our significant region mining algorithms is to cluster the significantly associated genomic regions in $\mathcal{M}_{\text{sig,raw}}$, forming a set $\mathcal{M}_{\text{sig,clustered}}$ of disjoint clusters, each composed of possibly many overlapping genomic regions. This is carried out by the routine `cluster_overlapping_regions`, summarised in Algorithm 6.4. Since $\mathcal{M}_{\text{sig,raw}}$ is most often a small subset of the set $\mathcal{M}_{\text{test}}(\delta)$ of testable genomic regions, which itself tends to be a small subset of the search space \mathcal{M} of all candidate genomic regions, the computational overhead of this final step is negligible in practice. In Line 2, Algorithm 6.4 begins by creating an empty set of clusters $\mathcal{M}_{\text{sig,clustered}}$, which will eventually become the output of the algorithm. In the next line, the algorithm considers the trivial case in which no significantly associated regions were retrieved by Algorithm 6.3, terminating early if this situation occurred. Under the assumption that $\mathcal{M}_{\text{sig,raw}}$ is not empty, Algorithm 6.4 next sorts all significantly associated genomic regions by starting position in ascending order, as shown in Line 4². After sorting all significantly associated regions, in Line 5 Algorithm 6.4 creates an empty cluster \mathcal{C}_{sig} to which the first significantly associated regions will be added. By construction, each cluster consists of overlapping genomic regions, implying that the union $\cup_{\mathcal{S} \in \mathcal{C}_{\text{sig}}} \mathcal{S}$ of all genomic regions in the cluster is itself a genomic region $\llbracket c_s, c_e \rrbracket$, where c_s and c_e are the starting and ending positions of the cluster, respectively. As shown in

2. This is the most computationally intensive part of Algorithm 6.4, with average complexity $O(|\mathcal{M}_{\text{sig,raw}}| \log |\mathcal{M}_{\text{sig,raw}}|)$. Nevertheless, as mentioned previously, since $|\mathcal{M}_{\text{sig,raw}}|$ is often much smaller than $|\mathcal{M}|$, the runtime of Algorithm 6.4 tends to be negligible compared to that of Algorithms 6.2 and 6.3.

Algorithm 6.4 cluster_overlapping_regions**Input:** Set of significantly associated genomic regions $\mathcal{M}_{\text{sig,raw}}$ **Output:** Set of non-overlapping clusters of overlapping significantly associated genomic regions $\mathcal{M}_{\text{sig,clustered}}$

```

1: function cluster_overlapping_regions( $\mathcal{M}_{\text{sig,raw}}$ )
2:   Initialise  $\mathcal{M}_{\text{sig,clustered}} \leftarrow \emptyset$ 
3:   if  $\mathcal{M}_{\text{sig,raw}}$  is empty then return  $\mathcal{M}_{\text{sig,clustered}}$ 
4:   Sort  $\mathcal{M}_{\text{sig,raw}}$  by starting position in ascending order
5:   Create empty cluster  $\mathcal{C}_{\text{sig}}$  of overlapping regions  $\triangleright \cup_{\mathcal{S} \in \mathcal{C}_{\text{sig}}} \mathcal{S} = \llbracket c_s, c_e \rrbracket$ 
6:   for  $\mathcal{S} \in \mathcal{M}_{\text{sig,raw}}$  do  $\triangleright$  In ascending order of starting position
7:     if  $\mathcal{C}_{\text{sig}}$  is empty or  $j_s \leq c_e + 1$  then
8:       Append  $\mathcal{S}$  to  $\mathcal{C}_{\text{sig}}$  and update  $c_s$  and  $c_e$ 
9:     else
10:      Append current cluster  $\mathcal{C}_{\text{sig}}$  to  $\mathcal{M}_{\text{sig,clustered}}$ 
11:      Create new cluster  $\mathcal{C}_{\text{sig}}$  containing  $\mathcal{S}$ 
12:   Append last cluster  $\mathcal{C}_{\text{sig}}$  to  $\mathcal{M}_{\text{sig,clustered}}$ 
13:   Return  $\mathcal{M}_{\text{sig,clustered}}$ 
14: end function

```

Line 6, Algorithm 6.4 processes one region at a time, proceeding in ascending order of starting position. For each region \mathcal{S} , we are confronted with two possibilities, managed between Lines 7 and 11. If the starting position j_s of \mathcal{S} is smaller or equal than the ending position c_e of the current cluster plus one, then \mathcal{S} either overlaps or is adjacent to the current cluster \mathcal{C}_{sig} and should thus be incorporated to this cluster. In contrast, if that were not the case, not only does region \mathcal{S} not belong to the current cluster \mathcal{C}_{sig} , but no other region not yet processed will belong to it either. This is a direct consequence of the fact that regions are processed in ascending order of starting position. As a result, when this occurs, the current cluster \mathcal{C}_{sig} is added to $\mathcal{M}_{\text{sig,clustered}}$ and a new cluster \mathcal{C}_{sig} is created to contain region \mathcal{S} . Finally, once the algorithm has processed all significantly associated regions, the last cluster \mathcal{C}_{sig} is added to $\mathcal{M}_{\text{sig,clustered}}$ in Line 12, which is then returned as output of the algorithm in the last line. The region clusters in $\mathcal{M}_{\text{sig,clustered}}$ provide an alternative, easier-to-interpret representation of the findings made by the algorithm than the often heavily redundant set of all significantly associated genomic regions $\mathcal{M}_{\text{sig,raw}}$. In particular, for each cluster \mathcal{C}_{sig} in $\mathcal{M}_{\text{sig,clustered}}$, there are at least two different ways to summarise the set of overlapping significantly associated genomic regions belonging to \mathcal{C}_{sig} . One possibility is to provide the region $\mathcal{S}^* \in \mathcal{C}_{\text{sig}}$ with the smallest (i.e. most significant) P-value, that is, $\mathcal{S}^* = \underset{\mathcal{S} \in \mathcal{C}_{\text{sig}}}{\operatorname{argmin}} p_{\mathcal{S}}$. The other option is to report the genomic region $\llbracket c_s, c_e \rrbracket$ corresponding to the entire cluster, which is equivalent to reporting the union of all regions belonging to the cluster. Both approaches are complementary in the following way: the former can often slightly underestimate the length of the truly associated genomic region, missing a small number of markers, whereas the latter can slightly overestimate it, including a few additional, noisy markers at the extremes. Since the output of our algorithms is meant to be used by practitioners as part of a more

thorough analysis, possibly taking into consideration existing prior knowledge or performing follow-up experiments to validate the findings, we decided to provide the user with both forms of summarisation as the output of our significant region mining algorithms, which also includes the total number $|\mathcal{C}_{\text{sig}}|$ of significantly associated overlapping genomic regions included in each cluster.

Extension to permutation testing-based significant pattern mining

Algorithm 6.5 Permutation testing-based procedures for Algorithm 6.2

Input: Number of permutations j_p

- 1: **procedure** `init_specific_wylight_marginal()`
- 2: **for** $h = 1, 2, \dots, j_p$ **do**
- 3: Obtain a random permutation $\pi^{(h)} : \llbracket 1, n \rrbracket \rightarrow \llbracket 1, n \rrbracket$
- 4: Initialise global variable $\tilde{p}_{\text{ms}}^{(h)} \leftarrow 1$
- 5: **end procedure**
- 6: **procedure** `init_specific_wylight_conditional()`
- 7: **for** $c = 1, 2, \dots, k$ **do**
- 8: $\mathcal{J}(c) \leftarrow \{i \in \llbracket 1, n \rrbracket \mid c_i = c\}$
- 9: **for** $h = 1, 2, \dots, j_p$ **do**
- 10: Obtain a random permutation $\pi_c^{(h)} : \mathcal{J}(c) \rightarrow \mathcal{J}(c)$
- 11: **for** $h = 1, 2, \dots, j_p$ **do**
- 12: Define $\pi^{(h)} : \llbracket 1, n \rrbracket \rightarrow \llbracket 1, n \rrbracket$ as $\pi^{(h)}(i) = \pi_{c_i}^{(h)}(i)$
- 13: Initialise global variable $\tilde{p}_{\text{ms}}^{(h)} \leftarrow 1$
- 14: **end procedure**
- 15: **procedure** `process_region_wylight(S)`
- 16: **for** $h = 1, 2, \dots, j_p$ **do**
- 17: Compute P-value $p_S^{(h)}$ for resampled dataset $\tilde{D}^{(h)}$
- 18: $\tilde{p}_{\text{ms}}^{(h)} \leftarrow \min(\tilde{p}_{\text{ms}}^{(h)}, p_S^{(h)})$
- 19: $\widetilde{\text{FWER}}(\hat{\delta}) \leftarrow \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[\tilde{p}_{\text{ms}}^{(h)} \leq \hat{\delta} \right]$
- 20: **while** $\widetilde{\text{FWER}}(\hat{\delta}) > \alpha$ **do**
- 21: Decrease $\hat{\delta}$
- 22: $\widetilde{\text{FWER}}(\hat{\delta}) \leftarrow \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[\tilde{p}_{\text{ms}}^{(h)} \leq \hat{\delta} \right]$
- 23: **end procedure**

For the sake of clarity, this chapter has been written under the assumption that Tarone's improved Bonferroni correction is utilised to correct for multiple comparisons. Nevertheless, it is exceedingly simple to modify Algorithm 6.1 to incorporate the permutation testing-based approach proposed in Chapter 4. In particular, this can be accomplished by merely modifying the routines `init_specific` and `process_region` in Algorithm 6.2 whereas Algorithms 6.3 and 6.4 would remain intact.

The routine `init_specific` takes care of initialising all variables and data structures related to the correction for multiple comparisons. In permutation testing-based

significant pattern mining, this routine must initialise the estimate $\tilde{p}_{\text{ms}}^{(h)}$ of the most significant P-value for the h -th resampled dataset to 1 for each $h = 1, 2, \dots, j_p$, where j_p is the desired number of permutations. Moreover, `init_specific` must also obtain a permutation $\pi^{(h)} : \llbracket 1, n \rrbracket \rightarrow \llbracket 1, n \rrbracket$ used to define each resampled dataset $\tilde{D}^{(h)}$. However, the way in which this permutation is obtained differs slightly depending on whether there is a categorical covariate to correct for, as in `FastCMH`, or not, as in `FAIS`. The pseudocode corresponding to each of these situations is shown in Algorithm 6.5 under the names `init_specific_wylight_conditional` and `init_specific_wylight_marginal`, respectively. In the latter case, each of the j_p permutations is obtained as a random permutation $\pi^{(h)} : \llbracket 1, n \rrbracket \rightarrow \llbracket 1, n \rrbracket$, sampled uniformly from the set of all $n!$ possible permutations of $\llbracket 1, n \rrbracket$. However, as described in Section 5.5, when a categorical covariate is present the global null hypothesis changes, requiring a different approach to generate the permutations. Rather than obtaining a random permutation of $\llbracket 1, n \rrbracket$ directly, indices are only shuffled among samples having an identical value for the categorical covariate. That is, a random permutation $\pi_c^{(h)} : \mathcal{J}(c) \rightarrow \mathcal{J}(c)$ of the set $\mathcal{J}(c) = \{i \in \llbracket 1, n \rrbracket \mid c_i = c\}$ is obtained first, for each of the k categories of the covariate. Then, the permutation $\pi^{(h)} : \llbracket 1, n \rrbracket \rightarrow \llbracket 1, n \rrbracket$ is defined by “concatenating” these k “disjoint” permutations as $\pi^{(h)}(i) = \pi_{c_i}^{(h)}(i)$ for each sample $i = 1, 2, \dots, n$.

Finally, the routine `process_region` must also change to accommodate permutation testing-based significant pattern mining. In particular, as summarised in the procedure `process_region_wylight` in Algorithm 6.5, this can be accomplished by using the same steps as shown between Lines 13 and 19 of Algorithm 4.2.

In summary, combining significant region mining with permutation testing-based significant pattern mining poses no substantial algorithmic difficulties. Not unlike in other instances of significant pattern mining, doing so would allow to gain statistical power at the cost of increased runtime. In applications for which the signal-to-noise ratio is low and computational constraints are not too stringent, this option might thus be particularly appealing.

Extensions of the model of genetic heterogeneity

Perhaps the most important design choice in our formal introduction of significant region mining in Section 6.1 was to define the occurrence of a pattern (region) \mathcal{S} in an observation x as $g_{\mathcal{S}}(x) = \mathbb{1}[b_{\mathcal{S}}(x) > 0]$, where $b_{\mathcal{S}}(x) = \sum_{j \in \mathcal{S}} x(j)$ is the burden count for observation x in the genomic region \mathcal{S} . Conceptually, this model assumes that the presence of *one or more* copies of a minor allele in the region might be sufficient to affect its functionality. An obvious generalisation of this model would be to relax this assumption, introducing a minimum number b_{\min} of copies of a minor allele needed to see an effect, i.e. it would imply defining $g_{\mathcal{S}}(x) = \mathbb{1}[b_{\mathcal{S}}(x) \geq b_{\min}]$ instead. All algorithms proposed in this section could be applied in this setting without modifications. Nonetheless, a clear disadvantage of this generalisation is the introduction of b_{\min} as a hyperparameter that needs to be chosen *a priori*. Analogously to our discussion regarding burden tests in Section 6.1, attempting to select b_{\min} in a data-driven fashion would require an additional correction for multiple comparisons.

Nevertheless, it would be possible to modify Algorithms 6.2 and 6.3 to perform significant region mining with multiple candidate values for b_{\min} simultaneously. In particular, since $b_{\min,1} \geq b_{\min,2}$ implies that $\mathbb{1}[b_S(x) \geq b_{\min,1}] \leq \mathbb{1}[b_S(x) \geq b_{\min,2}]$, if the pruning condition evaluates to true for the largest value of b_{\min} among all candidates, it will also evaluate to true for all others. Thus, the pattern enumeration process in Algorithms 6.2 and 6.3 would be dictated by the largest value of b_{\min} , while regions could be processed for all choices of b_{\min} simultaneously during the execution of the algorithm. Finally, it is worth noting that test statistics corresponding to the same genomic region but using a distinct value for b_{\min} might be strongly correlated. As a result, using a permutation testing-based version of significant region mining to consider jointly the set of test statistics for all candidate values of the hyperparameter b_{\min} could be particularly beneficial in this setting.

6.3 EXPERIMENTS

In this section, we empirically evaluate the performance of our proposed significant region mining algorithms compared to several baseline approaches using synthetic data, for which the evaluation of statistical power and FWER is possible. We will also present a set of experiments on real-world data which comprise five *A. thaliana* datasets and one case/control study of COPD including two different ethnic groups. For the sake of conciseness, the experiments contained in this thesis are a strict subset of those described in the original articles upon which this chapter is based [51, 52]. For additional experiments, as well as a more detailed biological annotation and interpretation of our findings, we kindly refer the reader to the original publications and their corresponding supplementary materials.

6.3.1 Experimental setup

Compared to other instances of significant pattern mining, the relatively small size of the search space renders the specific choice of algorithm to enumerate patterns considerably less critical. Moreover, while there exist itemset mining algorithms that traverse the pattern enumeration tree breadth-first, such as APRIORI [31], the problem of enumerating genomic regions is remarkably simpler due to the existence of a natural ordering of features in this setting. As a result, both of our proposed algorithms, FAIS and FastCMH, directly implement the pattern enumeration approach described in Algorithm 6.2 without resorting to using or modifying any established itemset mining algorithm from the data mining literature. FAIS, FastCMH and all baseline algorithms were written in C++ and compiled using gcc 4.8.2 with -O3 optimisation. All experiments were executed on a single thread of a 2.5 GHz Intel Xeon CPU with 64 GB of memory available. The test statistic used to quantify statistical association in all results reported for FAIS was Pearson’s χ^2 test. This choice will be emphasised by referring to FAIS as FAIS- χ^2 throughout the remainder of this section. Since the CMH test utilised by FastCMH can be understood as a generalisation of Pearson’s χ^2 test, this experimental setup provides a more fair comparison between FAIS and FastCMH than using Fisher’s exact test would have.

6.3.2 Simulation experiments

In this chapter, we have proposed significant region mining as a new paradigm for region-wise GWASs. This section presents a series of simulation experiments to investigate under which circumstances and to which extent our proposed algorithms outperform alternative approaches. In particular, we will explore empirically the following fundamental questions:

- (i) Under the assumption that genetic heterogeneity holds, how does the statistical power and FWER of FAIS- χ^2 and FastCMH compare to that of univariate testing? In other words, how effective is significant region mining in exploiting genetic heterogeneity?
- (ii) Does Tarone’s method continue to bring substantial improvements in statistical power and computational efficiency compared to a naive Bonferroni correction in this new instance of significant pattern mining, or does the smaller size of the search space render naive approaches competitive?
- (iii) Under a model of genetic heterogeneity, how does the statistical power of our significant region mining algorithms compare to traditional region-wise GWASs approaches such as burden tests?
- (iv) Can significant region mining lead to an improvement in statistical power under models other than genetic heterogeneity? More precisely, we compare the performance of significant region mining and univariate testing under the common assumption that multiple neighbouring variants are indirectly associated with the phenotype via linkage disequilibrium with a single unobserved, causal variant.

Statistical power and FWER

BASELINES: The first simulation experiment in this chapter aims to answer question (i) posed above, as well as the statistical considerations of question (ii). To this end, our proposed significant region mining algorithms, FAIS- χ^2 and FastCMH, will be compared against univariate testing approaches, which we denote Univariate- χ^2 and Univariate-CMH. These baselines simply perform a statistical association test for each of the p markers, making use of Pearson’s χ^2 test and the CMH test respectively. Thus, while they require a considerably less stringent correction for multiple comparisons than significant region mining algorithms, these methods are unable to exploit genetic heterogeneity to improve statistical power. Additionally, in order to probe the usefulness of Tarone’s improved Bonferroni correction in this new regime, we also include in our experiments two naive region-wise GWASs methods that enumerate and test all possible genomic regions using the Bonferroni correction to account for multiple comparisons, which we refer to as Bonf- χ^2 and Bonf-CMH.

DATA GENERATION: We generate synthetic GWAS datasets $\mathcal{D} = \{(x_i, y_i, c_i)\}_{i=1}^n$, containing n triplets (x_i, y_i, c_i) sampled i.i.d. from a distribution $p(x, y, c)$. Following Section 6.1, each observation x is a sequence of p count-valued genomic markers, i.e., $x_i(j) \in \llbracket 0, q \rrbracket$ for all $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, p$. As shown in Section 6.2, all approaches considered in this chapter only access these markers through $\mathbb{1}[x_i(j) > 0]$.

Therefore, we consider that $q = 1$ without loss of generality³. Finally, $y \in \{0, 1\}$ is a binary class label that simulates a case/control phenotype and $c \in \{0, 1\}$ a binary covariate.

The data-generating distribution $p(x, y, c)$ is closely related to that introduced in Section 5.6.1. Two non-overlapping genomic regions $\mathcal{S}_{\text{true}}$ and $\mathcal{S}_{\text{conf}}$, both of identical length l , are randomly selected to represent a truly associated and a confounded region, respectively. Thus, $G_{\mathcal{S}_{\text{true}}}(X) \not\perp Y \mid C$ will hold for $\mathcal{S}_{\text{true}}$ whereas the conditions $G_{\mathcal{S}_{\text{conf}}}(X) \not\perp Y$ and $G_{\mathcal{S}_{\text{conf}}}(X) \perp Y \mid C$ will hold for $\mathcal{S}_{\text{conf}}$. As in Section 5.6.1, this is accomplished by specifying a joint distribution $p(g_{\mathcal{S}_{\text{true}}}(x), g_{\mathcal{S}_{\text{conf}}}(x), y, c)$ factored as $p(g_{\mathcal{S}_{\text{true}}}(x), g_{\mathcal{S}_{\text{conf}}}(x), y, c) = p(g_{\mathcal{S}_{\text{true}}}(x), y, c) p(g_{\mathcal{S}_{\text{conf}}}(x) \mid c)$. The first term in the factorisation, $p(g_{\mathcal{S}_{\text{true}}}(x), y, c)$, is designed to have the following first and second order moments:

- (a) $\mathbb{E}(G_{\mathcal{S}_{\text{true}}}(X)) = p_{\text{true}}$, $\mathbb{E}(Y) = p_y$ and $\mathbb{E}(C) = p_c$, where we use $p_{\text{true}} = p_y = p_c = 0.5$ for simplicity.
- (b) $\text{Corr}(Y, G_{\mathcal{S}_{\text{true}}}(X)) = \rho_{\text{true}}$ and $\text{Corr}(Y, C) = \rho_{\text{conf}}$ whereas $G_{\mathcal{S}_{\text{true}}}(X)$ and C are uncorrelated. Moreover, we consider a balanced experimental setup where $\rho_{\text{true}} = \rho_{\text{conf}} = \frac{\rho}{2}$, with ρ being the proportion of variance in the class labels jointly explained by the truly associated region and the covariate.

The second term in the factorisation, $p(g_{\mathcal{S}_{\text{conf}}}(x) \mid c)$, is defined by setting $G_{\mathcal{S}_{\text{conf}}}(X) = C$ with probability $1 - \epsilon$ and $G_{\mathcal{S}_{\text{conf}}}(X) = 1 - C$ with probability ϵ . We used $\epsilon = 0.05$ throughout all experiments, leading to $G_{\mathcal{S}_{\text{conf}}}(X)$ and C being strongly positively correlated.

For each synthetic GWAS dataset \mathcal{D} to be generated, $\{(g_{\mathcal{S}_{\text{true}}}(x_i), y_i, c_i)\}_{i=1}^n$ were sampled first as n i.i.d. draws from a multivariate Bernoulli distribution $p(g_{\mathcal{S}_{\text{true}}}(x), y, c)$ with the first and second-order moments specified above using the method in [85]. Next, the corresponding set $\{g_{\mathcal{S}_{\text{conf}}}(x_i)\}_{i=1}^n$ of realisations of the pattern occurrence indicator for the confounded genomic region is obtained as the output of a binary symmetric channel with error rate ϵ and input $\{c_i\}_{i=1}^n$. Finally, a set of n i.i.d. genomic sequences $\{x_i\}_{i=1}^n$ consistent with the outcomes of $\{g_{\mathcal{S}_{\text{true}}}(x_i)\}_{i=1}^n$ and $\{g_{\mathcal{S}_{\text{conf}}}(x_i)\}_{i=1}^n$ must be generated. The approach we follow here also closely parallels the reasoning used in Section 5.6.1:

- (i) All ‘‘background’’ markers $\{x(j) \mid j \in \llbracket 1, p \rrbracket \setminus (\mathcal{S}_{\text{true}} \cup \mathcal{S}_{\text{conf}})\}$ are sampled as i.i.d. draws from a Bernoulli distribution, i.e. $p(x(j)) = \text{Bernoulli}(x(j) \mid p_{x,b})$. In all our experiments, we set $p_{x,b} = 0.3$ as a rough approximation of the typical levels of sparsity encountered in real-world GWAS datasets.
- (ii) The markers $\{x(j) \mid j \in \mathcal{S}_{\text{true}}\}$ that belong to the truly associated region $\mathcal{S}_{\text{true}}$ must satisfy $x_i(j) = 0$ for all samples $i = 1, 2, \dots, n$ for which $g_{\mathcal{S}_{\text{true}}}(x_i) = 0$, since $g_{\mathcal{S}_{\text{true}}}(x_i)$ is defined as $g_{\mathcal{S}_{\text{true}}}(x_i) = \mathbb{1} \left[\sum_{j \in \mathcal{S}_{\text{true}}} x_i(j) > 0 \right]$. Similarly, for all samples $i = 1, 2, \dots, n$ for which $g_{\mathcal{S}_{\text{true}}}(x_i) = 1$, there must be at least one $j^*(i) \in \mathcal{S}_{\text{true}}$ such that $x_i(j^*) = 1$. Thus, for each $i = 1, 2, \dots, n$ satisfying $g_{\mathcal{S}_{\text{true}}}(x_i) = 1$, we sample a position $j^*(i)$ uniformly at random from $\mathcal{S}_{\text{true}}$ and set $x_i(j^*) = 1$. The

3. Since both univariate baselines, Univariate- χ^2 and Univariate-CMH, are based on tests of association between binary variables, this assumption does not provide any advantage for the region-wise tests.

remaining markers $\{x_i(j) \mid j \in \mathcal{S}_{\text{true}} \setminus \{j^*(i)\}\}$ have no effect on $g_{\mathcal{S}_{\text{true}}}(x_i)$ and can therefore take any value. We choose to set $x_i(j) = 0$ for all $j \in \mathcal{S}_{\text{true}} \setminus \{j^*(i)\}$ to emphasise the assumption of genetic heterogeneity.

- (iii) The markers $\{x(j) \mid j \in \mathcal{S}_{\text{conf}}\}$ which form the confounded region $\mathcal{S}_{\text{conf}}$ are sampled analogously as in (ii).

METRICS: For all of the region-wise GWAS approaches, including FAIS- χ^2 and FastCMH, we will consider that the truly significant region $\mathcal{S}_{\text{true}}$ has been successfully detected in a repetition of the experiment if it is contained within a cluster of overlapping, significantly associated genomic regions, i.e., if there exists \mathcal{C}_{sig} in $\mathcal{M}_{\text{sig,clustered}}$ such that $\mathcal{S}_{\text{true}} \subseteq \cup_{\mathcal{S} \in \mathcal{C}_{\text{sig}}} \mathcal{S}$. To prevent trivial solutions such as reporting the entire genome as a cluster in $\mathcal{M}_{\text{sig,clustered}}$ from obtaining good scores under our metrics, we will consider any cluster \mathcal{C}_{sig} in $\mathcal{M}_{\text{sig,clustered}}$ to be a false positive if more than half of the markers it spans do not belong to $\mathcal{S}_{\text{true}}$, i.e., if $|\cup_{\mathcal{S} \in \mathcal{C}_{\text{sig}}} \mathcal{S} \setminus \mathcal{S}_{\text{true}}| \geq \frac{1}{2} |\cup_{\mathcal{S} \in \mathcal{C}_{\text{sig}}} \mathcal{S}|$. Using this definition, approaches which appear to achieve satisfactory statistical power by means of substantially overestimating the length of the truly associated region will be severely penalised in terms of the FWER. Finally, in order to evaluate the performance of the univariate testing methods, we consider that the truly associated region has been detected if at least one of the markers in the region $\mathcal{S}_{\text{true}}$ is deemed statistically significant. Markers deemed significant but which do not belong to $\mathcal{S}_{\text{true}}$ are treated as false positives. All in all, this way of assessing the performance of the univariate baselines can be considered rather lenient, as unlike the region-wise GWAS approaches, univariate testing methods ought to identify only one marker in the region.

RESULTS: In this first simulation experiment, we generated synthetic GWAS datasets as described above, with $n = 500$ samples, $p = 1,000,000$ markers and truly associated and confounded regions of length $l = 5$ markers each. The target FWER was set to $\alpha = 0.05$. All results shown were obtained by averaging across 500 repetitions of the experiment.

In Figure 6.2 we report how the resulting statistical power varies as a function of the signal strength ρ for all methods under consideration. These results convey two particularly striking observations. Firstly, while it is unsurprising that the region-wise GWAS approaches outperform the univariate baselines, since the data was generated following precisely a model of genetic heterogeneity, the extent by which their resulting statistical power differs is remarkable. Even in the high signal-to-noise ratio regime, with ρ close to one, both univariate baselines fail to reliably detect the truly associated region. In contrast, both significant region mining algorithms, FAIS- χ^2 and FastCMH, are able to consistently retrieve the region for moderate-to-high signal strength. This outcome illustrates how much a strong signal can be diluted among multiple neighbouring markers under a model of genetic heterogeneity, emphasising the usefulness of developing algorithms able to exploit this phenomenon. The second remarkable finding shown in Figure 6.2 is the confirmation that Tarone's testability criterion continues to be an instrumental part of significant region mining. Both baselines based on a naive Bonferroni correction, Bonf- χ^2 and Bonf-CMH, perform considerably more poorly than FAIS- χ^2 and FastCMH. The gap in statistical power is

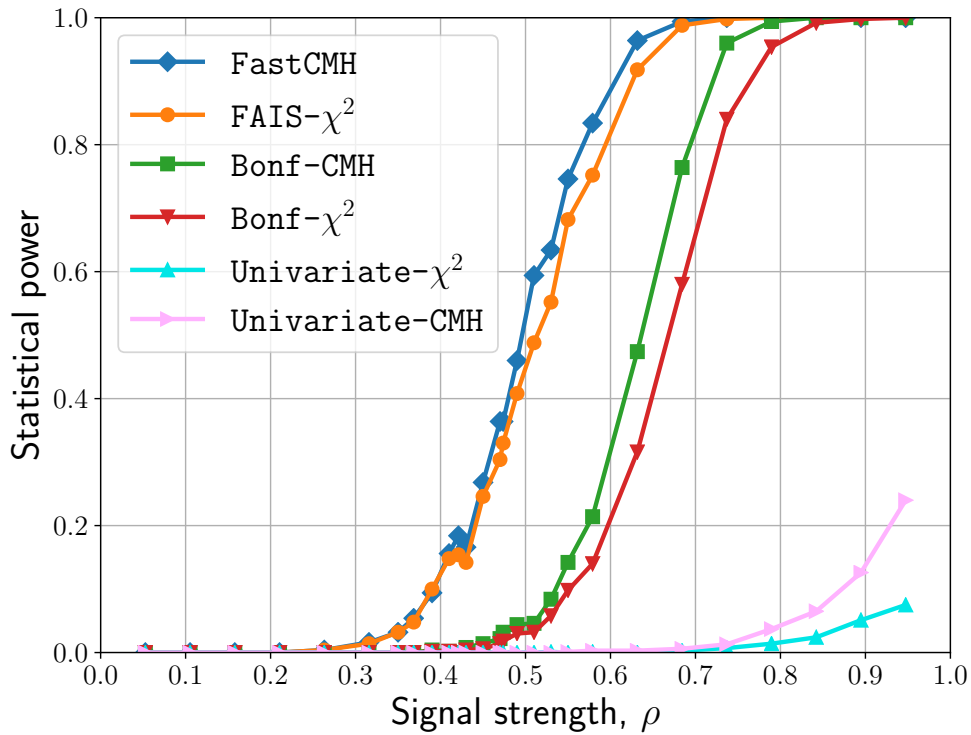


Figure 6.2. – Statistical power at different values of the signal strength ρ for our proposed approaches FAIS- χ^2 and FastCMH, as well as four baseline algorithms: Bonf- χ^2 , Bonf-CMH, Univariate- χ^2 and Univariate-CMH.

particularly pronounced for moderate values of the signal strength ρ , a situation of particular interest for biomarker discovery.

To complement the results depicted in Figure 6.2, the FWER of each method as a function of the signal strength ρ is shown in Figure 6.3. These additional results confirm the behaviour to be expected from each approach. All of the methods which use the CMH test to account for the categorical covariate are able to control the FWER at the desired level α , as guaranteed in theory. In contrast, the methods based on Pearson’s χ^2 test are by design unable to distinguish the truly associated region and the confounded region. As a consequence, they no longer control the FWER once the signal strength ρ is sufficiently large for the confounded region to be detected often. Therefore, in situations where confounding due to covariate factors is a possibility, using FAIS- χ^2 should be avoided in favour of FastCMH.

Runtime

BASELINES: The second set of simulation experiments in this chapter deals with the computational considerations of question (ii), that is, it aims to evaluate the impact of Tarone’s method in the computational efficiency of significant region mining. To this end, we compared our proposed approaches FAIS- χ^2 and FastCMH with the baselines Bonf- χ^2 and Bonf-CMH introduced in the previous set of experiments. Additionally,

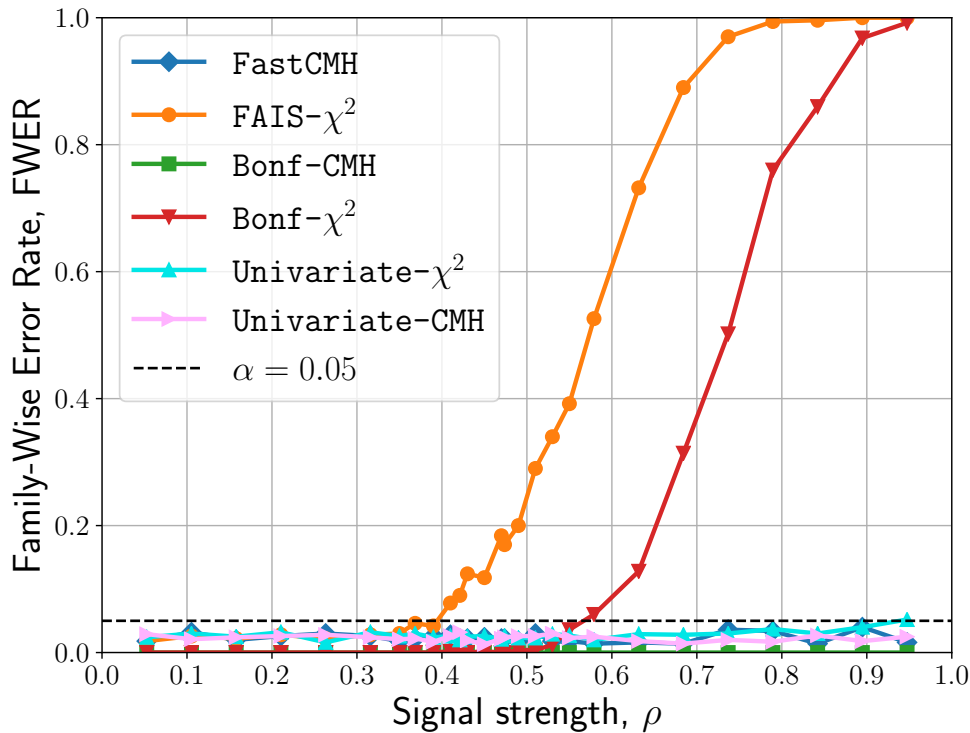


Figure 6.3. – FWER as a function of the signal strength ρ for our proposed approaches FAIS- χ^2 and FastCMH, as well as four baseline algorithms: Bonf- χ^2 , Bonf-CMH, Univariate- χ^2 and Univariate-CMH.

to assess the repercussions of the way in which the pruning criterion for the CMH test is evaluated in significant region mining, we compare FastCMH with 2^k -FastCMH, a baseline which does *not* use Algorithm 5.1. Instead, 2^k -FastCMH evaluates the pruning criterion by computing 2^k values of the minimum attainable P-value function in the same manner as the baseline algorithm 2^k -FACS introduced in Section 5.6.1.

DATA GENERATION: We generated synthetic GWAS datasets following the same approach as described for the previous set of experiments. However, in this case, all of the markers in the genomic sequences were “background” markers, that is, there was neither a truly associated region nor a confounded region in any of the datasets. We found this simplification to have no observable effect on the resulting computational efficiency of any of the approaches included in the experiments.

METRICS: The computational efficiency of all approaches under consideration will be assessed in terms of absolute runtime. Despite the reliance of this criterion on low-level implementation choices and the specific hardware of the system all experiments were executed in, all of these factors were identical for all methods included in the experiments. Thus, relative differences in runtime can be meaningfully interpreted to provide an evaluation of the computational efficiency of each algorithm.

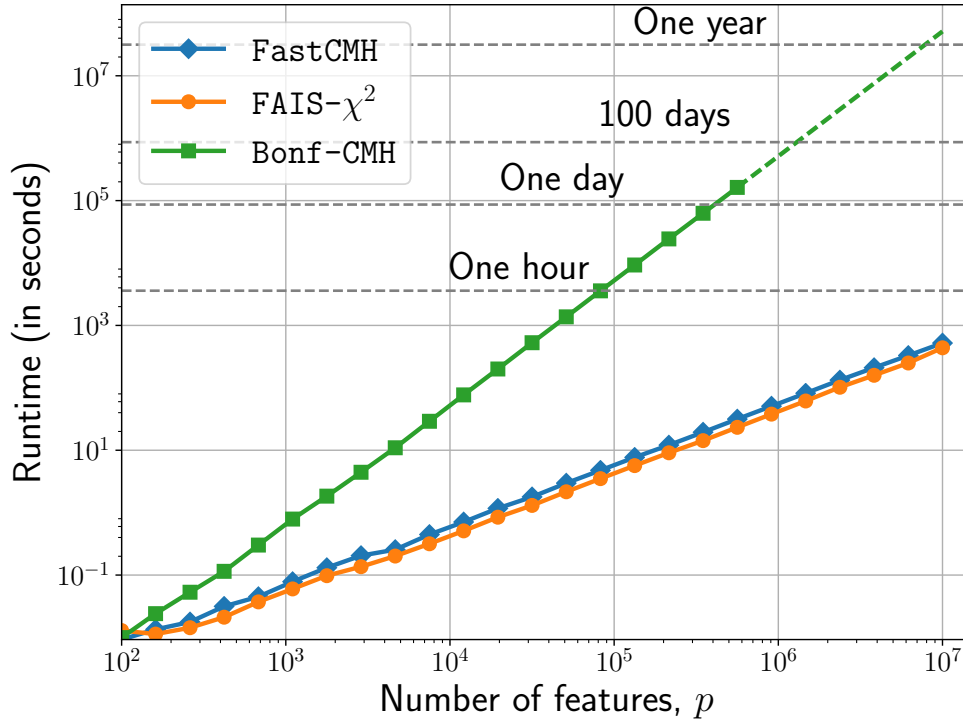


Figure 6.4. – Runtime as a function of the number of features p for our proposed approaches FAIS- χ^2 and FastCMH, as well as the baseline Bonf-CMH. Results for Bonf- χ^2 are virtually identical to those of Bonf-CMH and have been omitted from the figure for the sake of clarity. The discontinuous trace for Bonf-CMH corresponds to forecasts made using a trend model of the form $\log_{10} T = b + \eta \log_{10} p$ rather than values actually measured via experiments.

RESULTS: Figure 6.4 shows how the total runtime of each method varies as a function of the number of markers p . In this experiment the sample size n was set to 500 whereas the number of categories for the covariate c was $k = 4$. The results depicted in Figure 6.4 clearly indicate that Tarone’s method remains an essential component of our significant region mining algorithms. Empirically, the runtime of FAIS- χ^2 and FastCMH appears to increase linearly with the number of markers p . In contrast, the baselines based on the Bonferroni correction are unable to prune the search space and must test all $\frac{p(p+1)}{2} = O(p^2)$ regions. As a result, their runtime increases quadratically with p . In particular, for sufficiently large p , this can mean the difference between the analysis being computationally feasible or not. Finally, it is worth noting that the runtime overhead of FastCMH compared to FAIS- χ^2 which can be observed in Figure 6.4 is negligible.

Figure 6.5 in turn examines the effect of the sample size n on the total runtime. In this experiment the number of markers p was set to 100,000 while the number of categories for the covariate c remained at $k = 4$. According to these results, while

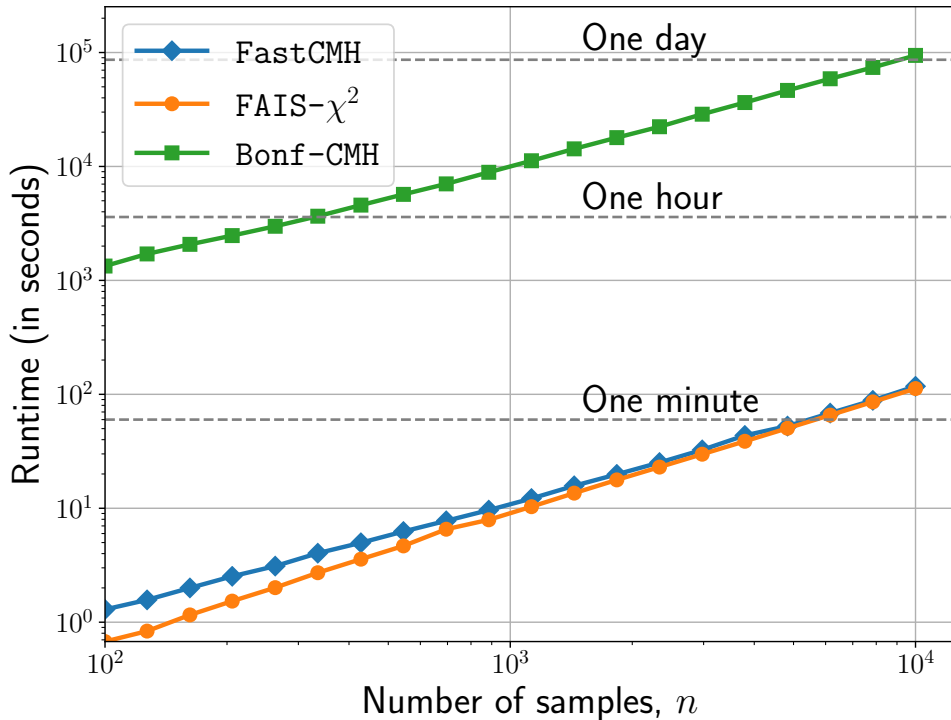


Figure 6.5. – Runtime as a function of the number of samples n for our proposed approaches FAIS- χ^2 and FastCMH, as well as the baseline Bonf-CMH. Results for Bonf- χ^2 are virtually identical to those of Bonf-CMH and have been omitted from the figure for the sake of clarity.

the Tarone-based algorithms are drastically faster, the runtime of all methods appears to scale approximately linearly with n . The runtime complexity of processing each enumerated genomic region is dominated by the computation of the support r_S , leading to $O(n)$ operations per region. Thus, the linear scaling exhibited in Figure 6.5 is unsurprising for the baseline approaches, which must enumerate and test all genomic regions. However, the fact that the runtime of FAIS- χ^2 and FastCMH also scales linearly with n in this experiment suggests that the number of testable regions was insensitive to changes in the sample size n . We hypothesise that this behaviour, which differs starkly from what can be observed in other instances of significant pattern mining, is mainly due to the distinctive characteristics of the search space in significant region mining, namely, the relatively small number of patterns it contains and the fact that regions of small length are more abundant than large regions.

Finally, Figure 6.6 studies the impact of the method used to evaluate the pruning condition on the approaches which use the CMH test. To this end, we generated synthetic GWAS datasets with $n = 500$ samples and $p = 100,000$ markers while varying the number of categories for the covariate k in the range $\llbracket 1, 30 \rrbracket$. These results reaffirm the findings reported in Chapter 5 in the context of significant itemset mining: the use of Algorithm 5.1 is a fundamental part of significant pattern mining

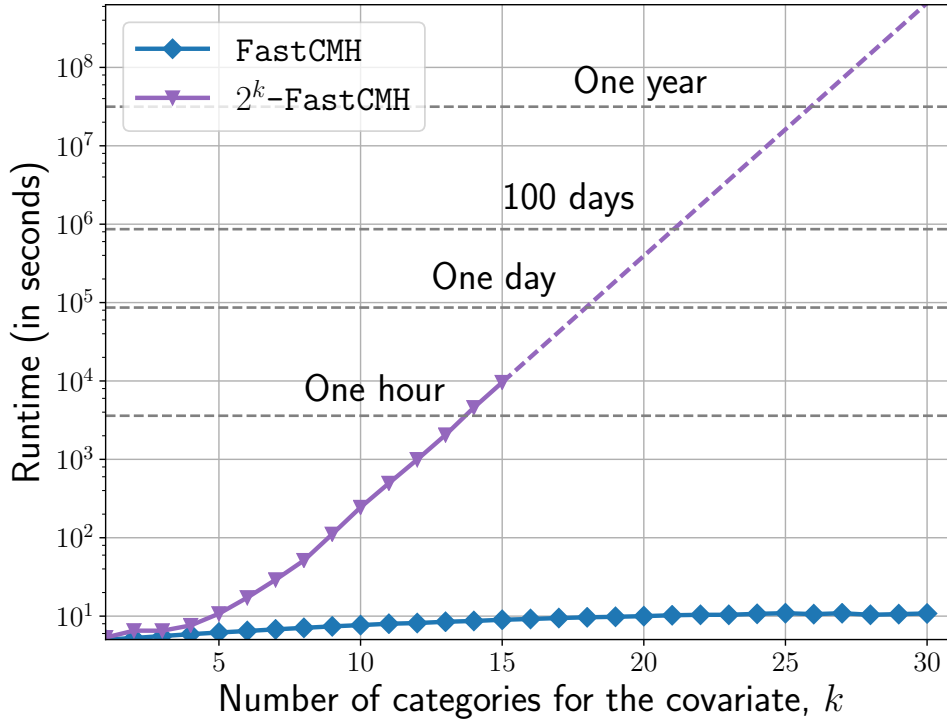


Figure 6.6. – Runtime as a function of the number of categories for the covariate k for our proposed approach FastCMH and the baseline 2^k -FastCMH. The discontinuous trace for 2^k -FastCMH corresponds to forecasts made using a trend model of the form $\log_{10} T = b + \eta k$ rather than values actually measured via experiments.

algorithms based on the CMH test, also in significant region mining. Without the help of Algorithm 5.1, the runtime of 2^k -FastCMH grows exponentially with k , quickly rendering the analysis unfeasible. Indeed, with as few as $k = 26$ categories, almost a year’s worth of computations would be required to complete a task that FastCMH completed in barely 10 seconds.

Comparison with burden tests

BASELINES: The next set of experiments deals with question (iii) posed at the beginning of this section, namely, comparing the statistical power of significant region mining with that of burden tests. Burden tests are a popular family of approaches to exploit genetic heterogeneity in GWASs, making them an ideal comparison partner for our proposed algorithms. As discussed in Section 6.1, the key difference between burden tests and significant region mining lies in the selection of the set of candidate genomic regions to be tested. Burden tests only consider regions of a certain length w fixed *a priori*, whereas significant region mining tests *all* genomic regions regardless of their length. Compared to burden tests, this paradigm shift frees significant region mining from the need to choose a value for w , making the resulting algorithms

robust to misspecification of this hyperparameter. However, this comes at the price of requiring a considerably more stringent corrected significance threshold to account for multiple comparisons. In short, the aim of the next experiment is to discern which of these two opposing forces has a stronger effect on the resulting statistical power, under the common assumption that the length of the truly associated regions is not known beforehand, nor necessarily constant throughout the genome.

To this end, we compare FastCMH⁴ with the two types of burden tests introduced in Section 6.1: burden tests with non-overlapping windows and burden tests with sliding windows. The former only consider regions of length w whose starting position is a multiple of w plus one whereas the latter include all regions of length w regardless of their starting position. Thus, burden tests with non-overlapping windows are not only sensitive to misspecification of the length w , but can also lose statistical power due to misalignments between the candidate regions in the search space and the (unknown) truly associated regions. However, they also require the least stringent correction for multiple comparisons across all methods under consideration. For each of the two types of burden tests, we use five different choices for the hyperparameter w , all of them within the range of potential lengths for the truly associated regions that will be included in the synthetic datasets. For each genomic region \mathcal{S} in their search space, the burden test baselines carry out an association test between the burden count $\sum_{j \in \mathcal{S}} X(j)$ and the class labels Y given the categorical covariate C using the LRT under a logistic regression model. More precisely, the null model only includes C as an explanatory variable while the alternative model includes both C and $\sum_{j \in \mathcal{S}} X(j)$. We also experimented with another version of the burden test baselines that made use of the CMH test to assess the association between $G_{\mathcal{S}}(X) = \mathbb{1} \left[\sum_{j \in \mathcal{S}} X(j) > 0 \right]$ and Y given C for each region \mathcal{S} to be considered. Under this alternative setting, FastCMH and the burden test baselines only differ in the definition of the search space. Thus, while this would perhaps constitute the most fair experimental setup for the comparison, we observed that the burden tests performed consistently better when the integer-valued burden count $\sum_{j \in \mathcal{S}} X(j)$ was used as an explanatory variable instead of the indicator of the presence of minor alleles in the region $G_{\mathcal{S}}(X)$. Consequently, despite being based on different explanatory variables, we chose to compare FastCMH against the best performing version of the burden test baselines.

DATA GENERATION: The generative model used to simulate synthetic GWAS datasets is almost identical to that used in the first set of experiments described in this section. However, there is one significant difference. In order to simulate variability in the length of the truly associated and confounded genomic regions, each of the two regions is replicated seven times with lengths l in the set $\mathcal{L} = \{2, 4, 6, 8, 10, 12, 14\}$. More precisely, once $\{g_{\mathcal{S}_{\text{true}}}(x_i)\}_{i=1}^n$ and $\{g_{\mathcal{S}_{\text{conf}}}(x_i)\}_{i=1}^n$ have been obtained as described above, we randomly choose seven non-overlapping genomic regions $\{\mathcal{S}_{\text{true},l}\}_{l \in \mathcal{L}}$ that will satisfy the conditions $\{g_{\mathcal{S}_{\text{true},l}}(x_i)\}_{i=1}^n = \{g_{\mathcal{S}_{\text{true}}}(x_i)\}_{i=1}^n$ and $|\mathcal{S}_{\text{true},l}| = l$ as well as other seven non-overlapping genomic regions $\{\mathcal{S}_{\text{conf},l}\}_{l \in \mathcal{L}}$ that will satisfy the conditions $\{g_{\mathcal{S}_{\text{conf},l}}(x_i)\}_{i=1}^n = \{g_{\mathcal{S}_{\text{conf}}}(x_i)\}_{i=1}^n$ and $|\mathcal{S}_{\text{conf},l}| = l$. In a nutshell, each of the

4. As shown in the two previous experiments, FAIS- χ^2 performs comparably to FastCMH in terms of statistical power while not providing the possibility to correct for covariates. Thus, for the remaining simulation experiments, we mainly focus on FastCMH.

replicated regions represents the same underlying signal, but scrambled across a different number of neighbouring markers l . For each region $\mathcal{S}_{\text{true},l}$ and $\mathcal{S}_{\text{conf},l}$, the markers $x_i(j)$ for $j \in \mathcal{S}_{\text{true},l}$ and $j \in \mathcal{S}_{\text{conf},l}$ are generated as described in the data generation section corresponding to the first set of simulation experiments. In particular, the sampling of $j^*(i)$ for each of the replicated regions is performed independently of the others. Finally, it is worth noting that while both FastCMH and the burden test baseline with sliding windows are entirely insensitive to the choice of starting position for the truly associated and confounded regions, this is not the case for the burden test baseline with non-overlapping windows. While perhaps the most fair approach would have been to select the starting position of each region completely at random, we decided to compare FastCMH against the most favourable scenario for the burden test baselines. Consequently, we forced the starting position of all seven truly associated genomic regions and all seven confounded genomic regions to be randomly chosen among the set of starting positions of the regions in the search space of the burden test baseline with non-overlapping windows. In other words, their starting positions are selected at random from the set $\{1 + iw \mid 0 \leq i \leq \lfloor \frac{p}{w} \rfloor\}$. In particular, this biased experimental setup implies that the version of the burden test baselines using non-overlapping windows must necessarily outperform the alternative version using sliding windows. In summary, the appropriate way to interpret our experimental setup is as follows: we compare FastCMH against burden tests with sliding windows and against a *best-case scenario* for burden tests with non-overlapping windows.

METRICS: Statistical power for FastCMH is also assessed in an identical manner as in the first set of experiments: a truly associated region is considered to be detected if it is fully contained within a cluster \mathcal{C}_{sig} in $\mathcal{M}_{\text{sig,clustered}}$. However, we chose a more lenient way to evaluate statistical power for the burden test baselines. In particular, for some choices of w , it is possible that some of the truly associated regions $\mathcal{S}_{\text{true},l}$ are never fully contained in any of the genomic regions tested by the burden test. To avoid this from placing the baseline approaches at an unfair disadvantage, we relax the requirement for a truly associated region to be considered detected when evaluating the burden tests. As long as *any* region deemed significantly associated by the burden tests overlaps with a truly associated region, we consider the latter to have been successfully detected. Statistical power for a single repetition of the experiment is then defined as the proportion of truly associated regions, out of all seven replicates, retrieved by the algorithms under consideration.

RESULTS: We generated synthetic GWAS datasets as described above, using $n = 500$ samples and $p = 100,000$ markers for each dataset. Figure 6.7 shows the statistical power⁵ for FastCMH and all burden test baselines as a function of the signal strength ρ , measured by averaging across 200 repetitions of the experiment. The results are strikingly clear: FastCMH outperforms all burden test baselines by a large margin, regardless of the choice for the hyperparameter w . For each of the seven truly

5. Results regarding the FWER obtained in these experiments can be found in the original publication [52]. As already shown in the first set of experiments, these results empirically confirm that FastCMH is able to control the FWER at the desired level $\alpha = 0.05$.

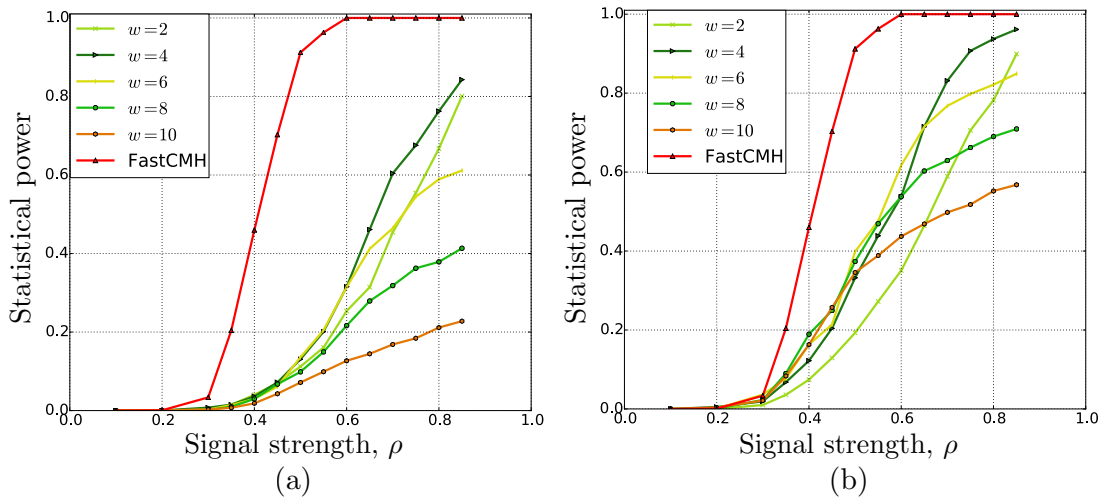


Figure 6.7. – A comparison of the statistical power of FastCMH and several burden tests with (a) sliding windows and (b) non-overlapping windows. Both types of burden tests were executed with five different choices for the hyperparameter w that controls the length of the regions to be tested.

associated regions $\mathcal{S}_{\text{true},l}$, all of the burden test baselines contain at least one region in their search space which includes some of the markers in $\mathcal{S}_{\text{true},l}$. On many occasions, multiple regions will satisfy that criterion. Nevertheless, the poor performance of burden tests in this experiment indicates that their statistical power deteriorates sharply as the mismatch between w and l increases. Since each synthetic GWAS dataset contains multiple truly associated regions with lengths ranging between 2 and 14, no single choice for w is able to consistently detect all seven truly associated regions. In contrast, FastCMH showcases the usefulness of defining an exhaustive search space when coupled with an efficient correction for multiple comparisons. By testing *all* genomic regions, it guarantees that the truly associated regions are always included in the search space rather than only regions that have a partial overlap with them. Our results show that, despite the need for a more stringent corrected significance threshold, this inherent trade-off leans clearly in favour of FastCMH. In summary, the results of this experiment strongly suggest that FastCMH holds the potential to substantially outperform burden tests in exploratory analyses for which reliable prior knowledge regarding the length of the associated regions is unavailable.

Statistical power under a model of linkage disequilibrium

BASELINES: The final set of simulation experiments concerns question (iv) posed at the beginning of this section. We compare the performance of significant region mining and univariate approaches under a model of linkage disequilibrium. In this model, we consider that a single unobserved causal variant harbours the association signal. However, we also assume that multiple observed neighbouring markers are correlated with the unobserved causal variant, thus being indirectly associated with the phenotype. From a statistical perspective, this model differs drastically from the assumption of genetic heterogeneity that motivated our proposed algorithms. In

particular, the different markers in the associated regions do *not* provide independent (weak) signals to be aggregated. Instead, each of these markers acts as a probe for the latent causal marker, with the amount of signal preserved depending mostly on the correlation (linkage disequilibrium) ρ_{ld} between the observed markers in the region and the unobserved causal marker. Under this model, univariate approaches are expected to perform well, specially if ρ_{ld} is sufficiently large to preserve a substantial proportion of the association signal. Nevertheless, we postulate that, by aggregating neighbouring variants as our significant region mining algorithms do, we might partly eliminate the noise introduced when the correlation ρ_{ld} is imperfect. If this hypothesis is true, significant region mining would outperform univariate approaches, specially for low-to-moderate ρ_{ld} . In order to verify this, we compare our proposed approach FastCMH with its corresponding univariate baseline Univariate-CMH. We also include as a baseline the naive version of significant region mining Bonf-CMH to explore the importance of Tarone’s method in this new setting.

DATA GENERATION: To simulate synthetic GWAS datasets under a simplified model of linkage disequilibrium, we first divide the p markers into b disjoint blocks of $p_b = \frac{p}{b}$ markers each. Within each block, all markers have pairwise correlation ρ_{ld} whereas markers belonging to different blocks are uncorrelated. This results in genomic sequences x which exhibit a block-wise correlation structure, resembling the spatial correlation patterns which arise in real-world GWAS datasets due to linkage disequilibrium.

All except two randomly chosen blocks will be populated by “background” markers. For each marker j in each of these blocks, we first randomly sample a value for the first order moment $\mathbb{E}(X(j))$ from a uniform distribution in the range $[0.01, 0.20]$. Using terminology from statistical genetics, this determines the minor allele frequency of the j -th marker, $MAF_j = \mathbb{E}(X(j)) \sim U(0.01, 0.20)$. Then, we generate all p_b markers in each block using the approach in [85] to obtain i.i.d. draws from a multivariate Bernoulli distribution with the first order moments sampled previously and second order moments determined by the condition $\text{Corr}(X(j), X(j')) = \rho_{ld}$ for all $j \neq j'$.

Generating the markers for the two remaining blocks, which carry the truly associated and confounded signals, is slightly more involved. The key difference lies in the markers located at the centre of each block, which are used to generate the case/control labels Y and the covariate C . In particular, Y has correlation $\frac{\rho}{2}$ with each of the two central markers, where ρ is the overall signal strength in the dataset. In turn, the covariate C is obtained as the output of a binary symmetric channel with a small error rate $\epsilon = 0.05$ and the central marker of the confounded block as an input. Thus, the covariate C and the central marker of the confounded block are strongly positively correlated. Moreover, both central markers, as well as Y and C , have first order moments set to 0.5.

The final and most crucial step in the generation of the synthetic datasets is the removal of the central marker of each block, including those which harbour the signal. As a result, the total number of markers in the dataset decreases to $p - b$ but, most importantly, this causes the signal to be present in the observed markers only indirectly.

METRICS: Unlike under a model of genetic heterogeneity, in this new setting the markers in the truly associated block are merely proxies for the unobserved causal variant. As previously discussed, all of them reflect the same underlying signal rather than providing weak but complementary signals. Motivated by this, in this set of experiments we consider that the truly associated block has been detected as long as any of the clusters \mathcal{C}_{sig} in $\mathcal{M}_{\text{sig,clustered}}$ has non-zero overlap with the block. Our definition of detection for the univariate approaches is the same as in previous experiments: if at least one of the markers in the block is deemed as significant, the truly associated block will be counted as successfully discovered for these baselines.

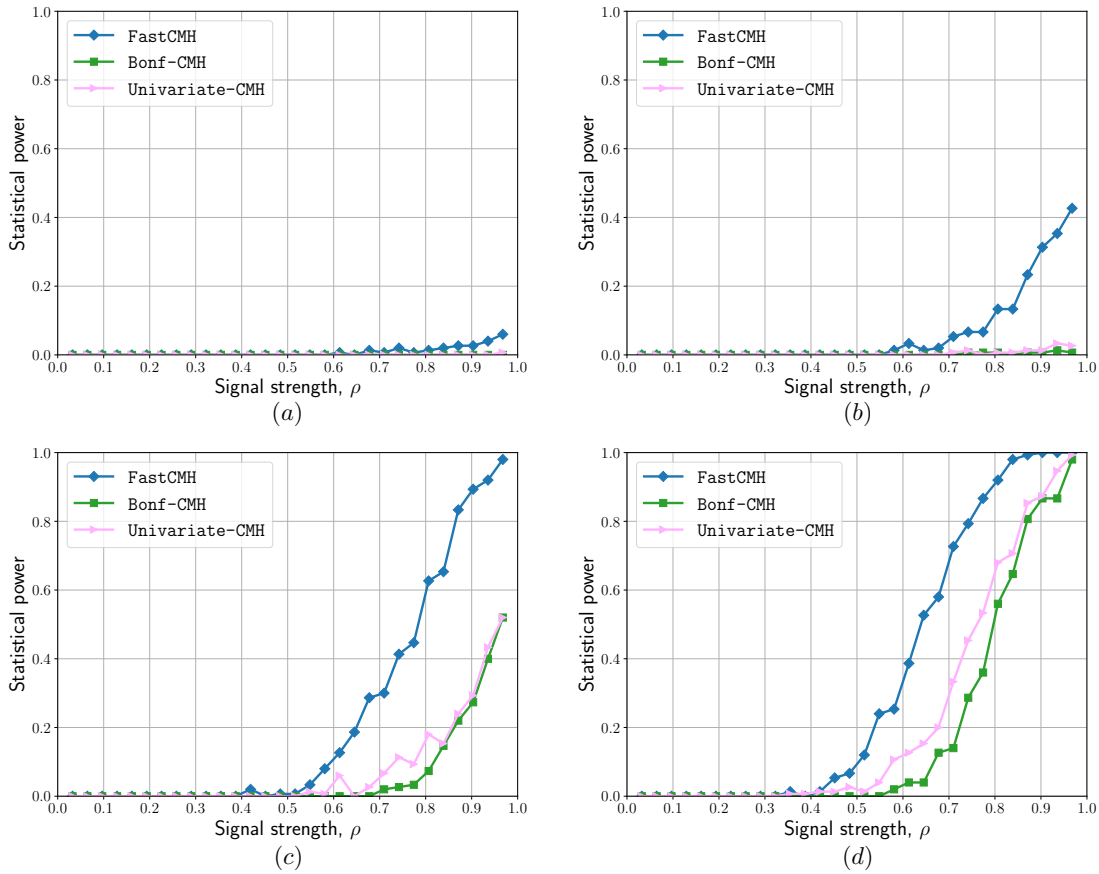


Figure 6.8. – Statistical power at different values of the signal strength ρ for our proposed approach FastCMH, as well as two baseline algorithms: Bonf-CMH and Univariate-CMH. Four different values of the block-wise linkage disequilibrium hyperparameter ρ_{ld} were considered: (a) $\rho_{\text{ld}} = 0.175$, (b) $\rho_{\text{ld}} = 0.25$, (c) $\rho_{\text{ld}} = 0.375$ and (d) $\rho_{\text{ld}} = 0.5$.

RESULTS: We generated synthetic GWAS datasets using the approach described above, with $n = 500$ samples and $p = 1,000,000$ markers divided into $b = 10,000$ blocks of $p_b = 100$ markers each. In order to investigate the effect of the strength of linkage disequilibrium ρ_{ld} , we repeated the experiments for four values of this hyperparameter: $\rho_{\text{ld}} = \{0.175, 0.25, 0.375, 0.5\}$. Figure 6.8 shows the resulting statisti-

cal power⁶ as a function of the signal strength ρ , measured by averaging across 150 repetitions of the experiment. These results clearly demonstrate that significant region mining algorithms can also be useful under models other than genetic heterogeneity. While, as expected, the univariate approaches are competitive under a model of linkage disequilibrium, specially for large ρ_{ld} , FastCMH offers the best performance. The gap in statistical power is particularly pronounced when the linkage disequilibrium is low-to-moderate, that is, for $\rho_{ld} \in [0.25, 0.375]$. Moreover, it is also remarkable that Bonf-CMH underperforms Univariate-CMH for all settings of ρ_{ld} we considered, reaffirming the key role that Tarone’s method plays in making significant region mining practical.

6.3.3 Experiments on real-world human and plant GWAS datasets

In the final section of this chapter, we analyse six real-world GWAS datasets using both of our proposed significant region mining algorithms, FAIS- χ^2 and FastCMH. In particular, we consider COPDGene, a case/control association study of COPD including two human cohorts, as well as five GWAS datasets involving five different dichotomous phenotypes for the plant model organism *A. thaliana*. In these experiments, we aim to investigate the effectiveness of significant region mining in exploiting genetic heterogeneity when analysing real-world data. To this end, we performed an exhaustive comparison of our proposed approaches with several burden test baselines. Moreover, we will evaluate the extent to which FastCMH is able to correct for population structure in significant region mining, contrasting its results with those of FAIS- χ^2 . Throughout all experiments, the target FWER was $\alpha = 0.05$ for all methods under consideration.

Description of the datasets and preprocessing

HUMAN DATA: We analysed data from the COPDGene study [102], which aims to discover genetic risk factors for COPD. The individuals which participated in this study and are included in the dataset we had access to belong to two distinct ethnic groups: non-Hispanic whites and African Americans. After eliminating samples for which the case/control status and/or height was unknown, there were 7,993 individuals in total, 3,633 of which were cases and 4,360 of which were controls. For quality control purposes, we filtered any SNPs which had a minor allele frequency smaller than 0.01 and/or for which a statistical test for Hardy-Weinberg equilibrium yielded a P-value smaller or equal than 10^{-6} . We also removed any markers which were not measured in both cohorts. As a result, we kept a total of 615,906 SNPs. Any sporadic missing values were imputed following [103].

PLANT DATA: We downloaded *A. thaliana* GWAS datasets from the well-known collection described in [87] using the easyGWAS online resource [88]. This collection includes a total of 107 phenotypes, 21 of which are dichotomous. In this chapter, we focus specifically on 5 out of those 21 binary traits, which were subjected to a

6. The corresponding FWER results, which confirm that all methods under consideration control the FWER at the target level $\alpha = 0.05$, can be found in [52].

detailed analysis in our second publication on significant region mining [52]. These five phenotypes were chosen in [52] to be part of the experiments mainly on the basis of two criteria: (i) being relatively balanced, with the proportion of cases not falling below 0.2 or exceeding 0.8 and (ii) exhibiting a non-trivial amount of confounding due to population structure, as measured by the genomic inflation factor [90] (see metrics in Section 5.6.3). Owing to this selection criteria, we believe that these phenotypes constitute a good test-bed for evaluating the ability of FastCMH to account for population structure in real-world data as well as for comparing significant region mining with the burden test baselines. The five selected phenotypes can be subdivided into three hypersensitive-response traits (*avrB*, *avrPphB* and *avrRpm1*) and two lesioning or yellowing leaves traits (LY and LES). The sample size in the five resulting datasets varies between 84 and 95. In this study, no minor allele frequency filter was applied, leading to the number of markers ranging from 214,022 to 214,051. For additional experimental results concerning the remaining, less informative 16 binary traits not discussed in this thesis, we kindly refer the reader to our first publication on significant region mining [51].

Table 6.1. – Characteristics of the six GWAS datasets used in this chapter.

Dataset and phenotype	Samples <i>n</i>	Cases %	SNPs <i>p</i>	<i>k</i> *
COPDGene				
▷ COPD	7,993	45.4	615,906	8
<i>A. thaliana</i>				
▷ <i>avrB</i>	87	63.2	214,032	3
▷ <i>avrRpm1</i>	84	66.7	214,022	3
▷ <i>avrPphB</i>	90	51.1	214,032	4
▷ LES	95	22.1	214,051	3
▷ LY	95	30.5	214,051	5

Definition of the categorical covariates

As extensively discussed in Chapter 5, incorporating into the model covariate factors that might have a confounding effect is of utmost importance. If unaccounted for, these could lead to a potentially large number of spurious discoveries, seriously compromising the reliability of the results. Population structure [104] is an ubiquitous source of confounding in GWAS. All six datasets covered in our experiments contain individuals with considerably diverse genetic ancestries. The COPDGene study includes both a cohort of non-Hispanic white individuals and a cohort of African American individuals. Similarly, the *A. thaliana* samples from the collection described in [87] originate from a large number of locations in Europe and Asia. In order to account for population structure, we will utilise the ability of FastCMH to correct for a categorical covariate, contrasting its results with those obtained by FAIS- χ^2 , which remains susceptible to confounding due to population structure. In order to define a categorical covariate that represents population structure, we follow the same approach introduced in Section 5.6.3. In summary, we first used EIGENSTRAT [91] to

obtain the three eigenvectors of the kinship matrix corresponding to the three largest eigenvalues. This results in a real-valued, three-dimensional covariate $\mathbf{c}_i \in \mathbb{R}^3$ for each sample $i = 1, 2, \dots, n$. Next, we applied the k -means algorithm to the collection of three-dimensional embeddings $\{\mathbf{c}_i\}_{i=1}^n$, resulting in a set $\{\tilde{c}_i\}_{i=1}^n$ of cluster assignments satisfying $\tilde{c}_i \in \{1, 2, \dots, k\}$ for all i . These cluster assignments can be understood as a discretised version of the original covariates $\mathbf{c}_i \in \mathbb{R}^3$ and will be used by FastCMH to account for population structure. As in Chapter 5, despite the simplicity of this heuristic vector quantisation approach, we empirically found its performance to be satisfactory. The number of categories k , which in this case corresponds to the number of centroids used by k -means, was chosen independently for each dataset as the value k^* that resulted in the smallest genomic inflation factor λ . To obtain k^* , we explored values of k in the range $\llbracket 2, 10 \rrbracket$ for the COPDGene study and in the range $\llbracket 2, 5 \rrbracket$ for the *A. thaliana* datasets, which have a considerably smaller sample size. Table 6.1 shows the resulting values of k^* for each dataset, as well as the sample size n , the percentage of cases in the dataset and the total number of markers p .

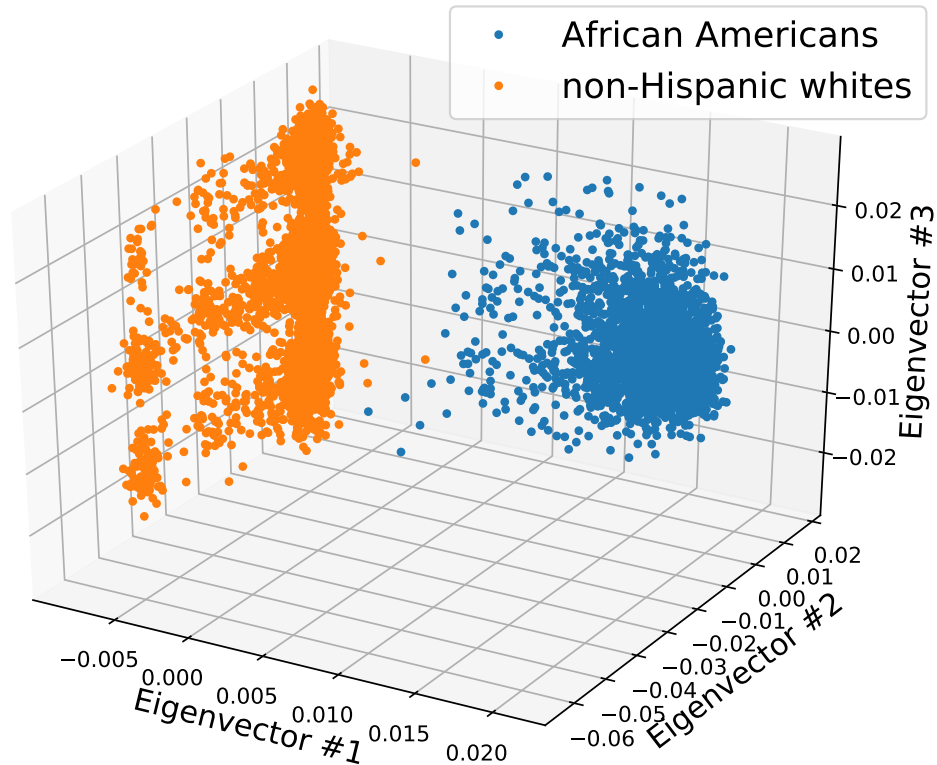


Figure 6.9. – Visualisation of the three-dimensional embedding $\mathbf{c}_i \in \mathbb{R}^3$ obtained by EIGENSTRAT for each sample in the COPDGene study. Each point has been coloured according to the ethnicity of the corresponding participant: blue for African Americans and orange for non-Hispanic whites.

In our original publication [52], we additionally considered a simpler alternative to define the categorical covariate for the COPDGene study. Since the ethnicity (non-Hispanic white or African American) of each participant is known, we can define a

covariate with two categories that indicates the ethnic group each individual belongs to, hoping that it will suffice to capture population structure. To better understand how this simple approach relates to the more sophisticated categorical covariate described above, Figure 6.9 depicts the three-dimensional embedding $\mathbf{c}_i \in \mathbb{R}^3$ of each sample in the COPDGene study obtained using EIGENSTRAT. This visualisation clearly shows that African American and non-Hispanic white participants are well-separated in this representation. Using k -means with $k \geq 2$ is likely to result in clusters consisting mostly of individuals of the same ethnicity. Hence, the categorical covariate obtained by applying k -means to the three-dimensional embeddings inferred with EIGENSTRAT are likely to recover all information regarding the ethnicity of the samples. Moreover, Figure 6.9 also shows substantial genetic diversity between individuals of the same ethnicity, particularly for non-Hispanic white participants. Consequently, merely using ethnicity as a covariate provides a more coarse representation of population structure than a categorical covariate obtained by discretising the output of EIGENSTRAT with k -means. The results in [52] seem to confirm this hypothesis: while ethnicity alone suffices to drastically reduce confounding due to population structure, the approach we will discuss in this thesis achieves an even better reduction in genomic inflation.

Finally, it is worth mentioning that in [52] we also investigated the effect of modifying the number d of eigenvectors used by EIGENSTRAT. To this end, we re-analysed the COPDGene study using all 81 possible settings of (k, d) when each of the two hyperparameters varies in the range $[[2, 10]]$. The main highlight of that experiment, described in Section S3.2.1 of the Supplementary Material of [52], is that either fixing the number of eigenvectors d and optimising the number of categories k (as we do in the experiments described in this thesis by setting $d = 3$) or, alternatively, fixing the number of categories k and optimising the number of eigenvectors d , suffices to achieve almost optimal performance. The only clear exception to this claim occurred when the number of categories k is fixed to 2. In that case, regardless of the choice for d , the resulting categorical covariate is almost identical to the ethnicity of the samples, losing the advantage that the finer representation of population structure obtained with EIGENSTRAT provides. While the same experiment was not repeated for the *A. thaliana* datasets, we do not expect choices other than $d = 3$ to substantially outperform the results we provide here.

Results

The results of our analysis of the COPDGene study and all five *A. thaliana* datasets will be presented next. First, we explore the impact of confounding due to population structure in significant region mining by contrasting the results of FastCMH with those of FAIS- χ^2 . Secondly, we briefly describe the findings of FastCMH and FAIS- χ^2 , investigating to which extent significant region mining is able to find associations between genotype and phenotype that would be missed by univariate testing approaches. Finally, we conclude the chapter by comparing these findings to those obtained by a battery of burden test baselines, with the aim of exploring whether significant region mining algorithms are also competitive with these popular methods developed by the statistical genetics community in real-world data.

Table 6.2. – Summary of the results of our proposed approaches FAIS- χ^2 and FastCMH. The columns λ and “Hits” refer to the genomic inflation factor and the number of disjoint clusters of overlapping genomic regions deemed significantly associated, respectively.

Dataset and phenotype	FAIS- χ^2		FastCMH	
	λ	Hits	λ	Hits
COPDGene				
▷ COPD	16.70	88,403	1.01	3
<i>A. thaliana</i>				
▷ <i>avrB</i>	1.66	14	1.17	11
▷ <i>avrRpm1</i>	1.53	15	1.13	13
▷ <i>avrPphB</i>	1.70	6	1.22	5
▷ LES	2.05	20	1.21	3
▷ LY	2.51	26	1.30	1

ACCOUNTING FOR POPULATION STRUCTURE: As summarised in Table 6.2, the genomic inflation factor λ for FAIS- χ^2 ranges between 1.53 and 2.51 in the *A. thaliana* datasets. This suggests the presence of a moderate-to-high amount of confounding affecting the discoveries made by FAIS- χ^2 . In contrast, FastCMH is able to eliminate most of that inflation, reducing λ to the range 1.13-1.30. The outcome for COPDGene is even more striking. As shown in Figure 6.9, African American and non-Hispanic white individuals have markedly different genetic makeups. Moreover, the proportion of cases differs substantially between both cohorts: only 31.0% of African American participants are cases compared to 52.6% for non-Hispanic whites. Consequently, the confounding effect of population structure is particularly strong in this dataset. Under these conditions, the findings provided by FAIS- χ^2 hold virtually no useful information, as indicated by the extremely large genomic inflation factor $\lambda = 16.70$. However, FastCMH appears to eliminate this inflation almost entirely: the resulting inflation factor is only $\lambda = 1.01$, suggesting a close agreement between the theoretical and empirical medians of the test statistics.

Figure 6.10 further illustrates the effect of confounding in our real-world experiments by means of quantile-quantile (Q-Q) plots. In Figure 6.10, the x-axis depicts the quantiles of the theoretical distribution of P-values under the null hypothesis, i.e. a $U(0, 1)$ distribution. In turn, the y-axis shows the empirical quantiles obtained using the P-values computed for all testable genomic regions. To ease the visualisation, these quantiles have been transformed as $-\log_{10}(p)$, making large (less significant) P-values appear on the bottom/left quadrant of the plot and small (more significant) P-values on the upper/right quadrant of the plot. If the model chosen for the null distribution of the test statistic were correct and there was absolutely no signal in the data, be it due to confounding or due to a truly useful association, the theoretical (expected) and empirical (observed) quantiles should agree closely. In practice, when analysing GWAS datasets we expect a large majority of markers/regions to be independent of the phenotype while a small number of markers/regions might reach significance. According to our parametrisation of the Q-Q plots, this implies that the expected and observed quantiles should agree closely in the bottom-left quadrant of the plot,

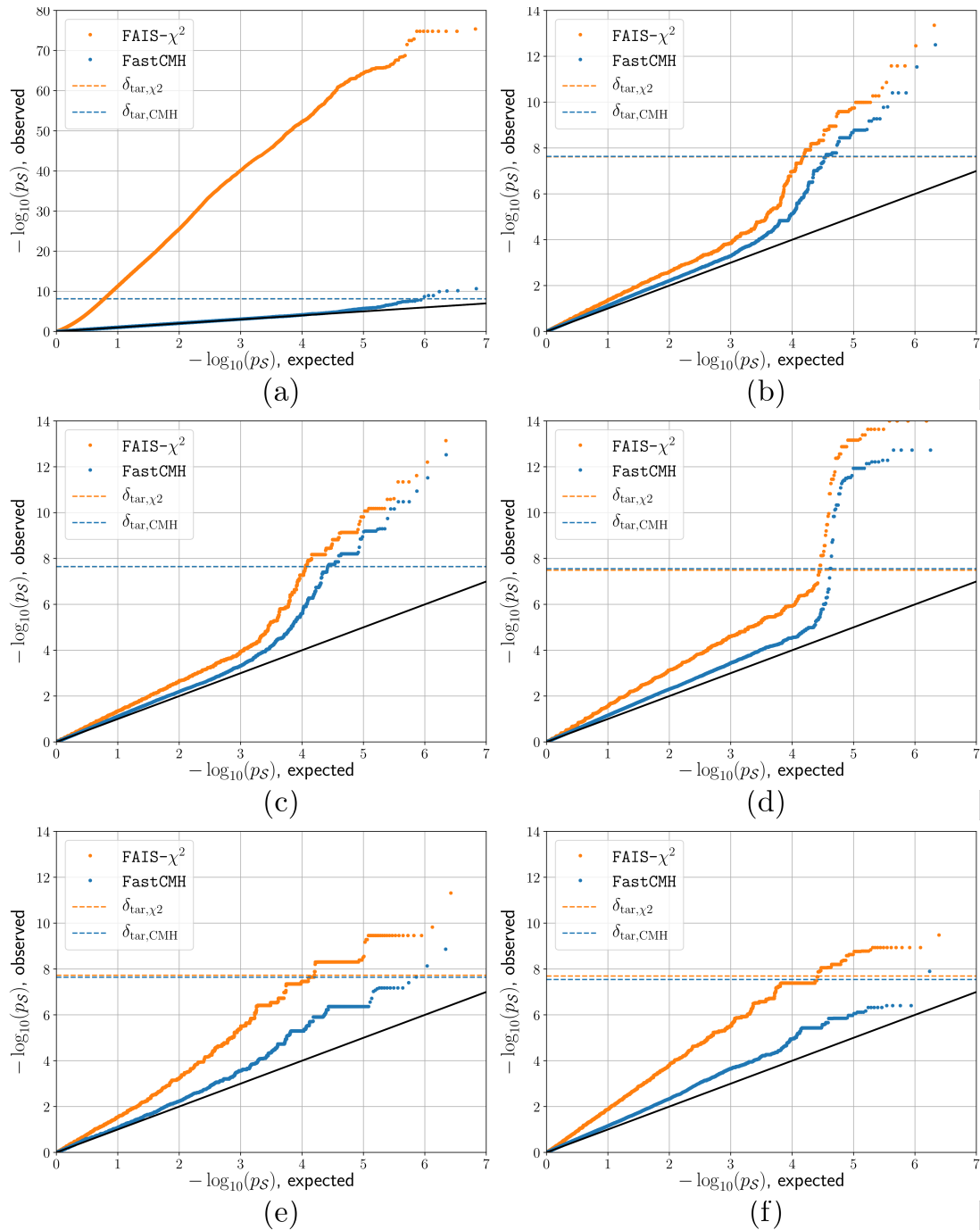


Figure 6.10. – Q-Q plots for the P-values of all testable genomic regions obtained with FAIS- χ^2 and FastCMH for six datasets: (a) COPDGene, (b) *A. thaliana avrB*, (c) *A. thaliana avrRpm1*, (d) *A. thaliana avrPphB*, (e) *A. thaliana* LES and (f) *A. thaliana* LY. Dashed horizontal lines show the corrected significance thresholds for each of the two approaches.

which corresponds to insignificant P-values, while deviations in the upper-right could be consistent with “true” hits. However, if the Q-Q plot showed that the expected and observed quantiles deviate substantially also for P-values of low-to-

moderate significance, i.e. in the bottom-left quadrant of the plot, the possibility that confounding is affecting the results should be analysed in detail. The Q-Q plots shown in Figure 6.10 for all six datasets indicate clearly that the observed quantiles for FastCMH resemble the expected quantiles much more closely than those of FAIS- χ^2 for P-values of low-to-moderate significance, paralleling the observations made by means of the genomic inflation λ .

Nevertheless, despite vastly improving over FAIS- χ^2 , it can be observed from both Table 6.2 and Figure 6.10 that a slight residual inflation remains present for the results of FastCMH in all five *A. thaliana* datasets. This phenomenon was already observed in Section 5.6.3 when analysing the datasets corresponding to phenotypes *avrB* and LY using FACS. To further explore this effect, we computed the genomic inflation factor λ obtained by univariate analyses using both the CMH test with the same categorical covariate as FastCMH and the LRT test under a logistic regression model with the original (real-valued) embedding provided by EIGENSTRAT as covariates. For both univariate baselines and all five datasets, the same residual inflation can be observed: λ ranges between 1.18 and 1.80. Consequently, as in Section 5.6.3, we hypothesise this to be mainly due to the difficulty in inferring a highly-accurate representation of population structure with such a small sample size. The COPDGene study, which has about 100 times more samples than the *A. thaliana* datasets, does not seem to be affected by this limitation.

RETRIEVED GENOMIC REGIONS: Table 6.2 also summarises the number of disjoint clusters of overlapping genomic regions deemed significantly associated (“hits”) by FAIS- χ^2 and FastCMH. Overall, it can be seen that the number of “hits” for FastCMH is systematically smaller than for FAIS- χ^2 .

For all *A. thaliana* datasets, FastCMH found 33 significantly associated clusters in total while FAIS- χ^2 found 81, more than twice as many. Most importantly, the difference in the number of “hits” appears to correlate with the difference in genomic inflation. For instance, for the phenotype LY, which suffers the most from confounding due to population structure ($\lambda = 2.51$ for FAIS- χ^2), FAIS- χ^2 deemed 26 clusters as significant, while only one was significantly associated after accounting for population structure with FastCMH. The situation is similar for the phenotype LES, for which the genomic inflation factor for FAIS- χ^2 is the second largest. 20 clusters of genomic regions were found to be significantly associated with FAIS- χ^2 whereas FastCMH only retrieved three. Out of all 33 clusters of overlapping genomic regions found by FastCMH, only 16 of them contain at least one marker that would have been deemed significantly associated by a univariate baseline. Even if we considered univariate “hits” up to ten kilobases apart to be sufficient for having “discovered” a region, a relaxation that could be justified due to linkage disequilibrium, six clusters retrieved by FastCMH would still be undetected. All in all, these findings illustrate the potential of significant region mining to complement the univariate analyses which remain common practice in the statistical genetics community.

The results obtained for the COPDGene study are once again even more dramatic. FAIS- χ^2 , which suffers from an extreme level of genomic inflation in this dataset ($\lambda = 16.70$), reports an anomalously large number of “hits” (88,403). The corresponding results for FastCMH tell a very different story. Only three clusters of overlapping

significantly associated genomic regions are deemed significant after population structure is taken into account. Moreover, these three clusters constitute biologically plausible findings. The regions in each cluster overlap with a distinct gene in a gene cluster referred to as the (CHRNA5-CHRNA3-CHRNA4) nicotinic acetylcholine receptor (nAChR), located in chromosome 15q25.1. Other studies have previously reported association between these genes and COPD [103, 105]. Most importantly, none of these three clusters contain markers that would have been deemed significantly associated by a univariate analysis, reaffirming the usefulness of exploiting genetic heterogeneity to detect faint association signals in real-world data.

Additional information about all genomic regions found to be significant after correction for population structure by FastCMH, including all markers involved, their location, gene annotations (when available) and the corresponding P-values can be found in the Supplementary Material of [52].

COMPARISON WITH BURDEN TEST BASELINES: To conclude this chapter, we compare FastCMH with a battery of burden tests, which, as stated throughout this chapter, are a related family of approaches from statistical genetics that also aim to exploit genetic heterogeneity in GWASs by aggregating contiguous markers.

As a first set of baselines, we used gene-based burden tests to analyse the five *A. thaliana* datasets as well as the data from the COPDGene study. Each known gene was defined to be a candidate genomic region, including all observed markers that lie in the gene or are at a distance of ten kilobases or less from the gene boundary. This led to a total of 24,426 candidate regions for the plant datasets and 17,817 regions for the human dataset. Moreover, we considered three variants of the burden test baselines, two of which were based on the LRT test and one on the CMH test. The latter performed tests of association between the pattern occurrence indicator $G_S(X) = \mathbb{1} \left[\sum_{j \in S} X(j) > 0 \right]$ and the phenotype Y given the categorical covariate C , thus being statistically analogous to FastCMH for genomic regions that are present in the search space of both approaches. For the two burden test baselines based on the LRT test, we used a logistic regression model with the covariate C represented as k dummy explanatory variables, which are present in both the null and alternative models. The alternative models additionally include as explanatory variables either the burden count $\sum_{j \in S} X(j)$ or the pattern occurrence indicator $G_S(X) = \mathbb{1} \left[\sum_{j \in S} X(j) > 0 \right]$. Furthermore, for the five *A. thaliana* datasets, for which it was previously shown that correcting for population structure is particularly challenging due to their small sample size, we considered yet another modification of the burden test baselines. For both LRT-based burden tests, we also repeated the analysis using the real-valued covariates provided by EIGENSTRAT as explanatory variables. The Bonferroni correction was used to account for multiple comparisons in all cases, with a target FWER of $\alpha = 0.05$.

As a result for *A. thaliana*, if we take the union of all regions deemed significantly associated by any of the five gene-based burden test baselines, we find that 21% of the regions they recover are also found by FastCMH, including those which are most significant. Moreover, gene-based burden tests are by design unable to discover associations between the phenotype and any marker or region that does not overlap or is near a gene. This causes up to 45% of all the SNPs discovered by FastCMH to

be missed by the gene-based burden test baselines. Overall, 40% of all significantly associated regions are only retrieved by FastCMH. The genomic inflation factor for the burden test baselines was highly variable and, on many occasions, was substantially higher than that of FastCMH. For example, for the phenotypes LES and LY, the results for some of the burden test baselines had genomic inflation factors higher than 1.6. Hence, it is plausible that some of the “hits” obtained with these baselines are conflated by population structure. Also, it is worth noting that we found considerable discrepancies between the results of the different five burden test baselines, suggesting that choosing a statistical model and representation for the covariate is non-trivial for these methods. Rather than simply taking the union of all “hits”, a more conservative approach would have required an additional correction for multiple comparisons. Nevertheless, we chose to overlook this limitation, resulting in a more challenging comparison for our proposed approach.

Concerning the COPDGene study, none of the three genes (CHRNA5-CHRNA3-CHRN4) found by FastCMH was significant when using any of the three gene-based burden test baselines. Taking the minimum (most significant) P-value across all three baselines, only the gene CHRN4 was relatively close to being deemed significantly associated, with a P-value of $5.72 \cdot 10^{-6}$. In contrast, the genes CHRNA5 and CHRNA3 had minimum P-values of 0.24 and 0.41, respectively. This apparent discrepancy owes to the fact that the significantly associated genomic regions found by FastCMH do not span the entire genes. By using candidate genomic regions which are too large, the truly associated subset of markers in the genes is mixed with other irrelevant features, making the association signal undetectable for these burden test baselines. FastCMH successfully circumvents this problem, as it is able to test all genomic regions regardless of their length. Nevertheless, the gene-based burden tests found the gene ZRANB3 to be significantly associated, with a minimum P-value of $1.56 \cdot 10^{-6}$. FastCMH assigns a similar level of significance to this gene, with a P-value of $2.31 \cdot 10^{-6}$. However, under the more stringent corrected significance threshold necessitated by FastCMH, P-values of the order of 10^{-6} are not sufficiently extreme to be deemed significant.

Finally, we also analysed all six datasets using burden tests with non-overlapping windows of two different sizes: 500 kilobases and one megabase. The statistical models and representations for the covariate C used for this study were the same as for the gene-based burden tests. For the COPDGene dataset, the results for these baselines mostly coincide with their gene-based counterparts: burden tests were still unable to retrieve any of the three genes in the nicotinic acetylcholine receptor (nAChR), but some of them found ZRANB3 to be significant. In *A. thaliana*, once again we observed substantial variability across the different versions of the burden test baselines, and also between the gene-based burden tests and the burden tests with non-overlapping windows. Nevertheless, the same qualitative assessment remains valid: the burden test baselines are unable to retrieve many of the findings of FastCMH but also deem some regions not found by FastCMH as significant.

In summary, our results in real-world data illustrate the complementary nature of univariate tests, burden tests and significant region mining. None of these approaches appears to be intrinsically superior. Rather, when considered together, these methods provide the user the possibility to perform analyses at different levels of granularity.

Univariate methods are simple, computationally efficient and easy to interpret. They also suffice to pick up moderate-to-strong association signals, making them a good starting point when analysing a GWAS dataset. Burden tests and significant region mining are able to exploit genetic heterogeneity, allowing to detect weaker associations that would be missed by univariate tests. This effect is particularly relevant when analysing rare variants, and could become increasingly important in GWASs as the price of whole-genome sequencing continues to plummet. Burden tests are particularly useful when prior knowledge is available to select a small subset of promising candidate genomic regions. As burden tests often require not-too-stringent corrected significance thresholds, if the choice of candidate regions is adequate, they can discover associations that significant region mining algorithms would miss. In contrast, significant region mining makes it possible to carry out an exhaustive scan of the genome, being completely agnostic about the length or location of the truly associated regions. We believe this makes significant pattern mining a promising tool for situations in which sufficiently reliable prior knowledge is not available.

Part III

DISCUSSION AND OUTLOOK

SUMMARY AND OUTLOOK

This final chapter will be devoted to summarise the main ideas, contributions and results presented throughout this thesis, as well as discuss what in our opinion are some of the most pressing open problems in significant pattern mining.

7.1 SUMMARY

Significant pattern mining combines techniques from statistical association testing and data mining to efficiently discover patterns in data which occur statistically significantly more often in one class of samples than in another. What sets significant pattern mining apart from other approaches is its unprecedented ability to cope with enormous search spaces, possibly containing trillions of candidate hypotheses, while guaranteeing strict FWER control and exhibiting substantial statistical power. For example, significant itemset mining algorithms allow to assess the statistical association of *all* high-order interactions between binary features with a target of interest. In the context of biomarker discovery, these methods are a powerful complement to the univariate association tests and additive models commonly used by domain experts in computational biology and medicine, as interactions are known to play a key role in many biological mechanisms. Other instances of significant pattern mining considered in this thesis include significant subgraph mining, which aims to discover subgraphs whose occurrence is statistically significantly more common in one specific class of graphs, and significant region mining, which aims to discover genomic regions for which the presence of one or more copies of a minor allele is statistically significantly enriched in cases or controls.

According to the general formulation proposed in the first part of this thesis, an instance or variant of significant pattern mining can be defined on the basis of: (i) the type of data under consideration, more precisely, the input domain \mathcal{X} in which observations x lie; (ii) the notion of pattern \mathcal{S} and the search space \mathcal{M} containing all candidate patterns \mathcal{S} under study; and (iii) the concept of occurrence of a pattern \mathcal{S} in an observation $x \in \mathcal{X}$. As an example, significant itemset mining can be characterised by: (i) the input domain is $\mathcal{X} = \{0, 1\}^p$, the set of all p -dimensional binary vectors; (ii) any subset of features $\mathcal{S} \subseteq \{1, 2, \dots, p\}$ constitutes a pattern, leading to the search space \mathcal{M} being the power set of $\{1, 2, \dots, p\}$; and (iii) a pattern \mathcal{S} occurs in an observation $x = (u_1, u_2, \dots, u_p)$ if the multiplicative interaction of the binary features indexed by \mathcal{S} , $z_{\mathcal{S}}(x) = \prod_{j \in \mathcal{S}} u_j$, is non-zero. Both significant subgraph mining and significant region mining can also be characterised analogously. Irregardless of the data-specific aspects of the pattern mining problem, the goal of significant pattern mining can be stated simply using this abstract formulation. We aim to find all patterns \mathcal{S} in the search space \mathcal{M} for which the binary random variable $G_{\mathcal{S}}(X)$ that indicates the presence or absence of \mathcal{S} in a (random) observation $X \in \mathcal{X}$ is statistically associated with the binary random variable Y that represents the class label of X .

This problem statement highlights the crucial role that statistical association testing plays in significant pattern mining. Techniques such as Pearson’s χ^2 test and Fisher’s exact test, proposed almost a century ago, can be readily used to assess the statistical association between $G_S(X)$ and Y based on a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ with n i.i.d. realisations of (X, Y) . However, what poses an extraordinary challenge in this setting is the tremendous number of simultaneous association tests that need to be performed. In other words, significant pattern mining algorithms must handle an extreme instance of the multiple comparisons problem. A common criterion to account for the fact that many hypotheses are being tested in parallel is to control the FWER, defined as the probability of reporting *any* false positives in the entire collection of tests. Well-known techniques to control the FWER, such as the Bonferroni correction, rely on introducing a corrected significance threshold δ that becomes more stringent as the number of tests $|\mathcal{M}|$ being performed increases. Nevertheless, these techniques completely break down when applied to significant pattern mining, which involves an unprecedented number of association tests. In this extreme regime, methods like the Bonferroni correction would result in virtually no statistical power to detect the true associations present in the data. The solution to this conundrum, which has deterred the development of significant pattern mining for decades, relies on exploiting special properties of discrete data. In particular, Tarone’s improved Bonferroni correction for discrete data, introduced by R. E. Tarone in 1990, lies at the core of significant pattern mining. Tarone’s method is based on the observation that, when assessing the statistical association between two discrete random variables, there exists a minimum attainable P-value strictly larger than zero. Consequently, it can be shown that many candidate patterns in the search space are untestable, that is, they can neither be deemed significant nor cause a false positive. By proving that the corrected significance threshold δ must only be adjusted to account for the number of testable patterns, which in practice comprise only a small proportion of the entire search space, Tarone’s method provides an elegant way to solve the main statistical difficulties that significant pattern mining entails.

Despite being arguably the single most important concept in significant pattern mining, Tarone’s idea of testability was first proposed in a very different context. The original publication considered applications, such as clinical trials, which required fewer than fifty simultaneous association tests. In this setting, Tarone’s method was shown to outperform a naive Bonferroni correction. However, the difference between both approaches in the small-scale regime is nowhere as dramatic as it turned out to be in problems involving trillions of hypotheses, such as significant pattern mining. Perhaps most importantly, the applications considered in the original article did not necessitate a computationally efficient approach to compute the corrected significance threshold δ_{tar} resulting from Tarone’s method. A naive strategy, based on explicitly computing the minimum attainable P-value for every single hypothesis, was feasible and used instead. As a result, the enormous potential of Tarone’s method for significant pattern mining remained unnoticed until 2013, when the authors of the LAMP algorithm showed that a computationally tractable implementation of Tarone’s method for significant pattern mining was possible and, in doing so, that Tarone’s method held the key to successfully correct for multiple comparisons in significant pattern mining. LAMP can be understood as a combination of Tarone’s concept of

testability and classical techniques from discriminative pattern mining. In particular, computational tractability is achieved by means of a carefully designed search space pruning criterion, which allows to exactly compute δ_{tar} without the need to neither enumerate the majority of patterns in the search space nor evaluate their respective minimum attainable P-values. This pruning criterion relies on specific properties of the minimum attainable P-value which results from certain statistical association tests, including both Pearson's χ^2 test and Fisher's exact test. Given a certain input dataset, with fixed sample size n and number of samples in the positive class n_1 , it can be shown that the minimum attainable P-value $p_{\mathcal{S},\text{min}}$ of a pattern \mathcal{S} can be computed as a function $p_{\text{min}}(r_{\mathcal{S}})$ of the support $r_{\mathcal{S}}$ of \mathcal{S} , that is, as a function of the number of samples in which pattern \mathcal{S} occurs. This finding is not only remarkable because it provides a computationally efficient way to compute $p_{\mathcal{S},\text{min}}$. Its most important consequence is that, by additionally proving that the function $p_{\text{min}}(r_{\mathcal{S}})$ is monotonically decreasing for any $r_{\mathcal{S}} \leq \min(n_1, n - n_1)$, the well-known Apriori property of pattern mining can be invoked to design a valid search space pruning criterion: if $r_{\mathcal{S}} \leq \min(n_1, n - n_1)$ and \mathcal{S} is untestable, that is, $p_{\mathcal{S},\text{min}} > \delta$, then $p_{\mathcal{S}',\text{min}} > \delta$ necessarily holds for any $\mathcal{S}' \supseteq \mathcal{S}$. In summary, provided that $r_{\mathcal{S}} \leq \min(n_1, n - n_1)$, i.e. that pattern \mathcal{S} is "rare enough", \mathcal{S} being untestable implies that all supersets of \mathcal{S} must also be untestable and can be pruned from the search space. By incorporating this pruning criterion as part of the pattern enumeration process, Tarone's corrected significance threshold δ_{tar} can be efficiently computed in practice.

The LAMP algorithm was originally proposed as a method to find arbitrary combinations of transcription factors which are statistically associated with up-regulation of gene expression. However, by solving that problem, the authors of LAMP proved that assessing the statistical association of *all* high-order interactions between binary features with a target of interest was, after all, a solvable problem. In spite of this achievement, the original LAMP algorithm is not devoid of limitations. The main contributions of this thesis precisely aimed at addressing what we believe were some of the most pressing shortcomings of the state of the art in significant pattern mining.

A key limitation of Tarone's improved Bonferroni correction for discrete data and, consequently, of the LAMP algorithm, is its inability to exploit the existence of statistical dependencies between the test statistics corresponding to different patterns. The search space in significant pattern mining is not only enormous, but also heavily redundant. Perhaps the most evident source of statistical dependence are subset/superset relationships between patterns. If $\mathcal{S}' \supseteq \mathcal{S}$ then, except in pathological cases, it follows that the pattern occurrence indicators $G_{\mathcal{S}}(X)$ and $G_{\mathcal{S}'}(X)$ are positively correlated random variables. More generally, the same phenomenon might take place when any two patterns $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{M}$ share a common substructure, i.e. when $\mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset$. $G_{\mathcal{S}}(X)$ and $G_{\mathcal{S}'}(X)$ being positively correlated has profound implications, namely, false positives for \mathcal{S} will tend to co-occur with false positives for \mathcal{S}' . Since any given candidate pattern \mathcal{S} in the search space might be related in this manner to many other candidate patterns, the effective number of tests one must account for in order to guarantee FWER control might be drastically smaller than the total number of (testable) patterns. This offers an ideal opportunity to improve statistical power compared to simpler, more conservative approaches which ignore this phenomenon. Westfall-Young permutation testing is a popular resampling-based procedure that

can be used to empirically estimate a corrected significance threshold that accounts for the dependence between test statistics. First, j_p resampled datasets $\{\tilde{\mathcal{D}}^{(k)}\}_{k=1}^{j_p}$ are obtained by randomly permuting the class labels with respect to the observations, that is, $\tilde{\mathcal{D}}^{(k)} = \left\{ \left(x_i, y_{\pi^{(k)}(i)} \right) \right\}_{i=1}^n$, where $\pi^{(k)} : \llbracket 1, n \rrbracket \rightarrow \llbracket 1, n \rrbracket$ is a random permutation of $\llbracket 1, n \rrbracket$. By construction, the global null hypothesis that no pattern is statistically associated with the (randomly permuted) class labels holds for all resampled datasets $\{\tilde{\mathcal{D}}^{(k)}\}_{k=1}^{j_p}$. Hence, the FWER at an arbitrary corrected significance threshold δ can be estimated as the proportion of resampled datasets for which one or more patterns were erroneously deemed significantly associated with the permuted class labels at level δ . If $p_{\text{ms}}^{(k)}$ denotes the smallest, i.e. most significant, P-value among all patterns in the search space for the k -th resampled dataset, the resulting corrected significance threshold would be given by $\delta_{\text{wy}} = \max \left\{ \delta \mid \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[p_{\text{ms}}^{(k)} \leq \delta \right] \leq \alpha \right\}$. Provided that j_p is sufficiently large, e.g. in the order of 10,000, the estimate δ_{wy} will be robust in practice. Unfortunately, naively applying permutation testing in significant pattern mining is computationally unfeasible. Finding $p_{\text{ms}}^{(k)}$ via brute force is not computationally tractable even for a single resampled dataset, let alone for $j_p \approx 10,000$ of them. FastWY, the only previously existing permutation testing-based significant pattern mining algorithm, uses Tarone’s concept of testability to compute $p_{\text{ms}}^{(k)}$ efficiently for each resampled dataset. However, even though FastWY can obtain $p_{\text{ms}}^{(k)}$ while enumerating only a small subset of patterns in the search space, it is severely hindered by the need to repeat the entire pattern enumeration process from scratch for each of the j_p resampled datasets. In this thesis we proposed Westfall-Young light, a novel permutation testing-based significant pattern mining algorithm that computes δ_{wy} exactly while enumerating patterns only once. FastWY and Westfall-Young light are statistically indistinguishable but differ vastly in terms of computational efficiency. A comprehensive empirical study comprising 12 itemset mining datasets and other 12 subgraph mining databases showed that Westfall-Young light is one to three orders of magnitude faster than FastWY. Even most importantly, the runtime gap appears to increase with the absolute runtime needed to analyse the dataset, hinting at the possibility that Westfall-Young light scales more gently than FastWY in practice. Both approaches also exhibit stark differences in memory usage, specially for the largest datasets considered in our experiments. For these, Westfall-Young light was often found to use two to three orders of magnitude less memory. In particular, three of the itemset mining datasets could not be analysed using FastWY due to its memory requirements, which exceeded the 256 GB our server was equipped with by a large margin. In contrast, our proposed approach Westfall-Young light did not have the same problem, being able to run until completion for all 24 datasets included in our study.

Perhaps the single most pressing limitation of the first generation of significant pattern mining algorithms is the impossibility to model external covariate factors. The need to account for factors of variation that could have a confounding effect is an ubiquitous problem when exploring data for the life sciences. Given a covariate C that takes values in a domain \mathcal{C} , we say that C has a confounding effect on the association between $G_S(X)$ and Y if these two random variables are (marginally) statistically

associated yet are statistically independent given C . Intuitively, this situation implies that the presence or absence of pattern \mathcal{S} in an observation X provides no additional information about Y besides the information that is already contained in the covariate C . In most potential applications of significant pattern mining in the life sciences, the aim is to discover association signals which are unrelated to certain sources of variation such as age, gender or population structure. However, LAMP, FastWY or Westfall-Young light are all designed to detect patterns \mathcal{S} in the search space \mathcal{M} for which $G_{\mathcal{S}}(X)$ and Y are deemed (marginally) significantly associated. As a result, many of the patterns retrieved by these algorithms could turn out to be uninformative when examined in the light of relevant covariate factors that have been unaccounted for. A fundamental contribution described in this thesis is FACS, a novel significant pattern mining algorithm that allows to correct for a categorical covariate C with an arbitrary number of categories k . More precisely, FACS allows to find patterns \mathcal{S} in the search space \mathcal{M} for which $G_{\mathcal{S}}(X)$ and Y are statistically associated given C , under the assumption that C takes values in $\mathcal{C} = \{1, 2, \dots, k\}$. The Cochran-Mantel-Haenszel (CMH) test allows assessing the statistical association between two binary random variables $G_{\mathcal{S}}(X)$ and Y given a categorical random variable C , thus being an ideal choice for this problem. However, replacing unconditional association tests such as Pearson's χ^2 test or Fisher's exact test by the CMH test entails considerable methodological complications. The first obstacle that had to be overcome concerns the existence of a computationally tractable minimum attainable P-value function for this test. Owing to the discrete nature of the CMH test, a minimum attainable P-value does indeed exist, and can be computed in only $O(k)$ operations as a multivariate function $p_{\min}(\mathbf{r}_{\mathcal{S}})$ of k variables $\{r_{\mathcal{S},c}\}_{c=1}^k$, where $r_{\mathcal{S},c}$ stands for the support of pattern \mathcal{S} in samples for which the covariate belongs to category c . Nevertheless, the mere existence and tractability of the minimum attainable P-value function $p_{\min}(\mathbf{r}_{\mathcal{S}})$ does not suffice to make use of the CMH test as part of a significant pattern mining algorithm; an alternative search space pruning criterion valid for the CMH test must also be derived. Existing significant pattern mining algorithms, such as LAMP, rely on the fact that the function $p_{\min}(r_{\mathcal{S}})$ is monotonically decreasing for $r_{\mathcal{S}} \leq \min(n_1, n - n_1)$. This allows them to use a simple yet effective pruning criterion, consisting of removing from the search space all supersets of untestable patterns \mathcal{S} which satisfy $r_{\mathcal{S}} \leq \min(n_1, n - n_1)$. Unfortunately, the minimum attainable P-value function $p_{\min}(\mathbf{r}_{\mathcal{S}})$ for the CMH test does not exhibit an analogous behaviour. To circumvent this limitation, FACS introduces a monotonically decreasing lower bound of $p_{\min}(\mathbf{r}_{\mathcal{S}})$ as a surrogate to construct a pruning criterion. In spite of the fact that naively evaluating this lower bound would require $O(n^k)$ operations, an efficient algorithm that exactly computes its value with only $O(k \log k)$ operations was proposed. Equipped with these new techniques, FACS was found to be successful in drastically reducing the number of false positives due to confounding in both synthetic data and genome-wide association studies for the plant model organism *A. thaliana*. Moreover, it appears that the ability to account for a categorical covariate does not come at the price of a loss of statistical power or computational efficiency: FACS was as effective as LAMP in detecting true associations and only required a small runtime overhead. Finally, despite being designed to handle a single categorical covariate, our experiments also suggested that a heuristic approach based on discretising low-

dimensional real-valued covariates can result in satisfactory performance. In particular, this approach was found to be successful in substantially reducing genomic inflation due to population structure in two *A. thaliana* genome-wide association studies.

By providing a principled way to discover statistically significant associations in enormous search spaces, significant pattern mining offers a unique opportunity to revisit classic biomarker discovery problems from a different perspective. In the last part of this thesis, we showcased this potential by proposing a new family of significant pattern mining-based approaches to carry out genome-wide association studies at a region level.

During the last decade, genome-wide association studies have become one of the most popular study designs for biomarker discovery, leading to the identification of more than 61,000 unique SNP-trait associations as of May 2018. Early genome-wide association studies have predominantly relied on univariate association tests. However, as the strongest SNP-trait associations are being found, detecting the remaining weaker associations, which might jointly explain a large proportion of phenotypic variation, is becoming increasingly challenging. This difficulty is particularly severe when studying rare variants, a regime in which unrealistically large sample sizes could be required for univariate tests to reach satisfactory statistical power. A popular approach to alleviate this limitation, pioneered by the statistical genetics community, is to assess the association between genotype and phenotype at a region level, rather than looking at each marker individually. In this way, multiple neighbouring markers can be pooled together, forming a joint meta-marker that, under the right conditions, might exhibit a stronger association with the phenotype than any of the markers in the region do in isolation. Genome-wide association studies at a region level have been successful in facilitating the discovery of associations that would have otherwise remained undetected by univariate approaches. However, state-of-the-art methods for this task, such as burden tests, are hindered by their inability to perform a large number of association tests. In a typical dataset with one million genomic markers, there are more than 500 billion genomic regions to be explored. In order to reduce the search space to a manageable size, burden tests require the analyst to prespecify a small subset of candidate regions *a priori*. In practice, the absence of strong prior knowledge to guide the definition of the set of regions to be tested leads to heuristic alternatives, such as only exploring regions of a certain length w . If the search space was misspecified by, for example, setting the hyperparameter w too small or too large, the resulting statistical power would plummet. Moreover, a data-driven choice for w or, more generally, the collection of regions that make up the search space, would necessitate either an additional correction for multiple comparisons or the use of a separate validation dataset. While burden tests require such prior assumptions to be computationally and statistically feasible, significant pattern mining allows attacking this problem from a fundamentally different angle. As a last contribution of this thesis, we explored the possibility to carry out genome-wide association studies at a region level by means of a new instance of significant pattern mining, which we denoted significant region mining. In particular, we proposed FAIS and FastCMH, two novel methods that utilise techniques from significant pattern mining to assess the statistical association of *all* genomic regions, regardless of their length or starting position, with a binary phenotype of interest. Formally, given a

search space $\mathcal{M} = \{\llbracket j_s, j_e \rrbracket \mid 1 \leq j_s \leq j_e \leq p\}$ comprising all possible genomic regions one could enumerate, FAIS aims to discover all regions $\mathcal{S} \in \mathcal{M}$ for which the binary random variable $G_{\mathcal{S}}(X)$ that indicates the presence or absence of one or more copies of a minor allele in \mathcal{S} is significantly associated with a binary phenotype Y . Motivated by the susceptibility of genome-wide association studies to confounding due to factors of variation such as population structure, FastCMH borrows ideas from our previous contribution FACS, extending FAIS to allow correcting for a categorical covariate C . From an algorithmic perspective, significant region mining can be understood as a restricted version of significant itemset mining. Consequently, FAIS and FastCMH reuse most of the algorithmic machinery described in this thesis, being heavily based on LAMP and FACS, respectively. In an exhaustive simulation study, our proposed approaches were shown to dramatically outperform univariate testing when the association signal is spread among multiple neighbouring markers, resulting in individually-weak associations. Most importantly, our results also demonstrated that significant region mining algorithms exhibit higher statistical power than burden tests whenever the length l of the truly associated regions is unknown and varies substantially throughout the genome. The potential of significant region mining to perform genome-wide association studies at a region level was further corroborated by an analysis of six real-world genome-wide association studies: five *A. thaliana* datasets and one study of Chronic Obstructive Pulmonary Disease (COPD) comprising two human cohorts. All in all, we found that more than half of all regions for which FastCMH discovered a statistically significant association would have been undetected by univariate testing, including some that have strong support in the literature. Many of these regions were also undetected by burden tests, owing to the restrictive definition of their search space. Nonetheless, burden tests also detected regions that were missed by our proposed approaches. More specifically, several regions that had a moderate association with the phenotype were deemed significant by burden tests but not by FastCMH, which necessitated a more stringent correction for multiple comparisons. In summary, we believe these results to be a strong indicator of the role that significant pattern mining could play in the broad field of biomarker discovery. By expanding the set of problem formulations which are computationally and statistically tractable, significant pattern mining should be considered a powerful complement to existing approaches rather than a substitute.

7.2 OUTLOOK

Before the introduction of the LAMP algorithm less than five years ago, accounting for the multiple comparisons problem in discriminative pattern mining was considered an unsolvable problem by many. Since then, the young field of significant pattern mining has experienced substantial progress. Recent developments, including the contributions described in this thesis, have made significant pattern mining algorithms faster, able to correct for confounding covariates, improved their statistical power and expanded the types of problems they can be applied to beyond itemset mining. In spite of all of these innovations, significant pattern mining still remains an exciting field, filled with open problems and avenues for future research awaiting to be explored.

Going beyond FWER control

Throughout this thesis, the FWER has been used as the unifying principle to account and correct for the multiple comparisons problem. Guaranteeing that the probability of reporting *any* false discoveries is bounded from above by a user-defined level α provides an unequivocally meaningful measure of confidence on the output of a significant pattern mining algorithm. Moreover, existing techniques to control the FWER, such as Tarone's method, have certain properties which are particularly appealing for their use in data mining. Firstly, they tend to require little processing time for each enumerated pattern and, as extensively discussed in this thesis, lend themselves relatively easily to the design of pruning criteria that can be exploited to drastically reduce the search space. Moreover, they are also extraordinarily general, guaranteeing strict FWER control regardless of the specific joint distribution of the test statistics. This is fundamentally important for significant pattern mining since, as studied in Chapter 4, the collection of test statistics corresponding to all patterns in the search space might exhibit extensive dependence structures that are not easily characterised. In spite of all these advantageous properties, FWER control is not devoid of limitations. In particular, the FWER has been known to be an overly conservative criterion for several decades, as can be readily understood from its definition. Strictly requiring that, with high probability, not a single error will be made might be adequate in critical applications with a low tolerance for errors. However, in many other exploratory tasks, reporting a small number of false discoveries might be an acceptable compromise in exchange for gaining statistical power. This observation has motivated statisticians to develop a wealth of alternative criteria to correct for the multiple comparisons problem.

FDR CONTROL : Controlling the False Discovery Rate (FDR) [106] is arguably the most popular alternative to FWER control, having achieved widespread adoption in the life sciences. Mathematically, the FDR is defined as $\text{FDR} = \mathbb{E} \left(\frac{\text{FP}}{\text{TP} + \text{FP}} \right)$ ¹, that is, as the expected False Discovery Proportion (FDP). By using a rate as an error measure, FDR control does not necessarily require a high probability of not producing false discoveries. Intuitively, a small number of false positives (FP) might be tolerated provided that the total number of discoveries (TP + FP) is large enough. In practice, this makes FDR control less stringent than FWER control, improving statistical power at the expense of an increase in the amount of Type I errors. This trade-off could be desirable in many potential applications of significant pattern mining, making the development of novel algorithms that control the FDR instead of the FWER one of the most promising research directions in the field. However, substantial methodological complications need to be overcome in order to reach this milestone.

Firstly, unlike the Bonferroni correction or Tarone's method, most existing techniques to control the FDR only allow for certain types of dependence between test statistics. For example, the Benjamini-Hochberg (BH) procedure [106], which is arguably the most popular approach to control the FDR, was initially only proven to be correct under the assumption that the test statistics are independent, a condition that is clearly violated in significant pattern mining. Follow-up work showed that the BH procedure

1. $\text{FDP} = \frac{\text{FP}}{\text{TP} + \text{FP}}$ is typically defined to be zero when no discoveries are made, i.e. when $\text{TP} + \text{FP} = 0$.

also controls the FDR under certain types of dependence [107, Theorem 1.2] and proposed an extension, often referred to as the Benjamini-Yekutieli (BY) procedure [107, Theorem 1.3], which is also provably correct under general dependence. However, both of these alternatives have shortcomings in the context of significant pattern mining. Verifying whether the types of dependence under which the BH procedure is valid apply to significant pattern mining or not still remains an open problem. Also, the BY procedure is known to be overly conservative in many situations, conflicting with the main motivation to substitute the FWER by the FDR as a criterion to account for the multiple comparisons problem. Other methods to control the FDR have been developed as an attempt to achieve a compromise between both extremes, being applicable to a broader set of types of dependence than the BH procedure while retaining high statistical power (e.g. [108–110]). Nevertheless, once again, whether the types of dependence these approaches consider are general enough to be used in significant pattern mining or not has not yet been shown.

Secondly, most existing procedures to control the FDR have been developed for applications involving a much smaller number of hypotheses than significant pattern mining. Designing an algorithm to make any of those procedures computationally tractable when trillions of simultaneous association tests need to be performed would be a remarkable contribution in its own right. In particular, many popular methods, including the BH and BY procedures, use a step-up strategy to find the set of hypotheses to be rejected. In a nutshell, these methods require computing the P-values corresponding to all hypotheses and sorting them in decreasing order. In other words, in order to assess the significance of the collection of tests, step-up procedures proceed from the least significant hypotheses to the most significant. Nonetheless, naively applying this strategy in the context of significant pattern mining would in principle require enumerating all patterns in the search space and computing their respective P-values, which would not be computationally feasible. Thus, step-down FDR-controlling procedures, even if less commonly known, might be easier to integrate into a significant pattern mining algorithm. Perhaps most importantly, a computationally tractable significant pattern mining algorithm using the FDR to account for the multiple comparisons problem will most likely necessitate a novel pruning criterion. A simple two-step combination of Tarone’s method and the BH procedure has been shown to successfully control the FDR [111]. However, deriving an equivalent notion of testability that is fully optimised for FDR control remains an open problem.

It is worth noting that the development of significant pattern mining algorithms based on the FDR has already begun to be explored. In particular, recent work [112] proposed a novel emerging pattern mining algorithm designed to control the FDR instead of the FWER. The main contribution in this article is the notion of *quasi-testability*, which essentially substitutes the implicit use of a Bonferroni correction to define testability by either the BH or the BY procedures. However, quasi-testability as defined in [112] requires estimating the number of patterns that would be deemed significant by the BH or BY procedures. In order to solve this difficulty, the authors resort to splitting the original dataset into a *calibration dataset*, used to estimate a quasi-testability threshold, and a *main dataset*, in which the significance of the patterns found to be testable according to the previous result is assessed using either the BH or the BY procedures. Despite the success of this approach, which has pioneered the use

of FDR control in significant pattern mining, there are several aspects which could be further improved in the future. Firstly, the use of data splitting is known to be inefficient in terms of statistical power and, most importantly, can lead to the output of the algorithm being unstable, hindering the interpretability of the results. Also, as previously discussed, the BH procedure might fail to control the FDR in pattern mining while the BY procedure might be overly conservative. Thus, we expect the development of FDR-based significant pattern mining algorithms to continue being a fruitful direction of research.

SELECTIVE INFERENCE: Post-selection inference, or *selective inference* for short, is perhaps one of the most innovative and active areas of research in multiple hypothesis testing [113]. Selective inference aims to devise techniques to carry out valid statistical inferences after model selection. Formally, consider a search space \mathcal{M} containing a collection of hypotheses to be tested and an algorithm $A : (\mathcal{X} \times \mathcal{Y})^n \rightarrow \mathcal{P}(\mathcal{M})$ that chooses a subset $A(\mathcal{D}) \subseteq \mathcal{M}$ of hypotheses based on the observed data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$. “Classical” P-values p_S computed for each hypothesis $S \in A(\mathcal{D})$ using the same data \mathcal{D} would no longer be correct; intuitively, using the targets $\{y_i\}_{i=1}^n$ for model selection leads to “overfitting”. The traditional solution to this problem is to either avoid model selection altogether or to split the dataset, using only a subset $\mathcal{D}_{\text{ms}} \subset \mathcal{D}$ of the data for model selection and computing the P-values on the remaining samples $\mathcal{D} \setminus \mathcal{D}_{\text{ms}}$. However, data splitting is an inefficient, overly conservative approach, causing a loss of statistical power. Moreover, different random splits might lead to distinct results, complicating the interpretation of any findings reported by the procedure. Selective inference studies how to account for the bias introduced by model selection, providing a principled, more efficient alternative to data splitting.

The set of selected hypotheses $A(\mathcal{D})$ can be modelled as a random variable taking values in $\mathcal{P}(\mathcal{M})$, with the n targets $\{Y_i\}_{i=1}^n$ being the source of randomness. The information about the targets conveyed by observing a certain outcome $A(\mathcal{D}) = \mathcal{M}_{\text{ms}}$ of the model selection procedure effectively reduces the sample space for the targets to the set $\{\{y_i\}_{i=1}^n \in \mathcal{Y}^n \mid A(\{(x_i, y_i)\}_{i=1}^n) = \mathcal{M}_{\text{ms}}\}$. Motivated by this observation, selective inference accounts for model selection by replacing the original null distribution of the test statistics $\Pr(T_S(\mathcal{D}) = t_S \mid H_0)$ by conditional null distributions of the form $\Pr(T_S(\mathcal{D}) = t_S \mid A(\mathcal{D}) = \mathcal{M}_{\text{ms}}, H_0)$. Interestingly, Tarone’s method can be understood as an extreme instance of selective inference. The model selection algorithm implicitly defined by Tarone’s concept of testability, $A(\mathcal{D}) = \mathcal{M}_{\text{test}}(\delta_{\text{tar}})$, depends on the targets only through $N_1 = \sum_{i=1}^n Y_i$. As all test statistics considered throughout this thesis treat the observed n_1 as a fixed quantity, it follows that $\Pr(T_S(\mathcal{D}) = t_S \mid A(\mathcal{D}) = \mathcal{M}_{\text{test}}(\delta_{\text{tar}}), N_1 = n_1, H_0) = \Pr(T_S(\mathcal{D}) = t_S \mid N_1 = n_1, H_0)$. In other words, when Tarone’s method is used alongside statistical association tests which consider the margin N_1 an ancillary statistic, there is no need to adjust the “classical” P-values to account for the use of testability for model selection. Nevertheless, for more complex model selection procedures, the conditional null distribution can differ drastically from the original null distribution. One of the most celebrated results in selective inference [114, Lemma 5.1 and Theorem 5.2] shows that, under the assumption that the targets $\mathbf{Y} = (Y_1, Y_2, \dots, Y_n)$ follow a multivariate normal

distribution and that the set $\{\mathbf{y} \in \mathbb{R}^n \mid A(\{(x_i, y_i)\}_{i=1}^n) = \mathcal{M}_{\text{ms}}\}$ can be described as a polyhedron $\{\mathbf{y} \in \mathbb{R}^n \mid \mathbf{C}\mathbf{y} \leq \mathbf{d}\}$ with $\mathbf{C} \in \mathbb{R}^{m \times n}$ and $\mathbf{d} \in \mathbb{R}^m$ independent of \mathbf{y} , P-values adjusted for model selection can be computed in closed-form based on a truncated normal distribution. In particular, these assumptions are sufficiently general to be applicable to ubiquitous model selection procedures such as the lasso, elastic net, marginal screening or forward step-wise regression, among others.

Recent work [115] has started investigating how to make use of selective inference in significant pattern mining. In principle, the framework proposed in [114] can be directly applied to pattern mining by treating each pattern occurrence indicator $g_S(x)$ as a different feature in an extremely high-dimensional linear model. However, the design of a practical algorithm implementing this idea is beset by the sheer size of the search space. More precisely, in the context of significant pattern mining, the polyhedron $\{\mathbf{y} \in \mathbb{R}^n \mid \mathbf{C}\mathbf{y} \leq \mathbf{d}\}$ representing the selection event would be described by trillions of inequalities, rendering a naive implementation of the method in [114] computationally intractable. The algorithm proposed in [115] circumvents this challenge by means of a novel search space pruning criterion, drastically speeding-up the computation of valid post-selection P-values in significant pattern mining. Nevertheless, selective inference is still a modern field which continues to evolve as it progressively overcomes its own limitations. For instance, all the derivations in [114] rely on certain restrictive assumptions, such as considering that the noise variance σ_e^2 is known *a priori* or that the hyperparameters of the model selection algorithm are fixed. Follow-up work (e.g. [116–120]) alleviates these shortcomings, opening the door to many exciting possibilities to develop novel significant pattern mining algorithms based on selective inference.

Accounting for heterogenous directions of effect

In several instances of significant pattern mining studied in this thesis, the pattern occurrence indicator $g_S(x)$ can be understood as a specific nonlinear function of a subset of binary features indexed by the pattern \mathcal{S} , i.e., $g_S(x) = f(\{u_j\}_{j \in \mathcal{S}})$ with $x = (u_1, u_2, \dots, u_p)$. For instance, significant itemset mining investigates multiplicative interactions of the form $f(\{u_j\}_{j \in \mathcal{S}}) = \prod_{j \in \mathcal{S}} u_j$ whereas significant region mining effectively uses a pooling transformation $f(\{u_j\}_{j \in \mathcal{S}}) = \max_{j \in \mathcal{S}} u_j$ after an appropriate encoding of the p original integer-valued features, $u_j = \mathbb{1}[x(j) > 0]$. Since the function which defines the pattern occurrence indicator $g_S(x)$ is chosen *a priori*, it is possible for a subset of features $\{U_j\}_{j \in \mathcal{S}}$ to be jointly statistically associated with the target Y even if the corresponding pattern occurrence indicator $G_S(X)$ is statistically independent of Y . Ultimately, this limitation might cause interesting associations to be missed.

Carrying out a nonparametric association test between the target Y and all feature subsets $\{\{U_j\}_{j \in \mathcal{S}} \mid \mathcal{S} \in \mathcal{M}\}$ might be unrealistic in the context of significant pattern mining. However, developing significant pattern mining algorithms which can adapt the function $g_S(x) = f(\{u_j\}_{j \in \mathcal{S}})$ defining the pattern occurrence indicator in a data-driven manner could be an impactful intermediate step in that direction. A common situation whereby an existing statistical association between $\{U_j\}_{j \in \mathcal{S}}$ and

Y might be missed by existing significant pattern mining algorithms occurs when the features indexed by \mathcal{S} exhibit *heterogeneous directions of effect*. That is, when a subset $\mathcal{S}_p \subset \mathcal{S}$ of the features are positively correlated with Y whereas the remaining features $\mathcal{S}_n = \mathcal{S} \setminus \mathcal{S}_p$ are negatively correlated. Intuitively, in this case the effects of the features indexed by \mathcal{S}_p and \mathcal{S}_n “cancel out” in the pattern occurrence indicator $G_{\mathcal{S}}(X)$ of pattern \mathcal{S} . If $G_{\mathcal{S}_p}(X)$ and $G_{\mathcal{S}_n}(X)$ were not associated sufficiently strongly with Y to be detected individually, the association signal would remain undiscovered by the significant pattern mining algorithm. This example is of critical importance, specially in significant region mining: as only contiguous subsets of features (regions) are considered, it could occur that the subsets \mathcal{S}_p and \mathcal{S}_n are never tested on their own, making it very unlikely for the algorithm to deem the region significantly associated. If, for each pattern \mathcal{S} , the subsets \mathcal{S}_p and \mathcal{S}_n could be identified, it would in principle be possible to define a pattern occurrence indicator that combines their effects in a constructive, rather than destructive manner, thereby alleviating this limitation.

Define $S_j = \text{sign}\left(A_j - r_j \frac{N_1}{n}\right) \in \{-1, 0, +1\}$ to be the *direction of effect* of feature $j \in \mathcal{S}$ estimated from the data. S_j is a random variable, as it depends on the targets $\{Y_i\}_{i=1}^n$ through $A_j = \sum_{i=1}^n u_{i,j} Y_i$. One simple possibility to make the pattern occurrence indicator flexible enough to account for heterogenous directions of effect is to consider functions of the form $G_{\mathcal{S}}(x) = f\left(\{u_j\}_{j \in \mathcal{S}} \mid \{S_j\}_{j \in \mathcal{S}}\right)$. Note that, in this case, the dependence of the pattern occurrence indicator $G_{\mathcal{S}}(x)$ on $\{S_j\}_{j \in \mathcal{S}}$ makes it a random variable as well, even if the observation $x = (u_1, u_2, \dots, u_p)$ was treated as fixed. For example, one such function $f\left(\{u_j\}_{j \in \mathcal{S}} \mid \{S_j\}_{j \in \mathcal{S}}\right)$ could be

$$G_{\mathcal{S}}(x) = \prod_{j \in \mathcal{S} \mid S_j = +1} u_j - \prod_{j \in \mathcal{S} \mid S_j = -1} u_j = G_{\mathcal{S}_p}(X) - G_{\mathcal{S}_n}(X).$$

However, the discussion which follows applies to any function that depends only on the targets $\{Y_i\}_{i=1}^n$ only through $\{S_j\}_{j \in \mathcal{S}}$ and $N_1 = \sum_{i=1}^n Y_i$. Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, define $A_{\mathcal{S}} = \sum_{i=1}^n G_{\mathcal{S}}(x_i) Y_i$ and $R_{\mathcal{S}} = \sum_{i=1}^n G_{\mathcal{S}}(x_i)$ analogously to Section 2.2. While it might be tempting to use the observed values of $a_{\mathcal{S}}$, $r_{\mathcal{S}}$, n_1 and n to perform an association test as discussed throughout this thesis, the fact that the pattern occurrence indicator $G_{\mathcal{S}}(x)$ depends on the targets $\{y_i\}_{i=1}^n$ through the observed directions of effect $\{s_j\}_{j \in \mathcal{S}}$ would render such an approach invalid. This setting is reminiscent of the problem of statistical inference after model selection and, as such, some of the core techniques of selective inference could be of great use.

We began to explore this connection in [121] where, provided that the sample size n is large enough to assume that the random vector $(A_{\mathcal{S}}, A_1, A_2, \dots, A_{|\mathcal{S}|})$ approximately follows a multivariate normal distribution, it was shown that the conditional null distribution

$$\Pr\left(A_{\mathcal{S}} = a_{\mathcal{S}} \mid \{S_j = s_j\}_{j \in \mathcal{S}}, N_1 = n_1, H_0\right)$$

can itself be satisfactorily approximated by a truncated normal distribution. Using this result, valid P-values that account for the fact that the pattern occurrence indicator $G_{\mathcal{S}}(X)$ was constructed with (partial) information about the targets $\{y_i\}_{i=1}^n$ can be computed in closed-form. Moreover, it was also shown in [121] that, for some specific functions $f\left(\{u_j\}_{j \in \mathcal{S}} \mid \{S_j\}_{j \in \mathcal{S}}\right)$ used to build the pattern occurrence indicator, a

minimum attainable P-value exists, allowing to use Tarone’s concept of testability. However, in order to incorporate this conceptual idea into a practical significant pattern mining algorithm, several key open problems remain to be solved. Firstly, computing Tarone’s minimum attainable P-value naively for the pattern occurrence indicators we considered is computationally intensive, with worst-case complexity exponential in the sample size n . While a lower bound on the exact minimum attainable P-value can be obtained by solving a linear program, the resulting approach might still be too computationally demanding for typical significant pattern mining problems. Perhaps most importantly, an effective pruning criterion has not yet been designed. Solving these challenges to propose the first significant pattern mining algorithm that can correct for heterogeneous directions of effect could be particularly impactful for applications such as genome-wide association studies. More generally, investigating the use of selective inference to allow for more expressive, data-driven functional forms for the pattern occurrence indicator is a promising avenue for future research.

Enhancing interpretability by identifying redundant significant patterns

The redundant nature of the search space in significant pattern mining has been a major theme in this thesis. One of its consequences, studied in great detail in Chapter 4, is to reduce the effective number of tests that need to be corrected for when accounting for the multiple comparisons problem. Algorithms which exploit this phenomenon, such as FastWY and Westfall-Young *light*, will exhibit greater statistical power than those which do not. However, the existence of strong statistical dependencies between test statistics also makes it extraordinarily difficult to differentiate between significantly associated patterns that correspond to original association signals and those whose association is merely mediated by a related pattern. In practice, this means that each individual association signal might be represented in the output of a significant pattern mining algorithm by hundreds or even thousands of heavily-redundant patterns, such as subsets or supersets of the (unknown) truly associated pattern. This phenomenon can seriously hinder the interpretability of the results, specially when compounded with statistical dependencies that might exist between the original features in the dataset.

In Chapter 6, we introduced a heuristic, *post hoc* approach to manage this problem in significant region mining. By simply grouping overlapping significantly associated genomic regions, the output can be represented as disjoint clusters, greatly aiding interpretability. Nevertheless, this naive solution is only possible due to the characteristics of the search space in significant region mining. In particular, any two non-overlapping, significantly associated regions \mathcal{S}_a and \mathcal{S}_b will belong to the same cluster if and only if there is a sequence $\mathcal{S}_a = \mathcal{S}_0, \mathcal{S}_1, \dots, \mathcal{S}_j = \mathcal{S}_b$ of significantly associated patterns satisfying $\mathcal{S}_i \cap \mathcal{S}_{i-1} \neq \emptyset$ for all $i = 1, \dots, j$. As this will seldom occur unless \mathcal{S}_a and \mathcal{S}_b lie close to each other in the genome, individual association signals tend to form disjoint clusters of overlapping, significantly associated regions. In contrast, in other instances of significant pattern mining, for any two significantly associated patterns \mathcal{S}_a and \mathcal{S}_b there are many other patterns \mathcal{S} in the search space which are related (e.g. have a sufficiently small Jaccard distance) to both \mathcal{S}_a and \mathcal{S}_b . As

a result, using the naive strategy described in Chapter 6 in this situation would often result in very few “super-clusters” that merge many individual association signals together.

Given a set $\mathcal{M}_{\text{sig,raw}}$ of significantly associated patterns, it could be argued that the gold standard *post hoc* pattern summarisation approach would aim to find a “minimal Markov blanket of patterns”, that is, a minimal set $\mathcal{M}_{\text{sig,mb}} \subseteq \mathcal{M}_{\text{sig,raw}}$ that satisfies $\Pr(Y = y \mid \{G_{\mathcal{S}}(x) = g_{\mathcal{S}}(x)\}_{\mathcal{S} \in \mathcal{M}_{\text{sig,mb}}}) = \Pr(Y = y \mid \{G_{\mathcal{S}}(x) = g_{\mathcal{S}}(x)\}_{\mathcal{S} \in \mathcal{M}_{\text{sig,raw}}})$. Ideally, the method would also be able assign each “redundant pattern” $\mathcal{S} \in \mathcal{M}_{\text{sig,raw}} \setminus \mathcal{M}_{\text{sig,mb}}$ to one or more “driver patterns” in $\mathcal{M}_{\text{sig,mb}}$ which mediate their association with the target Y . However, such a procedure might be unfeasible in practice, both computationally, as $|\mathcal{M}_{\text{sig,raw}}|$ could be rather large, and statistically, since the pattern occurrence indicators of distinct patterns in $\mathcal{M}_{\text{sig,raw}}$ might be too correlated for $\mathcal{M}_{\text{sig,mb}}$ to be identifiable. Thus, we believe that developing computationally tractable methods to summarise the output of significant pattern mining algorithms in a parsimonious, stable and principled manner would constitute an impactful avenue for future research, specially in the context of biomarker discovery where interpretability is essential.

Alternatively, an even more ambitious research direction to combat the effects of redundancy on the interpretability of significant pattern mining algorithms would be to modify the problem statement itself, preventing redundant patterns from ever being discovered. For example, one could conceive an algorithm which, akin to forward stepwise regression, iteratively aims to find the pattern \mathcal{S}_j whose pattern occurrence indicator $G_{\mathcal{S}_j}(X)$ is most strongly associated with the target Y given the presence or absence of all patterns $G_{\mathcal{S}_1}(X), G_{\mathcal{S}_2}(X), \dots, G_{\mathcal{S}_{j-1}}(X)$ selected in previous iterations. For sufficiently small j , the techniques described in Chapter 5 could be used to solve this problem at each iteration. However, a naive implementation of such an algorithm would have serious shortcomings. Firstly, in principle patterns would be enumerated from scratch at each iteration, greatly increasing runtime. Secondly, approaches based on stepwise regression are often unstable, with small perturbations of the original dataset potentially leading to a drastically different set of patterns being selected. Also, it is *a priori* unclear how to select a stopping criterion or how to condition on the set of patterns already included in the model if the resulting number of categories was too large relative to the sample size of the datasets. It is worth noting that the version of the selective inference-based approach in [115] that uses Orthogonal Matching Pursuit (OMP) [122] offers protection against redundancy to some extent as a result of the model selection stage. However, despite being a promising direction to obtain a parsimonious set of significantly associated patterns, this method also shares some of the aforementioned limitations, such as the absence of a principled scheme to select the hyperparameters of the model selection algorithm and the potential lack of stability of the set of selected patterns to small perturbations of the dataset. Therefore, the design of novel significant pattern mining algorithms to discover a minimal and stable set of patterns that maximally explain the variation in the target still remains an open problem.

Extending significant pattern mining to real-valued features

Discrete data is at the heart of significant pattern mining. When testing the association between two discrete random variables, there is only a finite number of distinct P-values that can be obtained and, consequently, there exists a minimum attainable P-value strictly greater than zero. That is the basis of Tarone's concept of testability and the essential principle all significant pattern mining algorithms hinge on. In contrast, statistical association tests for continuous random variables can in principle produce arbitrarily small P-values, making it impossible for any hypothesis to be untestable. This discrepancy between the behaviour of discrete and continuous random variables can also be intuitively understood through the lens of information theory. The mutual information $I(G_S(X); Y)$ between two discrete random variables $G_S(X)$ and Y is bounded from above by the entropy $H(G_S(X))$ of $G_S(X)$. Thus, in the context of significant pattern mining, if a pattern \mathcal{S} is too rare or too common, it cannot convey enough information about the target Y to be deemed significant. The situation for continuous random variables is, however, drastically different. If $G_S(X)$ and Y were both real-valued, it would be possible for $G_S(X)$ to have an arbitrarily small (differential) entropy while simultaneously making the mutual information between $G_S(X)$ and Y arbitrarily large. In essence, this phenomenon is a consequence of the fact that differential entropies can be negative whereas entropies cannot. For example, suppose that $(G_S(X), Y)$ followed a bivariate normal distribution with zero mean, $\text{Var}(G_S(X)) = \sigma_1^2$, $\text{Var}(Y) = \sigma_2^2$ and $\text{Cov}(G_S(X), Y) = \rho\sigma_1\sigma_2$. Then, the differential entropy $h(G_S(X)) = \frac{1}{2}\log(2\pi e\sigma_1^2)$ of $G_S(X)$ can be made as small as desired by letting σ_1 tend to 0 without affecting the mutual information, which is equal to $I(G_S(X); Y) = -\frac{1}{2}\log(1 - \rho^2)$ and can grow without bound as ρ tends to 1. Without the possibility to reduce the search space by means of Tarone's notion of testability, significant pattern mining on real-valued data might appear to be hopeless. On some occasions, there might be a natural way to discretise continuous features. For instance, many clinical variables can be summarised as either normal or abnormal and, in some cases, it might be possible to establish a minimum level of gene expression above which the gene can be considered to be active. More generally, if the distribution of a feature is markedly bimodal, median-based binarisation could provide a reasonable heuristic to analyse the data with any of the algorithms discussed throughout this thesis. Nevertheless, such an approach is undoubtedly unsatisfactory: in many cases, it might be preferable to maintain the more nuanced representation of the data which real-valued features provide or there might not be a natural way to discretise the data.

Recently, [123] proposed a simple yet innovative idea to define a notion of support for multiplicative interactions between continuous features, which is used as part of a frequent itemset mining algorithm for real-valued data. In order to capture a large amount of information from the original continuous features in a discrete representation, their method makes use of a probabilistic ensemble of many possible ways to binarise the dataset. More precisely, let $\mathcal{D} = \{x_i\}_{i=1}^n$ be a dataset with n i.i.d. observations $x = (v_1, v_2, \dots, v_p) \in \mathbb{R}^p$. Define $\mathbf{T} = (T_1, T_2, \dots, T_p) \in \mathbb{R}^p$ to be a set of p (random) binarisation thresholds and denote $X(\mathbf{T}) = (U_1(v_1, T_1), U_2(v_2, T_2), \dots, U_p(v_p, T_p)) \in \{0, 1\}^p$ the resulting binarised observation, where $U_j(v_j, T_j) = \mathbb{1}[v_j > T_j]$ is the j -th binarised feature. Given a

probability density function $\Pr(\mathbf{T} = \mathbf{t}) = \prod_{j=1}^p \Pr(T_j = t_j)$, [123] defines the support of an itemset $\mathcal{S} \subseteq \{1, 2, \dots, p\}$ in \mathcal{D} as the average of the support of \mathcal{S} in the ensemble of binarised datasets $\{\{x_i(\mathbf{t})\}_{i=1}^n \mid \mathbf{t} \in \mathbb{R}^p\}$, that is,

$$\begin{aligned} r_{\mathcal{S}} &= \mathbb{E}_{\mathbf{T}}(r_{\mathcal{S}}(\mathbf{T})) = \mathbb{E}_{\mathbf{T}}\left(\sum_{i=1}^n G_{\mathcal{S}}(X_i(\mathbf{T}))\right) = \sum_{i=1}^n \mathbb{E}_{\mathbf{T}}(G_{\mathcal{S}}(X_i(\mathbf{T}))) \\ &= \sum_{i=1}^n \mathbb{E}_{\mathbf{T}}\left(\prod_{j \in \mathcal{S}} U_{i,j}(v_{i,j}, T_j)\right) = \sum_{i=1}^n \prod_{j \in \mathcal{S}} \mathbb{E}_{T_j}(U_{i,j}(v_{i,j}, T_j)) = \sum_{i=1}^n \prod_{j \in \mathcal{S}} \Pr(T_j \leq v_{i,j}), \end{aligned}$$

where the second-to-last step follows from the assumption that the thresholds $\{T_j\}_{j=1}^n$ are jointly independent. Therefore, the support $r_{\mathcal{S}}$ of a real-valued itemset, as defined by [123], heavily depends on the set of distributions $\{\Pr(T_j = t_j)\}_{j=1}^p$ chosen by the user. More specifically, it depends on $\Pr(T_j = t_j)$ through evaluations of its cumulative distribution function $\Pr(T_j \leq t_j)$ at n different points, which correspond to the n realisations $\{v_{i,j}\}_{i=1}^n$ of the j -th feature in the dataset \mathcal{D} . Consequently, without loss of generality, it suffices to consider piecewise-constant distributions of the form

$$\Pr(T_j = t_j) = \begin{cases} p_{0,j}, & \text{if } t_j < v_{\pi(1),j}, \\ p_{1,j}, & \text{if } v_{\pi(1),j} \leq t_j < v_{\pi(2),j}, \\ \vdots & \vdots \\ p_{n-1,j}, & \text{if } v_{\pi(n-1),j} \leq t_j < v_{\pi(n),j}, \\ p_{n,j}, & \text{if } t_j \geq v_{\pi(n),j}, \end{cases}$$

where $\pi(i)$ denotes the index of the i -th smallest element in $\{v_{i,j}\}_{i=1}^n$ ². Moreover, note that any threshold strictly smaller than $v_{\pi(1),j}$ or greater or equal than $v_{\pi(n),j}$ would lead to the j -th binarised feature being constant across all observations. As either of these choices would result in no information being captured by the binary representation, it is plausible to further impose $p_{0,j} = p_{n,j} = 0$, effectively leaving $n - 2$ degrees of freedom to define the distribution $\Pr(T_j = t_j)$. Motivated by its desirable computational properties, the authors in [123] specifically propose to use a distribution that assigns an identical probability mass to the remaining $n - 1$ “threshold bins”, i.e., $p_{i,j} = \frac{1}{n-1} \frac{1}{v_{\pi(i+1),j} - v_{\pi(i),j}}$ for $1 \leq i \leq n - 1$. In particular, it can be readily shown that this choice implies that $\Pr(T_j \leq v_{i,j}) = \text{rank}(v_{i,j}; \mathcal{D})$, where $\text{rank}(v_{i,j}; \mathcal{D}) = \frac{\pi^{-1}(i)-1}{n-1}$ denotes the normalised rank of $v_{i,j}$ in $\{v_{i,j}\}_{i=1}^n$ and $\pi^{-1}(i)$ its absolute rank. As a result, the average pattern occurrence indicator $\mathbb{E}_{\mathbf{T}}(G_{\mathcal{S}}(X_i(\mathbf{T})))$ of an itemset \mathcal{S} in an observation x_i can be easily computed as the product of the normalised ranks of $\{v_{i,j}\}_{j \in \mathcal{S}}$, that is,

$$\mathbb{E}_{\mathbf{T}}(G_{\mathcal{S}}(X_i(\mathbf{T}))) = \prod_{j \in \mathcal{S}} \text{rank}(v_{i,j}; \mathcal{D}) \in [0, 1].$$

Most importantly, recent work [124] identified the potential that the concepts proposed by [123] have to make significant pattern mining for real-valued data

2. For the sake of simplicity, in this informal discussion we assume there are no ties.

possible. In a nutshell, a 2×2 contingency table analogous to those introduced in Section 2.2 can be constructed using $a_S = \sum_{i=1}^n y_i \prod_{j \in S} \text{rank}(v_{i,j}; \mathcal{D})$ and $r_S = \sum_{i=1}^n \prod_{j \in S} \text{rank}(v_{i,j}; \mathcal{D})$. The statistical association between an itemset S and the binary target Y can then be assessed based on the observed values of a_S , r_S , n_1 and n . Due to the way a_S and r_S are defined, they are not guaranteed to be integer-valued. To circumvent the difficulty that fractional counts pose for some statistical association tests, such as Fisher’s exact test, the authors proposed to use the G-test [125] instead. One of the most remarkable contributions of [124] is to prove that, when the G-test is used in conjunction with the notion of support introduced by [123], a minimum attainable P-value does exist and can be computed in closed-form in a computationally efficient manner. Moreover, [124] also showed that the resulting minimum attainable P-value function is monotonically decreasing for $r_S \leq \min(n_1, n - n_1)$ and that $S' \supseteq S$ implies $r_{S'} \leq r_S$ also when r_S is defined as in [123]. Thus, the same pruning criterion used by the LAMP algorithm (see Proposition 3.4) is valid for this new approach. Combining all these techniques, [124] proposed the first algorithm able to assess the statistical association of *all* high-order multiplicative interactions of continuous features with a binary target of interest while guaranteeing strict FWER control.

Consider a naive baseline algorithm, consisting of arbitrarily choosing a single set of thresholds \mathbf{t} to obtain a binary representation $\{x_i(\mathbf{t})\}_{i=1}^n$ of the input observations. Using such a method, the G-test P-value corresponding to an itemset S would be given by $p_G(a_S(\mathbf{t}) \mid n, n_1, r_S(\mathbf{t}))$. On many occasions, using a unique value for \mathbf{t} chosen *a priori*, perhaps by means of a heuristic such as median-based binarisation, will fail to capture an existing association between the multiplicative interaction of the features indexed by S and the target of interest Y . In order to obtain a more expressive model, the novel approach in [124] effectively computes each P-value as

$$p_G(\mathbb{E}_{\mathbf{T}}(a_S(\mathbf{T})) \mid n, n_1, \mathbb{E}_{\mathbf{T}}(r_S(\mathbf{T}))),$$

where, intuitively, the average values $a_S = \mathbb{E}_{\mathbf{T}}(a_S(\mathbf{T}))$ and $r_S = \mathbb{E}_{\mathbf{T}}(r_S(\mathbf{T}))$ across a probabilistic ensemble of thresholds provide a more faithful representation of the original data. The success of this approach, which pioneered the development of significant pattern mining algorithms for real-valued data, inevitably poses several fundamental questions.

Firstly, the P-values computed using the method in [124] strongly depend on the distribution of the binarisation thresholds. While a uniform distribution on the $n - 1$ “threshold bins”, as proposed in [123], might be a reasonable starting point, it is plausible that other distributions could lead to higher statistical power or better computational properties. Perhaps most interestingly, the aforementioned comparison with a naive baseline suggests a third, still unexplored approach. Rather than evaluating the test statistic at the average values of $a_S(\mathbf{T})$ and $r_S(\mathbf{T})$, i.e. defining $T_S = T_G(\mathbb{E}_{\mathbf{T}}(a_S(\mathbf{T})) \mid n, n_1, \mathbb{E}_{\mathbf{T}}(r_S(\mathbf{T})))$, one could compute instead the average value of the test statistic, that is,

$$T_S = \mathbb{E}_{\mathbf{T}}(T_G(a_S(\mathbf{T}) \mid n, n_1, r_S(\mathbf{T}))),$$

from which a P-value could then be simply obtained as $p_S = 1 - F_{\chi_1^2}(T_S)$ ³. This alternative approach might be advantageous in some situations. For example, consider a toy problem where a uniformly-distributed feature $V_j \sim U(0, 1)$ is single-handedly responsible for the association with the target Y according to the model

$$\Pr(Y = 1 \mid v_j) = \begin{cases} 1 - \epsilon, & \text{if } v_j \leq \frac{1}{4}, \\ \epsilon, & \text{if } \frac{1}{4} < v_j \leq \frac{3}{4}, \\ 1 - \epsilon, & \text{if } v_j > \frac{3}{4}. \end{cases}$$

As ϵ tends to zero, the association between V_j and Y can be made arbitrarily strong, up to the point where Y becomes a deterministic function of the input feature. However, evaluating the G-test T_G at $\mathbb{E}_{\mathbf{T}}(a_S(\mathbf{T}))$ and $\mathbb{E}_{\mathbf{T}}(r_S(\mathbf{T}))$ would fail to reveal this association. In contrast, there exists a range of values for \mathbf{t} with high probability under $\Pr(\mathbf{T} = \mathbf{t})$ for which $T_G(a_S(\mathbf{t}) \mid n, n_1, r_S(\mathbf{t}))$ is rather large. Consequently, the average value $\mathbb{E}_{\mathbf{T}}(T_G(a_S(\mathbf{T}) \mid n, n_1, r_S(\mathbf{T})))$ of the test statistic T_G will be sufficiently large for the association to be discovered. Perhaps most importantly, a minimum attainable P-value exists for this alternative approach as well; since for each $(n, n_1, r_S(\mathbf{t}))$ the G-test attains a finite maximum, its average over the distribution $\Pr(\mathbf{T} = \mathbf{t})$ must be finite too. Whether the resulting minimum attainable P-values are sufficiently large to be practically useful alongside Tarone's method or not will depend on the induced distribution of $r_S(\mathbf{T})$. However, it would in principle be possible to tweak the distribution $\Pr(\mathbf{T} = \mathbf{t})$ to ensure an adequate interplay with Tarone's method. Nonetheless, investigating the effectiveness of these ideas as part of a significant pattern mining algorithm for real-valued data remains an open problem. In particular, developing a computationally efficient algorithm to evaluate the resulting minimum attainable P-values and designing a valid pruning criterion are some of the most immediate hurdles in this direction. Finally, one could also consider a fourth, arguably more ambitious alternative: substituting the averaging over a distribution $\Pr(\mathbf{T} = \mathbf{t})$ by a maximisation over $\mathbf{t} \in \mathbb{R}^p$, using selective inference to account for the effects of model selection, that is,

$$T_S = \max_{\mathbf{t} \in \mathbb{R}^p} T_G(a_S(\mathbf{t}) \mid n, n_1, r_S(\mathbf{t})),$$

where P-values adjusted for the fact that the optimal set of thresholds \mathbf{t}^* was obtained using the input dataset \mathcal{D} could be derived based on a conditional null distribution of the form $\Pr(T_S(\mathcal{D}) = t_S \mid \mathbf{T}^*(\mathcal{D}) = \mathbf{t}^*, H_0)$. Key issues such as exploring how to efficiently compute those P-values, studying if a minimum attainable P-value exists for this test statistic and can be used effectively alongside Tarone's method, developing a valid pruning criterion or investigating whether the set of thresholds \mathbf{t}^* should be optimised in a per-pattern basis or, rather, jointly for all patterns, are all exciting open problems to be considered. More complex binarisation schemes than the use of a single threshold per feature as a splitting point could be researched next. All in

3. Under the null hypothesis $H_0 : \{V_j\}_{j \in S} \perp Y$, any function of $\{V_j\}_{j \in S}$ is statistically independent of the target Y , including $G_S(X(\mathbf{t})) = \prod_{j \in S} \mathbb{1}[V_j > t_j]$ for any $\mathbf{t} \in \mathbb{R}^p$. Thus, $T_G(a_S(\mathbf{t}) \mid n, n_1, r_S(\mathbf{t}))$ asymptotically follows a χ_1^2 distribution for all \mathbf{t} and so does the average test statistic $\mathbb{E}_{\mathbf{T}}(T_G(a_S(\mathbf{T}) \mid n, n_1, r_S(\mathbf{T})))$.

all, significant pattern mining for real-valued datasets is definitely one of the next frontiers in the field, and we hope to see a wealth of new approaches being proposed within the next few years.

Speeding-up significant pattern mining algorithms

Since the inception of LAMP, follow-up work [65, 66] has refined the algorithmic interplay between Tarone’s method and pattern enumeration, leading to drastic improvements in the computational efficiency of significant pattern mining algorithms. Nevertheless, significant pattern mining remains inexorably linked to frequent pattern mining, which is well-known to be an extraordinarily computationally demanding problem. As a result, existing methods still struggle to analyse large-scale datasets, particularly those which are not sparse. This limitation severely hinders the applicability of significant pattern mining in multiple biomarker discovery tasks. For example, datasets originating from genome-wide association studies might comprise millions of features, a large proportion of which could be rather dense, with typical sparsity ratios in the range [0.15, 0.5].

A possibility to further speed-up significant pattern mining algorithms and broaden the scope of problems they can be applied to is to redesign them to take advantage of modern computing resources. In particular, parallelising existing significant pattern mining algorithms is a key topic for future work. Due to the sequential nature of pattern enumeration and the tendency of depth-first search to create highly-unbalanced enumeration trees, naive strategies to distribute the workload are unlikely to exhibit a satisfactory performance. Recent work [89] started studying this subject, leading to the MP-LAMP algorithm, a parallel implementation of LAMP 2.0 which utilises global load balancing [126] to achieve a remarkable efficiency; empirically, up to 1,175-fold speed-ups were observed when using 1,200 cores. Additionally, [127] explored a GPU-based implementation of the Westfall-Young permutation testing procedure for significant itemset mining in the context of genome-wide association studies, reporting up to 619-fold speed-ups compared to a naive baseline implementation.

Alternatively, the redundant nature of the search space in significant pattern mining might provide a complementary opportunity to reduce the runtime of existing approaches. Intuitively, it could be possible to ascertain the significance of many patterns to a reasonable degree of accuracy without ever computing their test statistics. Instead, related patterns might be used to impute the missing test statistics, as it is often done in univariate genome-wide association studies by exploiting linkage disequilibrium (e.g. [128]). A hypothetical approach which successfully exploited this idea could devise a more aggressive form of search space pruning, effectively allowing to reduce runtime at the expense of possibly having a non-zero number of false negatives among the pruned patterns. Investigating to which extent such a technique could speed-up significant pattern mining while maintaining high statistical power still remains an open problem.

Circumventing existing flaws of Tarone’s method

As extensively discussed throughout this thesis, Tarone’s method is omnipresent in significant pattern mining. Despite its many virtues, the concept of testability also has

some arguably undesirable properties. Perhaps the most impactful of all is its inability to exploit the dependence between test statistics, a limitation which was the main subject of study of Chapter 4 and culminated in the design of the Westfall-Young light algorithm. However, other negative aspects of Tarone's method have not been addressed yet in the context of significant pattern mining.

A particularly unintuitive property of Tarone's method is its lack of monotonicity with respect to the target FWER α . In other words, it is entirely possible for a pattern to be deemed significantly associated at level α and yet fail to achieve significance at a less stringent target FWER $\alpha' > \alpha$. In essence, this can occur whenever the ratio between α' and α is exceeded by the ratio between the number of testable patterns at levels α' and α . This problem has long been known, and extensions of Tarone's method which are monotonic on the target FWER have been proposed [129, 130]. However, these have not been yet incorporated into a significant pattern mining algorithm.

Another aspect of Tarone's method which can be subjected to criticism is its reliance on a dichotomous interpretation of P-values. In a nutshell, P-values quantify how well the probabilistic model postulated under the null hypothesis fits the observed data, thus being an indirect, continuous measure of statistical significance. However, merely reporting whether a null hypothesis was accepted or rejected provides a much less informative description of the results than the exact P-value. For example, consider a toy problem in which three hypotheses are being tested at a certain a corrected significance threshold δ , with corresponding P-values $p_1 = \frac{1}{1000}\delta$, $p_2 = \frac{999}{100}\delta$ and $p_3 = \frac{1001}{1000}\delta$. In this situation, the data clearly does not support the first null hypothesis while the evidence against the second and third null hypotheses is inconclusive. Most importantly, the difference in significance between these two last tests is clearly inconsequential in practice. Thus, simply stating that the two first null hypotheses were rejected while the third was accepted is arguably misleading; doing so would implicitly overstate the significance of the second test while downplaying the significance of the first test, for which the evidence against the null hypothesis is much stronger, and third test, whose significance is virtually equivalent to that of the second test. This phenomenon is well understood, being one of the main reasons why statisticians strongly recommend always reporting exact P-values, alongside effect sizes and confidence intervals. Unfortunately, Tarone's concept of testability is defined explicitly in terms of a binary interpretation of P-values; a hypothesis is testable if and only if its P-value could possibly be smaller or equal than δ . Even though this aspect of testability might be philosophically unappealing, it only has a minor practical impact in significant pattern mining. Indeed, the primary role of significant pattern mining is not to be a standalone, definitive method to validate scientific findings but, rather, to provide a powerful tool for data exploration. As such, these algorithms should be used as part of a fluid process, complemented with both prior knowledge and follow-up studies.

Incorporating different forms of prior knowledge

Finally, another interesting avenue for future research is the possibility to incorporate certain types of domain knowledge by restricting the set of patterns which form the search space.

Significant region mining can be considered to be a representative example of this idea. Rather than attempting to aggregate the (possibly weak) effects of any subset of markers, only contiguous subsets of features are considered. This is motivated by the phenomenon of genetic heterogeneity, whereby it is postulated that neighbouring genomic variants could affect the phenotype in a similar manner, justifying pooling contiguous markers into a joint meta-marker.

Other ubiquitous form of prior knowledge in the context of biomarker discovery are biological networks, such as co-expression networks [131], pathways [132] or protein-protein interaction networks [133], among others. Formally, all of these can be described as a graph where each node corresponds to a different feature and edges represent the existence of a relationship between two features. This structure could be exploited to reduce the set of patterns in the search space, for example, by enumerating only subsets of features which form a connected subgraph in the network. Other heuristic alternatives, such as exploring shortest paths or a certain number of random walks could also be investigated.

In general, the use of prior knowledge can drastically reduce the number of candidate patterns in the search space. This can greatly improve computational efficiency and, if the assumptions turn out to be appropriate, it will also translate into a gain in statistical power.

7.3 CLOSING REMARKS

In recent years, vast improvements in data availability and computing power have spurred unprecedented progress in data science. Machine learning is at the forefront of these developments, with deep learning-based methods in particular revolutionising fields such as computer vision, speech recognition, natural language processing or reinforcement learning [134]. Many of these problems are characterised by high signal-to-noise ratios and the availability of a large number of training samples; the key challenge being to design predictive models which can efficiently learn the extraordinarily complex structures present in the training data in a way that generalises to unseen observations. This revolution has also transformed biomarker discovery, albeit in a different direction. While sample sizes are indeed getting bigger, so is the number of measurements (features) being acquired. The resulting datasets are often extremely high-dimensional, can have large amounts of noise in the explanatory variables and/or labels and, more generally, might not contain all the information necessary to build an accurate predictive model. In these settings, machine learning has been poised to become an integral component of the scientific method in the big data era, providing domain experts with the necessary tools to navigate large-scale, noisy datasets in search for salient patterns that can be further investigated in follow-up studies. As other modern developments in statistical methodology, such as sparsity-inducing regularisation, false discovery rate control, empirical Bayes estimation or post-selection inference, the emergence of significant pattern mining is a response to this outstanding challenge. By combining the search capabilities of discriminative pattern mining with a rigorous correction for the multiple comparisons problem, we hope that significant pattern mining will provide researchers in the life sciences and other domains with a valuable set of tools for knowledge discovery.

SUMMARY AND OUTLOOK

While still being a young field, the substantial progress it has experienced in the last five years, coupled with exciting new developments in related disciplines, strongly suggest that significant pattern mining will continue to be a fruitful research topic for many years to come.

Part IV

APPENDICES

CHAPTER SUMMARIES

SUMMARY OF CHAPTER 2

Problem Statement and Terminology

- *Significant pattern mining* provides tools to explore datasets $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ containing n labeled observations (x, y) , where x exists in a discrete input domain \mathcal{X} and y is a binary class label.
- A *pattern* \mathcal{S} is informally defined to be any discrete substructure of the input domain \mathcal{X} . The *pattern occurrence indicator* $g_{\mathcal{S}}(x)$ takes value 1 if pattern \mathcal{S} occurs in a sample $x \in \mathcal{X}$ and value 0 otherwise.
- Given a search space of candidate patterns \mathcal{M} , significant pattern mining aims to find discriminative patterns $\mathcal{S} \in \mathcal{M}$ for which the class labels Y and the pattern occurrence indicator $G_{\mathcal{S}}(X)$ are *statistically dependent (or associated)* random variables.
- Specific instances of significant pattern mining that will be covered in this thesis include:
 - (i) *Significant itemset mining*, which deals with p -dimensional binary vectors ($\mathcal{X} = \{0, 1\}^p$). A pattern \mathcal{S} indexes a subset of the p binary features and is defined to occur in a sample $x = (u_1, u_2, \dots, u_p)$ if the multiplicative feature interaction $z_{\mathcal{S}}(x) = \prod_{j \in \mathcal{S}} u_j$ it induces takes value 1, i.e. $g_{\mathcal{S}}(x) = z_{\mathcal{S}}(x)$.
 - (ii) *Significant subgraph mining*, which operates on graphs with categorical labels ($\mathcal{X} = \{x \mid x = (V, E, l_{V,E})\}$, where V is a set of nodes, $E \subseteq V \times V$ a set of edges and $l_{V,E} : V \cup E \rightarrow \Sigma_{V,E}$ a function labelling nodes and edges with a categorical label from an alphabet $\Sigma_{V,E}$). Each pattern \mathcal{S} is a different induced subgraph of a sample $x \in \mathcal{X}$ in the input dataset \mathcal{D} and $g_{\mathcal{S}}(x) = 1$ if and only if x contains a subgraph isomorphic to \mathcal{S} .

Statistical Association Testing in Significant Pattern Mining

- A *test statistic* $T: \{(g_{\mathcal{S}}(x_i), y_i)\}_{i=1}^n \rightarrow \mathbb{R}$ maps the set of n realisations of $(G_{\mathcal{S}}(X), Y)$ obtained from dataset \mathcal{D} to a scalar $t_{\mathcal{S}}$ that quantifies the statistical association between the random variables $G_{\mathcal{S}}(X)$ and Y .
- To account for the inherent uncertainty in the determination of $t_{\mathcal{S}}$ based on a finite sample $\{(g_{\mathcal{S}}(x_i), y_i)\}_{i=1}^n$, a *P-value* $p_{\mathcal{S}}$ is computed as the probability of observing a value of T more extreme than $t_{\mathcal{S}}$, i.e. a value indicative of an association at least as strong, under the *null hypothesis* that $G_{\mathcal{S}}(X)$ and Y are statistically independent.
- The random variables Y and $G_{\mathcal{S}}(X)$ will be deemed *significantly associated* if the P-value $p_{\mathcal{S}}$ falls below a *significance threshold* δ .

- The significance threshold δ determines the trade-off between *type I error* (probability to erroneously report associations that do not exist) and *statistical power* (probability to successfully detect associations that do exist).
- Pearson's χ^2 test [44] and Fisher's exact test [45] are extensively used in significant pattern mining, as they are popular techniques to test the statistical association of two binary random variables such as $G_S(X)$ and Y .
- For both Pearson's χ^2 test and Fisher's exact test, the resulting P-value p_S can be expressed as a function $p_S(a_S | n, n_1, r_S)$ of (i) a_S , the number of samples in class $y = 1$ which contain pattern S ; (ii) r_S , the total number of samples which contain pattern S ; (iii) n_1 , the number of samples in class $y = 1$ and (iv) n , the total number of samples in the dataset.

The Multiple Comparisons Problem

- Significant pattern mining requires performing an enormous number $|\mathcal{M}|$ of association tests.
- Controlling the type I error of each individual test at level α would produce on average $\alpha|\mathcal{M}_0|$ false positives, where $\mathcal{M}_0 \subseteq \mathcal{M}$ is the set of *null patterns* for which $G_S(X)$ and Y are statistically independent.
- Whenever $|\mathcal{M}_0|$ is a large number, as is the case for significant pattern mining, this strategy would result in an astounding number of false positives, compromising the reliability of the results. To circumvent this so-called *multiple comparisons problem*, error measures that consider the entire collection of association tests simultaneously are necessary.
- The *Family-Wise Error Rate (FWER)*, defined as the probability of producing any number of false positives in the entire body of tests, is one such measure. Formally, $\text{FWER}(\delta) = \Pr(\text{FP}(\delta) > 0)$, where $\text{FP}(\delta)$ is the number of false positives at significance threshold δ . Controlling the FWER at level α , rather than the individual type I error of each test, is a widespread approach to correct for the multiple comparisons problem.
- The *Bonferroni correction* [60, 61] achieves FWER control at level α by using an over-conservative significance threshold defined as $\delta_{\text{bonf}} = \max\{\delta \mid \delta|\mathcal{M}| \leq \alpha\} = \alpha/|\mathcal{M}|$. While being extremely popular due to its extraordinary simplicity, the Bonferroni correction is too conservative to cope with the enormous number $|\mathcal{M}|$ of association tests that significant pattern mining requires; the resulting statistical power would be too low for all practical purposes.
- *Tarone's improved Bonferroni correction for discrete data* [64] exploits the fact that, for some discrete test statistics such as Pearson's χ^2 test and Fisher's exact test, there exists a *minimum attainable P-value* $p_{S,\min}$.
- By definition, if $p_{S,\min} > \delta$ for a pattern S , the random variables $G_S(X)$ and Y cannot possibly be deemed significantly associated. Therefore, these *untestable patterns* could also never result in a false positive.
- It can be proven that untestable patterns do not need to be taken into consideration to achieve FWER control. Tarone's method makes use of this, propos-

ing a significance threshold defined as $\delta_{\text{tar}} = \max \{ \delta \mid \delta |\mathcal{M}_{\text{test}}(\delta)| \leq \alpha \}$, where $\mathcal{M}_{\text{test}}(\delta) = \{ \mathcal{S} \in \mathcal{M} \mid p_{\mathcal{S}, \text{min}} \leq \delta \}$ is the set of *testable patterns* at level δ .

- In significant pattern mining, a large proportion of all candidate patterns is often untestable, i.e. $|\mathcal{M}_{\text{test}}(\delta_{\text{tar}})| \ll |\mathcal{M}|$. Therefore, Tarone's method dramatically improves statistical power over a standard Bonferroni correction. Nevertheless, computing δ_{tar} is a challenging problem that requires developing data mining algorithms to efficiently explore the search space \mathcal{M} .

SUMMARY OF CHAPTER 3

- One of the main algorithmic challenges in significant pattern mining is to efficiently compute Tarone’s corrected significance threshold δ_{tar} , given by the largest δ that satisfies $\delta|\mathcal{M}_{\text{test}}(\delta)| \leq \alpha$.
- A brute-force approach to compute the number $|\mathcal{M}_{\text{test}}(\delta)|$ of testable patterns at level δ would be computationally intractable, as it would require computing the minimum attainable P-value $p_{\mathcal{S},\text{min}}$ of each pattern \mathcal{S} in the search space \mathcal{M} .
- The Limitless-Arity Multiple-testing Procedure (LAMP) algorithm, proposed in 2013 [26], was the first method to successfully solve this problem, allowing to use Tarone’s method to correct for the multiple comparisons problem in pattern mining. Recent work [65, 66] has further improved the original LAMP algorithm, leading to even faster significant pattern mining methods that are nevertheless based on the same core principles. Throughout this thesis, that enhanced version of the LAMP algorithm will be informally denoted as LAMP 2.0.

Overview

- Significant pattern mining algorithms derived from LAMP usually proceed in two phases:
 - (i) The first and most critical phase is responsible for exploring the search space of candidate patterns \mathcal{M} to efficiently compute Tarone’s corrected significance threshold δ_{tar} and retrieve the corresponding set of testable patterns $\mathcal{M}_{\text{test}}(\delta)$.
 - (ii) The second phase applies a test statistic of choice to compute a P-value $p_{\mathcal{S}}$ for each testable pattern $\mathcal{S} \in \mathcal{M}_{\text{test}}(\delta_{\text{tar}})$, returning those that are significantly associated at level δ_{tar} .

Pattern enumeration

- Significant pattern mining algorithms explore the search space of candidate patterns \mathcal{M} by arranging each pattern $\mathcal{S} \in \mathcal{M}$ as a node of a *pattern enumeration tree*.
- A valid pattern enumeration tree is any bijective mapping of patterns to nodes of a tree such that the descendants \mathcal{S}' of a pattern \mathcal{S} are all super-patterns of \mathcal{S} , i.e., $\mathcal{S} \subseteq \mathcal{S}'$.
- A direct consequence of enumerating patterns by traversing a pattern enumeration tree is the *apriori property* of pattern mining, which states that the number $r_{\mathcal{S}}$ of occurrences of a pattern \mathcal{S} in an input dataset \mathcal{D} must be larger or equal than the number $r_{\mathcal{S}'}$ of occurrences of any pattern \mathcal{S}' which is a descendant of \mathcal{S} in the pattern enumeration tree.
- State-of-the-art significant pattern mining algorithms compute Tarone’s corrected significance threshold δ_{tar} and the set of testable patterns $\mathcal{M}_{\text{test}}(\delta)$ exactly by iteratively adjusting estimates $\hat{\delta}_{\text{tar}}$ and $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$ as patterns are enumerated during a depth-first traversal of the pattern enumeration tree.

Evaluating Tarone's minimum attainable P-value

- Significant pattern mining methods require computing the minimum attainable P-value $p_{\mathcal{S},\min}$ of every single pattern \mathcal{S} that is enumerated by the algorithm, making the computational efficiency of this step critical.
- For most statistical association tests typically used in significant pattern mining, computing the minimum attainable P-value $p_{\mathcal{S},\min}$ is equivalent to finding the cell-count value $a_{\mathcal{S}}^* \in \llbracket \max(0, r_{\mathcal{S}} - (n - n_1)), \min(n_1, r_{\mathcal{S}}) \rrbracket$ that minimises the P-value $p_{\mathcal{S}}(a_{\mathcal{S}}' | n, n_1, r_{\mathcal{S}})$ that would result as an outcome of applying the test statistic to a 2×2 contingency table with cell count $a_{\mathcal{S}}'$, margins n_1 and $r_{\mathcal{S}}$ and sample size n .
- Solving that optimisation problem by evaluating all P-values $p_{\mathcal{S}}(a_{\mathcal{S}}' | n, n_1, r_{\mathcal{S}})$ for $a_{\mathcal{S}}' \in \llbracket \max(0, r_{\mathcal{S}} - (n - n_1)), \min(n_1, r_{\mathcal{S}}) \rrbracket$ would require $O(n)$ operations, leading to an unacceptable computational overhead.
- However, for both Pearson's χ^2 test and Fisher's exact test, closed-form expressions of $p_{\mathcal{S},\min}$ that can be evaluated with $O(1)$ complexity are available.
- The minimum attainable P-value $p_{\mathcal{S},\min}$ of a pattern \mathcal{S} is a function of the margins n_1 and $r_{\mathcal{S}}$ and sample size n . However, since n_1 and n are constant for all patterns in a dataset \mathcal{D} , the dependence on these parameters can be treated implicitly. This allows studying the minimum attainable P-value $p_{\mathcal{S},\min}$ as a univariate function $p_{\min}(r_{\mathcal{S}})$ of the number $r_{\mathcal{S}}$ of occurrences of pattern \mathcal{S} in the dataset \mathcal{D} .
- Qualitatively, the minimum attainable P-value $p_{\min}(r_{\mathcal{S}})$ is large whenever $r_{\mathcal{S}}$ is small or $r_{\mathcal{S}}$ is large. This formalises the intuition that patterns \mathcal{S} that are either too rare or too common in a dataset \mathcal{D} are less likely to be significantly associated.
- For sufficiently large values of the corrected significance threshold δ , there exists $r_{\min}(\delta)$ such that a pattern \mathcal{S} is testable at level δ if and only if $r_{\mathcal{S}}$ lies in $\llbracket r_{\min}(\delta), n - r_{\min}(\delta) \rrbracket$.

Designing a pruning condition

- Irregardless of how efficient the evaluation of the minimum attainable P-value function $p_{\min}(r_{\mathcal{S}})$ is, enumerating every single pattern in the search space is computationally intractable.
- A key component of significant pattern mining algorithms is a *pruning criterion* to test whether the descendants \mathcal{S}' of a pattern \mathcal{S} are testable or not based only on information present in the contingency table of pattern \mathcal{S} .
- The minimum attainable P-value function $p_{\min}(r_{\mathcal{S}})$ for both Pearson's χ^2 test and Fisher's exact test is monotonically decreasing on $r_{\mathcal{S}}$ in the range $\llbracket 0, \min(n_1, n - n_1) \rrbracket$. Combining this with the apriori property of pattern mining, it follows that if a pattern \mathcal{S} satisfying $r_{\mathcal{S}} \leq \min(n_1, n - n_1)$ is untestable, then all of its descendants \mathcal{S}' in the pattern enumeration tree must be untestable as well and can be pruned from the search space.

- This pruning criterion can be readily applied in significant pattern mining as part of the recursive traversal of the pattern enumeration tree; only if the pruning condition does *not* apply for the pattern \mathcal{S} currently being processed by the algorithm will patterns $\mathcal{S}' \in \text{Children}(\mathcal{S})$ be enumerated by the algorithm next.

Implementation considerations

- Important aspects to consider prior to implementing a significant pattern mining algorithm include:
 - (i) *Choosing an efficient algorithm to construct and navigate the pattern enumeration tree.* The design of efficient algorithms to enumerate patterns has been widely studied in data mining, leading to a myriad of existing approaches that can be readily used in significant pattern mining.
 - (ii) *Choosing a strategy to iteratively refine the estimate $\hat{\delta}_{\text{tar}}$ of Tarone's corrected significance threshold.* While approaches based on grid search are valid, the discreteness of the minimum attainable P-value function $p_{\min}(r_{\mathcal{S}})$ can be exploited to devise a more efficient strategy to decrease $\hat{\delta}_{\text{tar}}$ whenever necessary.
 - (iii) *Avoiding the need to keep the estimate $\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})$ of the set of testable patterns at level $\hat{\delta}_{\text{tar}}$ in memory during the execution of the algorithm.* In order to compute $\hat{\delta}_{\text{tar}}$, only the total number $|\widehat{\mathcal{M}}_{\text{test}}(\hat{\delta}_{\text{tar}})|$ of testable patterns is necessary. This allows using a two-step strategy: a first pattern enumeration pass is used to compute $\hat{\delta}_{\text{tar}}$ while a second, subsequent pass is dedicated to find which testable patterns are significantly associated at level $\hat{\delta}_{\text{tar}}$. This strategy requires enumerating patterns twice, approximately doubling runtime, but drastically reduces memory usage, being therefore often preferred in practice.

SUMMARY OF CHAPTER 4

- The search space of candidate patterns \mathcal{M} in significant pattern mining is not only enormous, but also harbours extensive dependencies between the many patterns it contains.
- This chapter is devoted to introduce *Westfall-Young light* [49], a fast and memory-efficient significant pattern mining algorithm that models the statistical dependencies between patterns in the search space using the *Westfall-Young permutation testing procedure* [74].
- Permutation testing allows obtaining a more accurate approximation of the FWER than that provided by Tarone’s method, ultimately leading to a gain in statistical power.

Introduction

- The dependence between patterns in significant pattern mining arises due to several factors, ultimately owing to the combinatorial nature of the search space \mathcal{M} :
 - (i) Subset/superset relationships between patterns $\mathcal{S}, \mathcal{S}' \in \mathcal{M}$ are one obvious source of dependencies; if $\mathcal{S} \subseteq \mathcal{S}'$, the random variables $G_{\mathcal{S}}(X)$ and $G_{\mathcal{S}'}(X)$ indicating the occurrence of patterns \mathcal{S} and \mathcal{S}' in an input sample X must be positively correlated.
 - (ii) More generally, the random variables $G_{\mathcal{S}_1}(X)$ and $G_{\mathcal{S}_2}(X)$ corresponding to any two patterns $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{M}$ which share a common substructure, i.e. $\mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset$, will be statistically dependent as well.
- In order to obtain a tractable upper bound of the FWER, both the Bonferroni correction and Tarone’s method rely on the fact that $\Pr(\bigcup_{\mathcal{S} \in \mathcal{M}'} [p_{\mathcal{S}} \leq \delta]) \leq \sum_{\mathcal{S} \in \mathcal{M}'} \Pr(p_{\mathcal{S}} \leq \delta)$ always holds for any subset of patterns $\mathcal{M}' \subseteq \mathcal{M}$. Nevertheless, since a large number of patterns in \mathcal{M}' might be positively correlated with each other, $\sum_{\mathcal{S} \in \mathcal{M}'} \Pr(p_{\mathcal{S}} \leq \delta)$ is likely to greatly overestimate the true value of $\Pr(\bigcup_{\mathcal{S} \in \mathcal{M}'} [p_{\mathcal{S}} \leq \delta])$. Consequently, whenever the search space \mathcal{M} contains many interdependent patterns, both approaches tend to overestimate the FWER, thus often leading to a severe loss of statistical power.

Empirically Approximating the FWER Via Random Permutations

- Rather than replacing the intractable, exact FWER by a surrogate upper bound, an alternative strategy is to empirically estimate the exact value of $\text{FWER}(\delta)$ using resampling techniques.
- A popular approach towards this end is the *Westfall-Young permutation testing procedure*. Given an input dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, the procedure can be summarised as follows:
 - (i) Obtain j_p resampled datasets $\{\tilde{\mathcal{D}}^{(k)}\}_{k=1}^{j_p}$ by applying, for each $k = 1, \dots, j_p$, a different random permutation $\pi^{(k)} : \llbracket 1, n \rrbracket \rightarrow \llbracket 1, n \rrbracket$ to the class labels,

i.e. $\tilde{\mathcal{D}}^{(k)} = \left\{ (x_i, y_{\pi^{(k)}(i)}) \right\}_{i=1}^n$. By construction, the *global null hypothesis* $\mathcal{M}_0 = \mathcal{M}$ holds for all resampled datasets, that is, no pattern can be associated with the (randomly permuted) class labels.

- (ii) For each $k = 1, \dots, j_p$, compute $p_{\text{ms}}^{(k)} = \min \left\{ p_S^{(k)} \mid \mathcal{S} \in \mathcal{M} \right\}$, the most significant P-value across all patterns. Since any pattern deemed significantly associated in the resampled datasets must be a false positive, one or more false positives occur for the k -th resampled dataset at level δ if and only if $p_{\text{ms}}^{(k)} \leq \delta$.
- (iii) Consequently, the FWER at level δ can be estimated as:

$$\widehat{\text{FWER}}(\delta) = \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[p_{\text{ms}}^{(k)} \leq \delta \right].$$

- (iv) Finally, a corrected significance threshold based on the estimator $\widehat{\text{FWER}}(\delta)$ can be obtained as:

$$\delta_{\text{wy}} = \max \left\{ \delta \mid \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} \left[p_{\text{ms}}^{(k)} \leq \delta \right] \leq \alpha \right\}.$$

- Permutation testing implicitly accounts for the dependence between patterns, often leading to vastly more accurate estimates of the FWER than those obtained with Tarone's method.
- Nonetheless, a naive application of permutation testing to significant pattern mining is computationally unfeasible:
 - (i) Computing $p_{\text{ms}}^{(k)} = \min \left\{ p_S^{(k)} \mid \mathcal{S} \in \mathcal{M} \right\}$ via brute force for any resampled dataset is unrealistic, as it would require evaluating the P-value $p_S^{(k)}$ of every single candidate pattern in the search space \mathcal{M} .
 - (ii) Moreover, the already computationally challenging task of obtaining $p_{\text{ms}}^{(k)}$ must be repeated for a large number j_p of resampled datasets to obtain a sufficiently accurate estimate of the FWER.

Permutation Testing in Significant Pattern Mining

- Permutation testing-based significant pattern mining algorithms proceed in two phases, just as significant pattern mining approaches derived from LAMP do. Compared to the methods introduced in the previous chapter, only the first phase, which involves the computation of the corrected significance threshold, needs to change.
- The FastWY algorithm [43] constitutes the only former attempt to design a computationally efficient algorithm to obtain a permutation testing-based corrected significance threshold δ_{wy} .
 - FastWY exploits the concept of testability to compute $p_{\text{ms}}^{(k)}$ without the need to evaluate the P-value $p_S^{(k)}$ of all candidate patterns. In particular,

it leverages the observation that if $p_{\text{ms}}^{(k)}(\delta) = \min \{p_{\mathcal{S}}^{(k)} \mid \mathcal{S} \in \mathcal{M}_{\text{test}}(\delta)\}$ satisfies $p_{\text{ms}}^{(k)}(\delta) \leq \delta$, then $p_{\text{ms}}^{(k)} = p_{\text{ms}}^{(k)}(\delta)$. This motivates an algorithm which iteratively computes $p_{\text{ms}}^{(k)}(\delta)$ for increasing values of δ until the condition $p_{\text{ms}}^{(k)}(\delta) \leq \delta$ is satisfied.

- Despite the fact that FastWY can successfully obtain $p_{\text{ms}}^{(k)}$ while enumerating only a small subset of the search space \mathcal{M} , its computational efficiency is hindered by the fact that it must repeat the entire pattern enumeration process for each of the j_p resampled datasets. In practice, this limits its applicability to only datasets of small-to-moderate size.
- Our contribution, the Westfall-Young light algorithm, is a novel permutation testing-based significant pattern mining algorithm that computes δ_{wy} exactly while only enumerating patterns once.
 - Contrary to FastWY, the Westfall-Young light algorithm proceeds by initialising the estimate $\widehat{\delta}_{\text{wy}}$ of the corrected significance threshold to 1, iteratively decreasing $\widehat{\delta}_{\text{wy}}$ in each iteration. This search strategy, also used by the LAMP 2.0 algorithm described in Chapter 3, has been shown [65, 66] to be several orders of magnitude faster than that used by FastWY.
 - Westfall-Young light processes all j_p resampled datasets simultaneously, updating the estimates $\{\widetilde{p}_{\text{ms}}^{(k)}\}_{k=1}^{j_p}$ of the most significant P-value of each resampled dataset as it enumerates patterns.
 - Westfall-Young light exploits the fact that if a pattern \mathcal{S} is untestable at level $\widehat{\delta}_{\text{wy}}$, it cannot affect the permutation testing-based FWER estimator $\widehat{\text{FWER}}(\delta)$ for any $\delta < \widehat{\delta}_{\text{wy}}$. This allows bypassing the computation of P-values $p_{\mathcal{S}}^{(k)}$ for untestable patterns, as well as using the search space pruning condition introduced in Section 3.4.
 - Once the algorithm concludes its execution, the estimates $\{\widetilde{p}_{\text{ms}}^{(k)}\}_{k=1}^{j_p}$ of the most significant P-value of each resampled dataset are *not* equal to the true values $\{p_{\text{ms}}^{(k)}\}_{k=1}^{j_p}$. However, the fundamental property of the Westfall-Young light algorithm is that the estimates $\{\widetilde{p}_{\text{ms}}^{(k)}\}_{k=1}^{j_p}$ satisfy:

$$\widehat{\text{FWER}}(\delta) = \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} [p_{\text{ms}}^{(k)} \leq \delta] = \frac{1}{j_p} \sum_{k=1}^{j_p} \mathbb{1} [\widetilde{p}_{\text{ms}}^{(k)} \leq \delta]$$

for any $\delta \leq \widehat{\delta}_{\text{wy}}$.

- Since the algorithm guarantees that $\delta_{\text{wy}} \leq \widehat{\delta}_{\text{wy}}$ at any point during its execution, including after its termination, it follows that Westfall-Young light does away with the need to exactly evaluate $\{p_{\text{ms}}^{(k)}\}_{k=1}^{j_p}$ while still being able to exactly compute δ_{wy} . Compared to FastWY, this leads to a drastic reduction in the number of patterns that need to be enumerated to obtain δ_{wy} , considerably improving computational efficiency.

- The same low-level implementation considerations discussed in the context of the LAMP algorithm in Section 3.5 also apply to Westfall-Young light. Moreover, as the algorithm processes all resampled datasets simultaneously, a careful implementation of Westfall-Young light can precompute all P-values an enumerated pattern \mathcal{S} could attain, storing them in the form of a look-up table. This reduces the complexity of evaluating j_p P-values from $O(j_p)$ to $O(\min(n_1, r_{\mathcal{S}}))$.
- Westfall-Young light and FastWY are statistically indistinguishable. However, Westfall-Young light is drastically faster and more memory-efficient than FastWY.
- Westfall-Young light takes into account the dependence between candidate patterns in the search space, gaining additional statistical power compared to the LAMP 2.0 algorithm described in Chapter 3. Nonetheless, Westfall-Young light must evaluate j_p P-values for each testable pattern. Besides, since $\delta_{wy} \leq \delta_{tar}$, it must also enumerate more patterns than LAMP 2.0. Consequently, despite improving over FastWY, Westfall-Young light remains considerably more computationally-intensive than LAMP 2.0.

Experiments

- An exhaustive set of experiments comparing the runtime and memory usage of FastWY and Westfall-Young light in 12 significant itemset mining datasets and 12 significant subgraph mining datasets was presented.
- The experimental results showed that Westfall-Young light is two to three orders of magnitude faster than FastWY in significant itemset mining tasks and one to two orders of magnitude faster in significant subgraph mining tasks. The runtime gap seems to increase with the absolute runtime required to analyse the dataset, suggesting that the execution time of Westfall-Young light scales more gently with the size and density of the dataset.
- For small-to-moderate sized datasets, the memory usage is dominated by the underlying pattern mining algorithm, leading to Westfall-Young light and FastWY exhibiting the same memory footprint. However, for more demanding datasets, the memory requirements of FastWY increase sharply. This caused FastWY to be unable to complete the analysis of three significant itemset mining datasets, despite being executed on a server with 256 GB of memory. In contrast, the memory usage of Westfall-Young light scales gracefully with dataset size, being able to run until termination for all datasets under consideration.
- Given a minimum attainable P-value function $p_{\min}(r_{\mathcal{S}})$ and a significance threshold $\delta \in [0, 1]$, define the *minimum support* $r_{\min}(\delta)$ at level δ as the unique integer in the range $\llbracket 0, \min(n_1, n - n_1) \rrbracket$ that satisfies $p_{\min}(r_{\min}(\delta)) \leq \delta$ and $p_{\min}(r_{\min}(\delta) - 1) > \delta$. Then, the pruning condition of Section 3.4 evaluates to true for a pattern \mathcal{S} if and only if $r_{\mathcal{S}} < r_{\min}(\delta)$.
 - To compute the corrected significance threshold δ_{wy} , Westfall-Young light must enumerate all patterns that occur at least $r_{\min}(\delta_{wy})$ times in the dataset.

- However, as FastWY needs to compute $\left\{p_{\text{ms}}^{(k)}\right\}_{k=1}^{j_p}$ exactly, it must enumerate all patterns that occur at least $r_{\min}(\tilde{\delta}_{\text{fastwy}})$ times in the dataset, where $\tilde{\delta}_{\text{fastwy}}$ is defined as $\tilde{\delta}_{\text{fastwy}} = \max \left\{p_{\text{ms}}^{(k)} \mid k = 1, \dots, j_p\right\}$.
- Since $\tilde{\delta}_{\text{fastwy}} \leq \delta_{\text{wy}}$ and the minimum support is decreasing on δ , it follows that $r_{\min}(\delta_{\text{wy}}) \geq r_{\min}(\tilde{\delta}_{\text{fastwy}})$. Moreover, as real-world datasets are abundant with patterns that have a small *support* r_S , in practice FastWY must enumerate a much larger number of patterns than Westfall-Young light to compute δ_{wy} . This phenomenon was confirmed experimentally for both significant itemset mining and significant subgraph mining datasets.
- The performance of Westfall-Young light is roughly insensitive to the number j_p of random permutations used to estimate the FWER, provided that j_p is large enough (e.g. $j_p \geq 1,000$). When executed on a dataset for which the global null hypothesis holds, Westfall-Young light and FastWY obtain a resulting FWER close to the target FWER α . In contrast, the LAMP 2.0 algorithm yields a resulting FWER considerably smaller than α , a manifestation of the over-conservative nature of Tarone’s method. This indicates that Westfall-Young light and FastWY are superior to LAMP 2.0 in terms of statistical power, albeit less computationally efficient.

SUMMARY OF CHAPTER 5

- The need to incorporate into the model covariate factors that might have a confounding effect is an ubiquitous problem in computational biology and clinical data analysis.
- This chapter is devoted to introduce the Fast Automatic Conditional Search (FACS) algorithm [50], a novel significant pattern mining approach that can correct for a categorical covariate with an arbitrary number of categories.
- Compared to existing methods, FACS allows to drastically reduce spurious false positives due to confounding effects without sacrificing neither computational efficiency nor statistical power.

Introduction

- Let $G_S(X)$ be the binary random variable that indicates the occurrence of pattern S in an input sample X , Y the binary class label and C a random variable that represents a covariate factor that takes values in a domain \mathcal{C} . We say that the covariate C has a confounding effect on the statistical association between $G_S(X)$ and Y when:
 - (i) $G_S(X)$ and Y are *marginally* statistically associated, that is, $G_S(X) \not\perp Y$.
 - (ii) $G_S(X)$ and Y are *conditionally independent given C* , that is, $G_S(X) \perp Y \mid C$.
- *Confounded patterns*, that is, patterns for which conditions (i) and (ii) hold, provide no additional information about the class membership of an input sample beyond the information that is already contained in the covariate. Moreover, on many occasions, these patterns represent misleading associations, impairing our ability to acquire useful knowledge from the salient patterns in the data.
- All methods discussed in this thesis so far, such as LAMP 2.0 (Chapter 3) and Westfall-Young light (Chapter 4), aim at finding all patterns $S \in \mathcal{M}$ that are (marginally) statistically associated with the class labels, that is, they look for the set of patterns $\{S \in \mathcal{M} \mid G_S(X) \not\perp Y\}$.
- In contrast, an approach able to correct for the effect of a covariate C would aim to find the set of patterns $\{S \in \mathcal{M} \mid G_S(X) \not\perp Y \mid C\}$ instead.
- Our proposed approach, the FACS algorithm, can be understood as a generalisation of LAMP 2.0 which makes use of the Cochran-Mantel-Haenszel (CMH) test [46] to incorporate a categorical covariate C into the model. The fundamental methodological contributions leading to the FACS algorithm are:
 - (i) Proving that a minimum attainable P-value exists for the CMH test and deriving a tractable expression to evaluate it.
 - (ii) Developing a novel search space pruning criterion that can be used in combination with the CMH test.

Conditional Association Testing in Significant Pattern Mining

- The CMH test allows assessing the *conditional* association of two binary random variables $G_S(X)$ and Y given a categorical random variable C with k categories.

That is, the CMH test can be used to falsify the null hypothesis $H_0 : G_S(X) \perp\!\!\!\perp Y \mid C$.

- Unlike in previous chapters, we consider an input dataset $\mathcal{D} = \{(x_i, y_i, c_i)\}_{i=1}^n$ with each observation-label pair (x_i, y_i) being additionally tagged with a categorical covariate $c_i \in \{1, 2, \dots, k\}$, where k is the number of distinct categories that an outcome of the covariate C can belong to.
- At a high-level, the CMH test can be understood as the result of combining k distinct Pearson's χ^2 test statistics, each of them obtained on the subset $\mathcal{D}(c)$ of the original dataset \mathcal{D} which contains all samples for which the categorical covariate takes value c .
- As a consequence, the resulting P-value p_S can be expressed as a function $p_S(\mathbf{a}_S \mid \mathbf{n}, \mathbf{n}_1, \mathbf{r}_S)$ of:
 - (i) $\mathbf{a}_S = (a_{S,1}, a_{S,2}, \dots, a_{S,k})$, with $a_{S,c}$ being the number of samples in $\mathcal{D}(c)$ belonging to class $y = 1$ for which pattern S occurs.
 - (ii) $\mathbf{r}_S = (r_{S,1}, r_{S,2}, \dots, r_{S,k})$, with $r_{S,c}$ being the total number of samples in $\mathcal{D}(c)$ for which pattern S occurs, that is, the support of S in $\mathcal{D}(c)$.
 - (iii) $\mathbf{n}_1 = (n_{1,1}, n_{1,2}, \dots, n_{1,k})$, with $n_{1,c}$ being the number of samples in $\mathcal{D}(c)$ belonging to class $y = 1$.
 - (iv) $\mathbf{n} = (n_1, n_2, \dots, n_k)$, with n_c being the total number of samples in $\mathcal{D}(c)$.

The Minimum Attainable P-value for the CMH Test

- As Pearson's χ^2 test and Fisher's exact test, the CMH test is based on discrete data. Therefore, it can only attain a finite number of distinct values, implying the existence of a minimum attainable P-value $p_{S,\min}$ for any candidate pattern S .
- It can be shown (Proposition 5.1) that the minimum attainable P-value for the CMH test can be computed with only $O(k)$ operations as a multivariate function $p_{\min}(\mathbf{r}_S)$ of k variables.

A Search Space Pruning Condition for the CMH Test

- The existence of a search space pruning condition is essential for the computational feasibility of current significant pattern mining algorithms.
- In previous chapters it was shown that when the underlying test statistic is Pearson's χ^2 test or Fisher's exact test, the resulting minimum attainable P-value function $p_{\min}(r_S)$ is monotonically decreasing in $\llbracket 0, \min(n_1, n - n_1) \rrbracket$. Thus, descendants S' of an untestable pattern S with $r_S \leq \min(n_1, n - n_1)$ are untestable as well and can be pruned from the search space.
- Since the apriori property of pattern mining implies that $r_{S',c} \leq r_{S,c}$ if $S' \supseteq S$, an analogous pruning condition for the CMH test could be used if $r_{S',c} \leq r_{S,c} \leq \min(n_{1,c}, n_c - n_{1,c})$ for all $c = 1, 2, \dots, k$ implied that $p_{\min}(\mathbf{r}_{S'}) \geq p_{\min}(\mathbf{r}_S)$.
- Nevertheless, the former condition can be readily proven to be false. This has profound implications for the development of a valid pruning criterion for the

CMH test, as in principle there is no simple way to make a statement about the minimum attainable P-value $p_{S',\min}$ of a descendant S' of a pattern S based on $p_{S,\min}$ and \mathbf{r}_S alone.

- FACS uses a monotonically decreasing lower bound of the minimum attainable P-value as a surrogate of $p_{S,\min}$ in its pruning criterion. This surrogate, which will be referred to as the *lower envelope* of the minimum attainable P-value, is defined as

$$\tilde{p}_{S,\min} = \min_{S' \supseteq S} p_{S',\min}.$$

- An alternative characterisation of the lower envelope of the minimum attainable P-value can be obtained in terms of a combinational optimisation problem. Let $\mathcal{B}(\mathbf{r}_S) = \llbracket 0, r_{S,1} \rrbracket \times \llbracket 0, r_{S,2} \rrbracket \times \cdots \times \llbracket 0, r_{S,k} \rrbracket$ be the set of all $\mathbf{r}_{S'}$ satisfying $r_{S',c} \leq r_{S,c}$ for all $c = 1, 2, \dots, k$. Then, $\tilde{p}_{S,\min}$ is equivalently given by

$$\tilde{p}_{\min}(\mathbf{r}_S) = \min_{\mathbf{r}_{S'} \in \mathcal{B}(\mathbf{r}_S)} p_{\min}(\mathbf{r}_{S'}).$$

- By construction, the lower envelope of the minimum attainable P-value $\tilde{p}_{S,\min}$ satisfies (i) $\tilde{p}_{S,\min} \leq p_{S,\min}$ and (ii) $\tilde{p}_{S',\min} \geq \tilde{p}_{S,\min}$ for any $S' \supseteq S$. Consequently, if $\tilde{p}_{S,\min} > \hat{\delta}_{\text{tar}}$ and $r_{S,c} \leq \min(n_{1,c}, n_c - n_{1,c})$ for all $c = 1, 2, \dots, k$, all descendants of pattern S can be pruned from the search space (Proposition 5.3).
- As the pruning condition is evaluated for every pattern S which is enumerated during the execution of the algorithm, being able to efficiently compute $\tilde{p}_{S,\min}$ is of utmost importance. Attempting to obtain $\tilde{p}_{\min}(\mathbf{r}_S)$ by brute force, that is, by evaluating $p_{\min}(\mathbf{r}_{S'})$ for each $\mathbf{r}_{S'} \in \mathcal{B}(\mathbf{r}_S)$, is impractical, as it would require $O(n^k)$ operations each time the pruning condition needs to be assessed.
- A first improvement over this naive approach results from the fact that $p_{\min}(\mathbf{r}_{S'})$ must attain its minimum value at a vertex of $\mathcal{B}(\mathbf{r}_S)$ (Proposition 5.4). Consequently, the number of operations needed to compute $\tilde{p}_{\min}(\mathbf{r}_S)$ can be reduced from $O(n^k)$ to $O(2^k)$.
- Building on that result, a fundamental contribution of FACS is an algorithm to compute $p_{\min}(\mathbf{r}_{S'})$ exactly with only $O(k \log k)$ operations (Proposition 5.5). The resulting approach is instrumental in allowing to efficiently correct for a categorical covariate in significant pattern mining.

Miscellaneous Aspects of the FACS Algorithm

- Most low-level implementation considerations discussed in Section 3.5 in the context of LAMP 2.0 also apply to FACS. Nonetheless, some slight differences caused by the inclusion of covariates are:
 - (i) Unlike LAMP 2.0 or Westfall-Young `light`, FACS requires computing the support of each enumerated pattern in k disjoint subsets of the original input dataset \mathcal{D} , that is, it needs to compute $r_{S,c}$ for each $c = 1, 2, \dots, k$ instead of merely computing $r_S = \sum_{c=1}^k r_{S,c}$. Consequently, implementing the pattern enumeration routines for FACS often requires to slightly modify the original algorithm used to traverse the pattern enumeration tree.

- (ii) The number of distinct values that the minimum attainable P-value function for the CMH test can take grows as $O(n^k)$. As a result, two useful low-level optimisations for LAMP 2.0 and Westfall-Young light are no longer beneficial for FACS: (a) precomputing $p_{\min}(\mathbf{r}_S)$ as a look-up table becomes impractical and (b) grid search-based strategies to adjust the estimate $\hat{\delta}_{\text{tar}}$ of the corrected significance threshold each time it needs to be decreased become preferable to more complex alternatives that would require sorting all possible values of $p_{\min}(\mathbf{r}_S)$.
- By design, FACS allows to correct for a single categorical covariate. Nevertheless, this limitation can be alleviated as follows:
 - (i) Since FACS can handle a large number k of categories for the covariate, it is possible to correct for multiple categorical covariates C_1, C_2, \dots, C_d by defining a new covariate C with $k = \prod_{j=1}^d k_j$ categories.
 - (ii) Extending FACS to account for real-valued covariates is more challenging. However, our experiments suggest that a heuristic approach based on discretising continuous covariates can lead to satisfactory results.
- The FACS algorithm was designed to make use of Tarone’s method as the underlying approach to correct for the multiple comparisons problem. Nonetheless, all of the fundamental derivations behind FACS, including the minimum attainable P-value and pruning condition for the CMH test, can be directly applied to the Westfall-Young light algorithm introduced in Chapter 4. In particular, a version of Westfall-Young light able to account for a categorical covariate can be obtained by performing slight modifications to the original algorithm.

Experiments

- We carried out simulation experiments to evaluate our proposed approach, the FACS algorithm, in terms of computational efficiency, statistical power and ability to correct for confounding covariates. Additionally, a proof-of-concept application of FACS to analyse data from two genome-wide association studies in the plant model organism *A. thaliana* was also described.
- Using synthetic itemset mining data containing both a truly associated pattern and a confounded pattern, our simulation results showed that FACS can satisfactorily correct for confounding due to a categorical covariate. In contrast, as a result of its inability to account for the covariate, approximately half of all discoveries made by LAMP 2.0 in the same data are false positives owing to confounding. Moreover, our results strongly suggest that the possibility to correct for a categorical covariate does not lead to a loss of statistical power, as FACS and LAMP 2.0 display comparable performance in retrieving the truly associated pattern.
- Tarone’s method was found to be as effective in improving statistical power in conditional association testing as was known to be in marginal association testing; Bonf-CMH, a version of FACS which substitutes Tarone’s method by a naive Bonferroni correction, yields considerably less statistical power than FACS and LAMP 2.0.

- Tarone’s method was also found to be essential from a computational perspective. The runtimes of FACS and LAMP 2.0 grow at the same rate as the number of features p increases. In contrast, the runtime of Bonf-CMH increases at a much faster rate, becoming impractical even for moderate-sized datasets.
- Finally, our simulation experiments confirmed that being able to quickly assess the pruning condition for the CMH test is essential to obtain a computationally feasible approach. When using a brute force approach to evaluate the lower envelope of the minimum attainable P-value $\tilde{p}_{\min}(\mathbf{r}_S)$, the runtime grows so quickly with the number of categories for the covariate k that, with as few as $k = 10$ categories, even not applying any pruning results in a less inefficient algorithm. Most importantly, the runtime of FACS was shown to scale gently with k , empirically validating the derivations in Proposition 5.5. In summary, our experiments suggest that FACS provides the possibility to account for a categorical covariate, drastically reducing false discoveries due to confounding, at the expense of only a small computational overhead compared to LAMP 2.0.
- As a proof-of-principle, we tested the ability of FACS to correct for confounding due to population structure in two datasets derived from genome-wide association studies in plants. EIGENSTRAT [91], a popular method from statistical genetics, was used first to represent the genetic ancestry of each sample i as a d -dimensional, real-valued covariate $\mathbf{c}_i \in \mathbb{R}^d$. We followed a heuristic approach to discretise these covariates; the k -means algorithm [81] was applied to the set of d -dimensional vectors $\{\mathbf{c}_i\}_{i=1}^n$, resulting in a set of cluster assignments $\{\tilde{c}_i\}_{i=1}^n$ which was subsequently used as a categorical covariate for FACS. Our results showed that, despite the simplicity of this strategy, a categorical covariate obtained in this fashion allowed FACS to drastically reduce the *genomic inflation factor* [90], a popular measure of confounding in statistical genetics, compared to LAMP 2.0. Moreover, further analyses suggest that, at least in these two datasets, the resulting categorical covariate is as effective in correcting for the confounding effects of population structure as the original real-valued covariates provided by EIGENSTRAT.

SUMMARY OF CHAPTER 6

- In the past decade, *Genome-Wide Association Studies* (GWASs) have been extensively used to discover links between genomic markers, typically *single-nucleotide polymorphisms* (SNPs), and a wide variety of phenotypic traits.
- The overwhelming majority of GWASs look for statistically significant associations between individual SNPs and the phenotype being studied. As a result, if the sample size is small or the effect sizes of the associated markers are weak, the discovery of novel associations poses considerable difficulties.
- *Genetic heterogeneity*, the phenomenon that multiple genomic markers could affect a phenotype of interest in a similar manner, can be exploited to discover weakly associated markers that would remain undetected by univariate studies. For genomic regions for which the assumption of genetic heterogeneity holds, pooling several neighbouring markers into a joint *meta-marker* can result in a stronger and easier-to-detect association signal. This observation motivated the development of approaches to look for genomic regions (sets of contiguous markers) associated with the phenotype under study, rather than individual markers.
- Given a dataset with p markers, there are $\frac{p(p+1)}{2} = O(p^2)$ distinct genomic regions one could consider. Since typical GWAS datasets might comprise millions of SNPs, an exhaustive screening of *all* possible genomic regions would require performing hundreds of billions of association tests.
- The most popular methods used in statistical genetics are unable to cope with the statistical and computational challenges that would ensue. As a result, they require preselecting a small subset of candidate genomic regions *a priori*, for example, by dividing the genome into (possibly overlapping) fixed-length windows or defining regions to coincide with well-known functional units such as exons or entire genes.
- While this design choice drastically reduces the number of candidate regions in the search space, alleviating both the multiple comparisons burden and the computational complexity, statistical power will plummet if the set of candidate genomic regions to be tested was misspecified.
- In this chapter, we presented the Fast Automatic Interval Search (FAIS) [51] and FastCMH [52] algorithms, two approaches that make use of significant pattern mining to assess the statistical association of *all* genomic regions with a binary phenotype, regardless of their length or position.
- Unlike other existing methods, FAIS and FastCMH are insensitive to a potential misspecification of the set of regions to be tested. Empirically, this leads to a sharp gain in statistical power in situations for which reliable prior knowledge to select a subset of promising candidate genomic regions is unavailable.
- FAIS predates the development of the techniques to correct for a categorical covariate in significant pattern mining described in Chapter 5. Its resulting inability to correct for factors of variation that might have a confounding effect severely limits the range of GWAS datasets it can be applied to in practice. FastCMH does

away with this shortcoming by extending the original FAIS algorithm, leveraging the ideas of Chapter 5 to allow accounting for potential confounders such as age, gender or population structure.

Introduction

- In this chapter, each observation x will represent the genotype of an individual or specimen based on a set of p genomic markers measured at distinct positions (loci) in its genome. Accordingly, x will be defined as a sequence rather than a p -dimensional vector, with $x(j)$ denoting the value taken by the j -th genomic marker for $j = 1, 2, \dots, p$. Further assuming that the p genomic markers in x are SNPs, $x(j)$ can be considered to be a count variable that indicates the number of copies of the *minor allele* that the individual has at the j -th SNP. Thus, if the organism has q sets of chromosomes ($q = 2$ for humans), $x(j)$ will lie in $\llbracket 0, q \rrbracket$.
- Each observation x will be accompanied by a binary class label y , indicating the presence or absence of a phenotypic trait of interest, such as being a case for a certain disease under study. Optionally, observations might be additionally tagged by a categorical covariate c with an arbitrary number k of categories. In the context of GWAS, this covariate factor will be used to correct for potential confounders such as age, gender or population structure.
- We propose to carry out GWASs at a region level by using a new instance of significant pattern mining, which we refer to as *significant region mining*.
- In significant region mining, we identify each pattern \mathcal{S} in the search space \mathcal{M} with a different genomic region, i.e., $\mathcal{S} = \llbracket j_s, j_e \rrbracket$, where $1 \leq j_s \leq j_e \leq p$. The resulting search space $\mathcal{M} = \{\llbracket j_s, j_e \rrbracket \mid 1 \leq j_s \leq j_e \leq p\}$ thus contains $\frac{p(p+1)}{2}$ candidate regions.
- We say that a pattern \mathcal{S} occurs in an observation x if the total number of copies of a minor allele in the region, the so called *burden count* $b_{\mathcal{S}}(x) = \sum_{j \in \mathcal{S}} x(j)$, is non-zero. That is, the pattern occurrence indicator will be given by $g_{\mathcal{S}}(x) = \mathbb{1}[b_{\mathcal{S}}(x) > 0]$. Provided that the assumption of genetic heterogeneity holds for region \mathcal{S} and that all weakly associated markers in \mathcal{S} have homogeneous directions of effect, $g_{\mathcal{S}}(x)$ will exhibit a stronger association with the phenotype than any of the individual markers being pooled, facilitating the discovery of novel associations.
- In summary, the goal of significant region mining is to discover *all* genomic regions, regardless of their length or starting position, for which the presence or absence of one or more copies of a minor allele in the region is statistically associated (either marginally or given a categorical covariate) with a binary phenotype.
- Burden tests [48] are widely-used approaches to carry out GWASs at a region level. In particular, the Cohort Allelic Sums Test (CAST) [101] pools the markers in each genomic region in exactly the same manner as significant region mining, that is, as a binary variable that indicates if the genomic region contains minor alleles.

- However, burden tests and significant region mining differ drastically in the way they define their respective search spaces. Significant region mining aims to carry out an exhaustive scan of the genome by testing *all* genomic regions, making use of significant pattern mining to cope with the statistical and computational difficulties that this entails. In contrast, burden tests introduce *a priori* assumptions to limit the number of regions to be tested.
- One of the most widespread restrictions of the search space used by burden tests is to focus the analysis on regions of a certain length w , where w acts as a hyperparameter that must be chosen *a priori* on the basis of domain knowledge. Common examples include burden tests with non-overlapping windows, which have a search space $\mathcal{M} = \{\llbracket (i-1)w + 1, \min(iw, p) \rrbracket \mid 1 \leq i \leq \lceil \frac{p}{w} \rceil\}$, and burden tests with sliding windows of unit stride, which result in a search space $\mathcal{M} = \{\llbracket j_s, j_s + w - 1 \rrbracket \mid 1 \leq j_s \leq p - w + 1\}$.
- Burden tests and significant region mining can be expected to be complementary to some extent. If sufficiently reliable prior knowledge is available to narrow down the search space, burden tests will benefit from requiring a less stringent significance threshold, outperforming significant region mining algorithms in terms of statistical power and runtime. In contrast, if the set of regions to be tested is misspecified by, for example, setting the hyperparameter w to be too small or too large, the performance of burden tests could be severely compromised.
- Moreover, any attempts to choose w in a data-driven manner would require an additional correction for multiple comparisons or having access to a validation dataset to be used solely for hyperparameter selection. Significant region mining, which eliminates the need for such a hyperparameter, does away with this shortcoming, alleviating the potential for unintentional data dredging.

Method

- Among all other instances of significant pattern mining studied in this thesis, significant itemset mining is arguably the closest to significant region mining.
- In significant itemset mining, a pattern \mathcal{S} can be understood as a subset of features, that is, $\mathcal{S} \subseteq \{1, 2, \dots, p\}$. The same holds true for significant region mining. However, in the former paradigm *any* feature subset constitutes a valid pattern whereas in the latter only feature subsets that have consecutive indices correspond to genomic regions and thus are valid patterns. In other words, what sets apart both paradigms is the collection of patterns that are included in the search space \mathcal{M} .
- The search space in significant itemset mining is enormous, containing up to 2^p distinct patterns. Moreover, the number of patterns comprising $|\mathcal{S}|$ features is $\binom{p}{|\mathcal{S}|}$. Intuitively, this implies that the width of the pattern enumeration tree grows very rapidly as we descend levels in the tree.
- In contrast, while the search space in significant region mining is large when compared to other approaches in statistical genetics, such as burden tests, it

- “only” contains $\frac{p(p+1)}{2}$ different patterns, a much smaller number than in significant itemset mining. Besides, the number of patterns comprising $|\mathcal{S}|$ features is $p - |\mathcal{S}| + 1$. As a result, in significant region mining the pattern enumeration tree gets narrower, not wider, as we descend levels in the tree.
- From an algorithmic point of view, these differences justify a change in the strategy used to traverse the pattern enumeration tree.
 - In significant itemset mining breadth-first traversal is unfeasible: the pattern enumeration tree gets “too wide, too quickly” and the amount of memory necessary to follow that strategy would be unrealistic. As a result, all methods discussed in this thesis so far use depth-first traversal instead.
 - Nevertheless, enumerating patterns breadth-first has a clear advantage: it would allow to assess the search space pruning condition for all subsets \mathcal{S} of any given pattern \mathcal{S}' before enumerating \mathcal{S}' . In contrast, depth-first traversal would in general allow assessing it for only one of the subsets, resulting in a less effective pruning of the search space.
 - Due to the characteristics of the search space in significant region mining, breadth-first traversal is feasible. Consequently, we adopted this alternative strategy as part of both FAIS and FastCMH.
 - The pattern occurrence indicator $g_{\mathcal{S}}(x)$ also appears to differ drastically between significant itemset mining and significant region mining. In the former, $g_{\mathcal{S}}(x) = \prod_{j \in \mathcal{S}} u_j$, where an observation $x = (u_1, u_2, \dots, u_p)$ consists of p binary features. In the latter, $g_{\mathcal{S}}(x) = \mathbb{1} \left[\sum_{j \in \mathcal{S}} x(j) > 0 \right]$, where an observation x contains p integer-valued features in $\llbracket 0, q \rrbracket$.
 - Most importantly, the pattern occurrence indicator $g_{\mathcal{S}}(x)$ in significant region mining does *not* obey the Apriori property of pattern mining: in this new instance of significant pattern mining, $\mathcal{S}' \supset \mathcal{S}$ implies that $r_{\mathcal{S}'} \geq r_{\mathcal{S}}$ instead of $r_{\mathcal{S}'} \leq r_{\mathcal{S}}$.
 - Nevertheless, this apparently stark contrast is merely superficial. Two binary random variables $G_{\mathcal{S}}(X)$ and Y are statistically associated (either marginally or given a categorical covariate C) if and only if $1 - G_{\mathcal{S}}(X)$ and Y are statistically associated.
 - By looking for associations between $1 - G_{\mathcal{S}}(X)$ and Y rather than between $G_{\mathcal{S}}(X)$ and Y , the Apriori property is restored in significant region mining. Moreover, since $1 - g_{\mathcal{S}}(x) = \prod_{j \in \mathcal{S}} \mathbb{1} [x(j) = 0]$, it becomes evident that the pattern occurrence indicator in significant region mining is identical to that of a significant itemset mining dataset in which the original, integer-valued genomic markers $\{x(j)\}_{j=1}^p$ are represented by binary features $\{u_j\}_{j=1}^p$ defined as $u_j = 1$ if $x(j) = 0$ and $u_j = 0$ otherwise.
 - By viewing significant region mining as a restricted instance of significant itemset mining, we designed FAIS and FastCMH borrowing from methods already introduced in this thesis.
 - Our significant region mining algorithms proceed in three main steps:

- (i) Traverse the pattern enumeration tree once, in order to compute the corrected significance threshold δ .
 - (ii) Traverse the pattern enumeration tree a second time, aiming to retrieve the set $\mathcal{M}_{\text{sig,raw}}$ of genomic regions significantly associated with the phenotype at level δ .
 - (iii) Cluster the regions in $\mathcal{M}_{\text{sig,raw}}$, with the goal of obtaining a set $\mathcal{M}_{\text{sig,clustered}}$ of disjoint clusters of overlapping significantly associated genomic regions.
- The first two steps are entirely analogous to other significant pattern mining algorithms described in this thesis. More specifically, since FAIS and FastCMH make use of Tarone’s method to account for the multiple comparisons problem, FAIS parallels the LAMP 2.0 algorithm described in Chapter 3 and FastCMH the FACS algorithm proposed in Chapter 5. The only salient difference lies in the traversal of the pattern enumeration tree: FAIS and FastCMH enumerate patterns breadth-first, whereas LAMP 2.0 and FACS use depth-first traversal instead.
 - The last step aims at enhancing the interpretability of the results. As discussed in Chapter 4, related patterns tend to have correlated test statistics. In particular, this implies that if a certain genomic region \mathcal{S} is statistically associated with the phenotype, it is plausible that some other genomic regions which overlap with \mathcal{S} show a significant association too. As a result, the set $\mathcal{M}_{\text{sig,raw}}$ can often be summarised as a set $\mathcal{M}_{\text{sig,clustered}}$ of disjoint clusters of overlapping significantly associated genomic regions. For each cluster $\mathcal{C}_{\text{sig}} \in \mathcal{M}_{\text{sig,clustered}}$, we report both the individual genomic region $\mathcal{S}^* \in \mathcal{C}_{\text{sig}}$ with the smallest (most significant) P-value and the the union $\cup_{\mathcal{S} \in \mathcal{C}_{\text{sig}}} \mathcal{S} = \llbracket c_s, c_e \rrbracket$ of all regions in the cluster. Empirically, the former tends to slightly underestimate the length of the ground-truth associated region while the latter often slightly overestimates it.
 - While, for the sake of clarity, all algorithms described in this chapter have use Tarone’s method to obtain a corrected significance threshold, it is straightforward to extend FAIS and FastCMH to make use of permutation testing instead. This modification, which poses no algorithmic difficulties, would result in a similar trade-off as in other instances of significant pattern mining: statistical power would improve at the cost of additional runtime.
 - Another simple extension of FAIS and FastCMH concerns the definition of the pattern occurrence indicator. Rather than considering that just having a single minor allele in a genomic region might suffice to perturb its function, this assumption can be relaxed by introducing a burden threshold b_{min} . Under this alternative model of genetic heterogeneity, the pattern occurrence indicator would be given by $g_{\mathcal{S}}(x) = \mathbb{1}[b_{\mathcal{S}}(x) \geq b_{\text{min}}]$. No algorithmic modifications, other than the computation of $g_{\mathcal{S}}(x)$ itself, are necessary to implement this extension. However, choosing a value for the hyperparameter b_{min} is challenging and attempts to do so in a data-driven fashion would necessitate an additional correction for multiple comparisons.

Experiments

- We empirically evaluated the performance of our proposed significant region mining algorithms compared to several baseline approaches using synthetic

data, five *A. thaliana* datasets and one case/control study of Chronic Obstructive Pulmonary Disease (COPD) including two different ethnic groups (COPDGene).

- Using synthetic GWAS datasets generated in accordance with our model of genetic heterogeneity, we confirmed the ability of FAIS and FastCMH to gain statistical power by pooling weakly associated markers into a joint meta-marker. While this is not unexpected, due to the data being generated to fit our assumptions, the gap in statistical power between the significant region mining algorithms and their corresponding univariate baselines was dramatic. This illustrates that, under a model of genetic heterogeneity, it is perfectly possible to have genomic regions strongly associated with the phenotype while none of the markers it spans can be detected on their own.
- In spite of the fact that the search space in significant region mining is drastically smaller than in the other instances of significant pattern mining studied in this thesis, Tarone’s method was found to still be instrumental in making FAIS and FastCMH practical. When Tarone’s method is substituted by a naive Bonferroni correction, statistical power plummets and the runtime increases sharply. In particular, we empirically observed the runtime of FAIS and FastCMH to scale linearly with the number of markers p , being able to analyse datasets comprising millions of markers in less than one hour. In contrast, we estimated that the baselines based on a naive Bonferroni correction, which cannot prune the search space and thus scale quadratically with p , would have required more than a month’s worth of computations to complete the same analysis.
- Compared to burden tests, significant region mining algorithms eliminate the possibility to misspecify the set of candidate regions to be tested at the expense of necessitating a considerably more stringent corrected significance threshold. To investigate this trade-off, we generated synthetic GWAS datasets with seven truly associated genomic regions of distinct lengths and examined the statistical power of FastCMH and several burden test baselines. All burden tests were run with five different choices of the hyperparameter w determining the length of the regions to be tested. Our results showed that, despite requiring a stricter correction for multiple comparisons, significant region mining algorithms greatly outperform burden tests in this situation. In particular, we found that the probability of discovering a certain truly associated region using a burden test decreases sharply as the mismatch between w and the length l of the region increases. Consequently, no single choice for w allows the burden tests to consistently detect all seven regions. FastCMH, which is able to test regions of *all* lengths simultaneously, does not suffer from this limitation.
- To explore the performance of significant region mining algorithms under models other than genetic heterogeneity, we generated synthetic GWAS datasets in which a single, unobserved causal variant is associated with the phenotype. In these datasets, multiple observed neighbouring markers are correlated with the latent causal marker via linkage disequilibrium, thus being indirectly associated with the phenotype. Our results showed that, in this setting, significant region mining algorithms can lead to a gain in statistical power compared to traditional univariate approaches. This effect is maximised when the correlation between

the observed markers and the latent causal variant is modest, rendering the association signal present in the data barely detectable.

- The six real-world datasets considered in this thesis share two fundamental characteristics: (i) their class ratio is relatively balanced, with the proportion of cases not falling below 0.2 or exceeding 0.8, and (ii) they exhibit a substantial amount of confounding due to population structure, as measured by the genomic inflation factor [90]. However, they differ drastically in their sample size: while the COPDGene study comprises 7,993 participants, the five *A. thaliana* datasets have only between 84 and 95 samples.
- In order to account for population structure, we made use of the ability of FastCMH to correct for a categorical covariate. For each dataset, a categorical covariate representing population structure was defined using the same heuristic approach described in Chapter 5. First, the popular tool EIGENSTRAT [91] was used to obtain a low-dimensional embedding of the genotype of each sample, which was subsequently encoded as one of k categories using the k -means algorithm. The hyperparameter k was chosen to minimise the genomic inflation factor among a finite set of candidate values.
- Our results strongly suggest that, in this scenario, FastCMH is successful in correcting for the confounding effects of population structure. This observation holds true when assessing confounding using either the genomic inflation ratio or quantile-quantile (Q-Q) plots.
- FastCMH deems a much smaller number of regions to be significantly associated than FAIS. Most importantly, the decrease in the number of “hits” appears to be strongly correlated with the corresponding decrease in genomic inflation. This observation is consistent with the hypothesis that many of the genomic regions found by FAIS are only deemed significant due to the confounding effects of population structure.
- The results in all six real-world datasets further support the idea that significant region mining is a useful tool to complement univariate analyses in GWAS. In the COPDGene study, FastCMH discovered three significantly associated genomic regions, none of which contained any marker that would have been deemed significant by a univariate association test. Moreover, each of these regions overlaps with a gene that has been reported to be associated with COPD [103, 105], making the findings of FastCMH biologically plausible. The results in *A. thaliana* also confirm the potential of significant region mining to discover new associations: only 16 out of the 33 clusters of significantly associated regions retrieved by FastCMH contain markers that would have been discovered by a univariate association test.
- Most importantly, we found that the ability of significant region mining to carry out an exhaustive scan of the genome renders our proposed approaches complementary to burden tests as well. This effect could be seen particularly clearly in the analysis of the COPDGene dataset. None of the three significantly associated regions discovered by FastCMH were successfully retrieved by *any* of the burden tests we considered in our experiments. While all burden tests had candidate regions in their search space that fully contained the regions found by

FastCMH, the presence of additional, irrelevant markers contaminates the pooled meta-marker with noise. Once again, by testing *all* possible regions, FastCMH is resilient to this problem. Nonetheless, our results also suggest that, by requiring a less stringent corrected significance threshold, burden tests can find regions that would be missed by our approaches. This overall trend was confirmed by the results in the five *A. thaliana* datasets: the discoveries made by significant region mining and burden tests were often complementary.

- In summary, we believe that univariate association tests, burden tests and significant region mining should not be treated as mutually exclusive approaches but, rather, as tools that can be combined to analyse GWAS datasets at different levels of granularity.

AVAILABLE SOFTWARE

Table B.1 lists open source implementations for some of the most relevant significant pattern mining algorithms described in this thesis.

Table B.1. – A non-exhaustive list of existing software for significant pattern mining.

Algorithm	Reference	URL
LAMP, LAMP 2.0	[26, 65]	http://a-terada.github.io/lamp
LAMPLINK	[135]	http://a-terada.github.io/lamplink
MP-LAMP	[89]	http://github.com/tsudalab/mp-lamp
Westfall-Young light	[49]	http://significant-patterns.org
FACS	[50]	
FAIS, FastCMH	[51, 52]	

Several versions of LAMP, the first pattern mining approach based on Tarone’s method and arguably the most influential significant pattern mining algorithm to date, have publicly available implementations for practitioners to use. This includes both the original LAMP algorithm as presented in [26] as well as LAMP 2.0, the follow-up approach introduced by [65], which vastly improves the computational efficiency of LAMP in practice. Researchers who specifically aim to use significant pattern mining to detect high-order epistatic effects in genome-wide association studies might find LAMPLINK [135] particularly useful. This toolbox provides an implementation of LAMP 2.0 which is compatible with the input/output format and user interface of PLINK [136], a widely-used resource in the statistical genetics community. Finally, MP-LAMP [89], a parallel version of the LAMP algorithm, is also available as open source software. Empirically, MP-LAMP has been found [89] to provide a speed-up close to the number of processes used, rendering it a good option for analysing larger datasets.

Software tools for all contributions of this thesis have also been made available. In particular, this includes:

- (i) Westfall-Young light [49], our proposed approach to exploit the statistical dependence between patterns to improve statistical power (Chapter 4).
- (ii) FACS [50], our method to account for a categorical covariate in significant pattern mining to deal with observed confounding factors (Chapter 5).
- (iii) FAIS [51] and FastCMH [52], our novel algorithms to test all possible genomic regions for association with a phenotype of interest, regardless of their length or starting position (Chapter 6).

Providing additional implementations of these four algorithms compatible with the PLINK suite, in the same spirit as LAMPLINK, is currently pending as future work.

BIBLIOGRAPHY

1. FDA-NIH Biomarker Working Group. *BEST (Biomarkers, EndpointS, and other Tools) Resource [Online]* <https://www.ncbi.nlm.nih.gov/books/NBK326791/> (2017) (cit. on p. 3).
2. Bethesda (MD): National Library of Medicine (US) and National Center for Biotechnology Information. *GenBank [Online]* <https://www.ncbi.nlm.nih.gov/genbank/statistics/> (2017) (cit. on p. 3).
3. Moore, G. Cramming more components onto integrated circuits. *Electronics* **38**, 114–117 (1965) (cit. on p. 3).
4. Stephens, Z. D. *et al.* Big data: astronomical or genetical? *PLOS Biology* **13**, 1–11 (2015) (cit. on p. 3).
5. Ashley, E. A. Towards precision medicine. *Nature Reviews Genetics* **17**, 507–522 (2016) (cit. on p. 3).
6. 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**, 176–197 (2016) (cit. on p. 3).
7. Dugger, S. A., Platt, A. & Goldstein, D. B. Drug development in the era of precision medicine. *Nature Reviews Drug Discovery* (2017) (cit. on p. 3).
8. Poste, G. Bring on the biomarkers. *Nature* **469**, 156–157 (2011) (cit. on p. 3).
9. Waterton, J. C. & Pylkkanen, L. Qualification of imaging biomarkers for oncology drug development. *European Journal of Cancer* **48**, 409–415 (2012) (cit. on p. 3).
10. Listgarten, J. *et al.* Improved linear mixed models for genome-wide association studies. *Nature Methods* **9**, 525–526 (2012) (cit. on p. 3).
11. Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **58**, 267–288 (1996) (cit. on p. 3).
12. Zou, H. & Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**, 301–320 (2005) (cit. on p. 3).
13. Bondell, H. D. & Reich, B. J. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics* **64**, 115–123 (2008) (cit. on p. 3).
14. Li, C. & Li, H. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics* **24**, 1175–1182 (2008) (cit. on p. 3).
15. Azencott, C.-A., Grimm, D., Sugiyama, M., Kawahara, Y. & Borgwardt, K. Efficient network-guided multi-locus association mapping with graph cuts. *Bioinformatics* **29**, i171–i179 (2013) (cit. on p. 3).

Bibliography

16. Welter, D. *et al.* The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. *Nucleic Acids Research* **42**, D1001–D1006 (2014) (cit. on pp. 3, 99).
17. MacArthur, J. *et al.* The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog). *Nucleic Acids Research* **45**, D896–D901 (2017) (cit. on pp. 3, 99).
18. Visscher, P. M., Brown, M. A., McCarthy, M. I. & Yang, J. Five years of GWAS discovery. *The American Journal of Human Genetics* **90**, 7–24 (2012) (cit. on pp. 3, 7, 99).
19. Visscher, P. M. *et al.* 10 years of GWAS discovery: biology, function, and translation. *The American Journal of Human Genetics* **101**, 5–22 (2017) (cit. on pp. 3, 99).
20. Manolio, T. A. *et al.* Finding the missing heritability of complex diseases. *Nature* **461**, 747–753 (2009) (cit. on p. 3).
21. 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**, 1193–1198 (2012) (cit. on p. 3).
22. Hemani, G. *et al.* Detection and replication of epistasis influencing transcription in humans. *Nature* **508**, 249–253 (2014) (cit. on p. 3).
23. Forsberg, S. K., Bloom, J. S., Sadhu, M. J., Kruglyak, L. & Carlborg, Ö. Accounting for genetic interactions improves modeling of individual quantitative trait phenotypes in yeast. *Nature Genetics* **49**, 497–503 (2017) (cit. on p. 3).
24. Ravasi, T. *et al.* An atlas of combinatorial transcriptional regulation in mouse and man. *Cell* **140**, 744–752 (2010) (cit. on p. 4).
25. Teng, L., He, B., Gao, P., Gao, L. & Tan, K. Discover context-specific combinatorial transcription factor interactions by integrating diverse ChIP-Seq data sets. *Nucleic Acids Research* **42**, e24–e24 (2013) (cit. on p. 4).
26. Terada, A., Okada-Hatakeyama, M., Tsuda, K. & Sese, J. Statistical significance of combinatorial regulations. *Proceedings of the National Academy of Sciences* **110**, 12996–13001 (2013) (cit. on pp. 4–6, 23, 25, 53, 61, 170, 191).
27. Yeang, C.-H., McCormick, F. & Levine, A. Combinatorial patterns of somatic gene mutations in cancer. *The FASEB Journal* **22**, 2605–2622 (2008) (cit. on p. 4).
28. 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) (cit. on p. 4).
29. MacKay, D. J. *Information theory, inference and learning algorithms* (Cambridge University Press, 2003) (cit. on p. 4).
30. Agrawal, R., Imieliński, T. & Swami, A. Mining association rules between sets of items in large databases in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data* (1993), 207–216 (cit. on p. 4).
31. Zaki, M. J., Parthasarathy, S., Ogihara, M. & Li, W. New algorithms for fast discovery of association rules in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* (1997), 283–286 (cit. on pp. 4, 36, 85, 116).

32. Novak, P. K., Lavrač, N. & Webb, G. I. Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research* **10**, 377–403 (2009) (cit. on p. 4).
33. Zimmermann, A. & Nijssen, S. *Supervised pattern mining and applications to classification* in *Frequent Pattern Mining* (eds Aggarwal, C. C. & Han, J.) 425–442 (Springer, 2014) (cit. on pp. 4, 23).
34. Prinz, F., Schlange, T. & Asadullah, K. Believe it or not: how much can we rely on published data on potential drug targets? *Nature Reviews Drug Discovery* **10**, 712–712 (2011) (cit. on p. 4).
35. Begley, C. G. & Ellis, L. M. Drug development: raise standards for preclinical cancer research. *Nature* **483**, 531–533 (2012) (cit. on p. 4).
36. Open Science Collaboration. Estimating the reproducibility of psychological science. *Science* **349** (2015) (cit. on p. 4).
37. Head, M. L., Holman, L., Lanfear, R., Kahn, A. T. & Jennions, M. D. The extent and consequences of P-hacking in science. *PLOS Biology* **13**, e1002106 (2015) (cit. on p. 4).
38. Munafò, M. R. *et al.* A manifesto for reproducible science. *Nature Human Behaviour* **1** (2017) (cit. on p. 4).
39. Ioannidis, J. P. Acknowledging and overcoming nonreproducibility in basic and preclinical research. *JAMA* **317**, 1019–1020 (2017) (cit. on p. 4).
40. Webb, G. I. *Discovering significant rules* in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2006), 434–443 (cit. on pp. 5, 23).
41. Webb, G. I. Discovering significant patterns. *Machine Learning* **68**, 1–33 (2007) (cit. on pp. 5, 23).
42. Webb, G. I. Layered critical values: a powerful direct-adjustment approach to discovering significant patterns. *Machine Learning* **71**, 307–323 (2008) (cit. on pp. 5, 23).
43. Terada, A., Tsuda, K. & Sese, J. *Fast Westfall-Young permutation procedure for combinatorial regulation discovery* in *2013 IEEE International Conference on Bioinformatics and Biomedicine* (2013), 153–158 (cit. on pp. 6, 41, 48, 50, 55, 174).
44. 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**, 157–175 (1900) (cit. on pp. 6, 18, 168).
45. Fisher, R. A. On the interpretation of χ^2 from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society* **85**, 87–94 (1922) (cit. on pp. 6, 18, 168).
46. Mantel, N. & Haenszel, W. Statistical aspects of the analysis of data from retrospective studies of disease. *Journal of the National Cancer Institute* **22**, 719–748 (1959) (cit. on pp. 6, 65, 178).

Bibliography

47. Dickson, S. P., Wang, K., Krantz, I., Hakonarson, H. & Goldstein, D. B. Rare variants create synthetic genome-wide associations. *PLoS Biology* **8**, e1000294 (2010) (cit. on p. 7).
48. 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**, 5–23 (2014) (cit. on pp. 7, 99, 103, 184).
49. 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), 725–734 (cit. on pp. 8, 41, 173, 191).
50. 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 (cit. on pp. 8, 65, 75, 178, 191).
51. Llinares-López, F. *et al.* Genome-wide detection of intervals of genetic heterogeneity associated with complex traits. *Bioinformatics* **31**, i240–i249 (2015) (cit. on pp. 8, 100, 106, 116, 131, 183, 191).
52. Llinares-López, F. *et al.* Genome-wide genetic heterogeneity discovery with categorical covariates. *Bioinformatics* **33**, i1820–i1828 (2017) (cit. on pp. 8, 100, 116, 126, 130–133, 137, 183, 191).
53. Todeschini, R. & Consonni, V. *Handbook of molecular descriptors* (John Wiley & Sons, 2008) (cit. on p. 14).
54. Bullmore, E. & Sporns, O. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience* **10**, 186–198 (2009) (cit. on p. 14).
55. Van den Heuvel, M. P., Mandl, R. C., Stam, C. J., Kahn, R. S. & Pol, H. E. H. Aberrant frontal and temporal complex network structure in schizophrenia: a graph theoretical analysis. *Journal of Neuroscience* **30**, 15915–15926 (2010) (cit. on p. 14).
56. Bertsekas, D. P. & Tsitsiklis, J. N. *Introduction to probability* (Athena Scientific, 2002) (cit. on p. 16).
57. Shaffer, J. P. Multiple hypothesis testing. *Annual review of psychology* **46**, 561–584 (1995) (cit. on p. 21).
58. Dmitrienko, A., Tamhane, A. C. & Bretz, F. *Multiple testing problems in pharmaceutical statistics* (Chapman and Hall/CRC, 2009) (cit. on p. 21).
59. Noble, W. S. How does multiple testing correction work? *Nature Biotechnology* **27**, 1135–1137 (2009) (cit. on p. 21).
60. Bonferroni, C. E. *Teoria statistica delle classi e calcolo delle probabilita* (Libreria Internazionale Seeber, 1936) (cit. on pp. 22, 168).
61. Dunn, O. J. Estimation of the medians for dependent variables. *The Annals of Mathematical Statistics* **30**, 192–197 (1959) (cit. on pp. 22, 168).

62. Bay, S. D. & Pazzani, M. J. Detecting group differences: mining contrast sets. *Data Mining and Knowledge Discovery* **5**, 213–246 (2001) (cit. on p. 23).
63. Hämäläinen, W. StatApriori: an efficient algorithm for searching statistically significant association rules. *Knowledge and Information Systems* **23**, 373–399 (2010) (cit. on p. 23).
64. Tarone, R. E. A modified Bonferroni method for discrete data. *Biometrics* **46**, 515–522 (1990) (cit. on pp. 23, 168).
65. 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), 422–436 (cit. on pp. 25, 53, 161, 170, 175, 191).
66. Sugiyama, M., Llinares-López, F., Kasenburg, N. & Borgwardt, K. Significant subgraph mining with multiple testing correction in *Proceedings of the 2015 SIAM International Conference on Data Mining* (2015), 37–45 (cit. on pp. 25, 53, 161, 170, 175).
67. Aggarwal, C. C. & Han, J. *Frequent pattern mining* (Springer, 2014) (cit. on p. 26).
68. Han, J., Pei, J. & Yin, Y. Mining frequent patterns without candidate generation in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (2000), 1–12 (cit. on p. 36).
69. Uno, T., Kiyomi, M. & Arimura, H. LCM ver. 2: efficient mining algorithms for frequent/closed/maximal itemsets in *Proceedings of IEEE ICDM'04 Workshop FIMI'04* (2004) (cit. on pp. 36, 54).
70. Borgelt, C. Frequent item set mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2**, 437–456 (2012) (cit. on pp. 36, 85).
71. Yan, X. & Han, J. gSpan: graph-based substructure pattern mining in *2002 IEEE International Conference on Data Mining* (2002), 721–724 (cit. on p. 36).
72. Nijssen, S. & Kok, J. N. The Gaston tool for frequent subgraph mining. *Electronic Notes in Theoretical Computer Science* **127**, 77–87 (2005) (cit. on pp. 36, 54).
73. Jiang, C., Coenen, F. & Zito, M. A survey of frequent subgraph mining algorithms. *The Knowledge Engineering Review* **28**, 75–105 (2013) (cit. on p. 36).
74. Westfall, P. H. & Young, S. S. *Resampling-based multiple testing: Examples and methods for p-value adjustment* (John Wiley & Sons, 1993) (cit. on pp. 45, 47, 173).
75. Goethals, B. & Zaki, M. J. Frequent itemset mining dataset repository (FIMI'04) [Online] <http://fimi.ua.ac.be/data> (2017) (cit. on pp. 54–55).
76. Wörlein, M., Meinel, T., Fischer, I. & Philippsen, M. A quantitative comparison of the subgraph miners MoFa, gSpan, FFSM, and Gaston in *Knowledge Discovery in Databases: PKDD 2005, 9th European Conference on Principles and Practice of Knowledge Discovery in Databases* (2005), 392–403 (cit. on p. 55).
77. Lichman, M. *UCI Machine Learning Repository* [Online] <https://archive.ics.uci.edu/ml/index.php> (2017) (cit. on p. 55).

Bibliography

78. Kong, X. & Yu, P. S. *Semi-supervised feature selection for graph classification* in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2010), 793–802 (cit. on p. 56).
79. Clauset, A., Rohilla Shalizi, C. & Newman, M. E. J. Power-law distributions in empirical data. *Society for Industrial and Applied Mathematics Review* **51**, 661–703 (2009) (cit. on p. 60).
80. Vilhjálmsson, B. J. & Nordborg, M. The nature of confounding in genome-wide association studies. *Nature Reviews Genetics* **14**, 1–2 (2013) (cit. on p. 66).
81. Lloyd, S. Least squares quantization in PCM. *IEEE Transactions on Information Theory* **28**, 129–137 (1982) (cit. on pp. 84, 182).
82. Terada, A., Tsuda, K., *et al.* *Significant pattern mining with confounding variables* in *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (2016), 277–289 (cit. on p. 85).
83. Mehta, C. R. & Patel, N. R. Exact logistic regression: theory and examples. *Statistics in medicine* **14**, 2143–2160 (1995) (cit. on p. 85).
84. 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), 326–335 (cit. on p. 85).
85. Leisch, F., Weingessel, A. & Hornik, K. On the generation of correlated artificial binary data. *Working Paper Series, SFB « Adaptive Information Systems and Modelling in Economics and Management Science », Vienna University of Economics* (1998) (cit. on pp. 87, 118, 128).
86. Cover, T. M. & Thomas, J. A. *Elements of information theory* (John Wiley & Sons, 2012) (cit. on p. 87).
87. Atwell, S. *et al.* Genome-wide association study of 107 phenotypes in a common set of Arabidopsis thaliana inbred lines. *Nature* **465**, 627 (2010) (cit. on pp. 93, 130–131).
88. Grimm, D. G. *et al.* easyGWAS: A cloud-based platform for comparing the results of genome-wide association studies. *The Plant Cell* (2016) (cit. on pp. 94, 130).
89. Yoshizoe, K., Terada, A. & Tsuda, K. MP-LAMP: parallel detection of statistically significant multi-loci markers on cloud platforms. *Bioinformatics*, bty219 (2018) (cit. on pp. 94, 161, 191).
90. Devlin, B. & Roeder, K. Genomic control for association studies. *Biometrics* **55**, 997–1004 (1999) (cit. on pp. 95, 131, 182, 189).
91. Price, A. L. *et al.* Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics* **38**, 904–909 (2006) (cit. on pp. 96, 131, 182, 189).
92. Wilks, S. S. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics* **9**, 60–62 (1938) (cit. on p. 97).

93. 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) (cit. on p. 99).
94. Nolte, I. M. *et al.* Genetic loci associated with heart rate variability and their effects on cardiac disease risk. *Nature Communications* **8** (2017) (cit. on p. 99).
95. Li, Z. *et al.* Genome-wide association analysis identifies 30 new susceptibility loci for schizophrenia. *Nature Genetics* **49**, 1576 (2017) (cit. on p. 99).
96. Greene, C. S. *et al.* Understanding multicellular function and disease with human tissue-specific networks. *Nature Genetics* **47**, 569–576 (2015) (cit. on p. 99).
97. Wood, A. R. *et al.* Defining the role of common variation in the genomic and biological architecture of adult human height. *Nature Genetics* **46**, 1173–1186 (2014) (cit. on p. 99).
98. Sniekers, S. *et al.* Genome-wide association meta-analysis of 78,308 individuals identifies new loci and genes influencing human intelligence. *Nature Genetics* **49** (2017) (cit. on p. 99).
99. Burrell, R. A., McGranahan, N., Bartek, J. & Swanton, C. The causes and consequences of genetic heterogeneity in cancer evolution. *Nature* **501**, 338–346 (2013) (cit. on p. 99).
100. Listgarten, J. *et al.* A powerful and efficient set test for genetic markers that handles confounders. *Bioinformatics* **29**, 1526–1533 (2013) (cit. on p. 99).
101. Morgenthaler, S. & Thilly, W. G. A strategy to discover genes that carry multi-allelic or mono-allelic risk for common diseases: a cohort allelic sums test (CAST). *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis* **615**, 28–56 (2007) (cit. on pp. 103, 184).
102. Regan, E. A. *et al.* Genetic epidemiology of COPD (COPDGene) study design. *COPD: Journal of Chronic Obstructive Pulmonary Disease* **7**, 32–43 (2011) (cit. on p. 130).
103. Cho, M. H. *et al.* Risk loci for chronic obstructive pulmonary disease: a genome-wide association study and meta-analysis. *The Lancet Respiratory Medicine* **2**, 214–225 (2014) (cit. on pp. 130, 137, 189).
104. Marchini, J., Cardon, L. R., Phillips, M. S. & Donnelly, P. The effects of human population structure on large genetic association studies. *Nature Genetics* **36**, 512–517 (2004) (cit. on p. 131).
105. Cho, M. H. *et al.* Variants in FAM13A are associated with chronic obstructive pulmonary disease. *Nature Genetics* **42**, 200–202 (2010) (cit. on pp. 137, 189).
106. 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 (Statistical Methodology)* **57**, 289–300 (1995) (cit. on p. 150).
107. Benjamini, Y. & Yekutieli, D. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics* **29**, 1165–1188 (2001) (cit. on p. 151).

Bibliography

108. Yekutieli, D. & Benjamini, Y. Resampling-based false discovery rate controlling multiple test procedures for correlated test statistics. *Journal of Statistical Planning and Inference* **82**, 171–196 (1999) (cit. on p. 151).
109. Storey, J. D. & Tibshirani, R. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences* **100**, 9440–9445 (2003) (cit. on p. 151).
110. Romano, J. P., Shaikh, A. M. & Wolf, M. Control of the false discovery rate under dependence using the bootstrap and subsampling. *TEST* **17**, 417 (2008) (cit. on p. 151).
111. 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**, 143–158 (2005) (cit. on p. 151).
112. 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), 897–906 (cit. on p. 151).
113. Taylor, J. & Tibshirani, R. J. Statistical learning and selective inference. *Proceedings of the National Academy of Sciences* **112**, 7629–7634 (2015) (cit. on p. 152).
114. Lee, J. D., Sun, D. L., Sun, Y. & Taylor, J. E. Exact post-selection inference, with application to the lasso. *The Annals of Statistics* **44**, 907–927 (2016) (cit. on pp. 152–153).
115. Suzumura, S., Nakagawa, K., Umezu, Y., Tsuda, K. & Takeuchi, I. *Selective inference for sparse high-order interaction models* in *International Conference on Machine Learning* (2017), 3338–3347 (cit. on pp. 153, 156).
116. Tian, X., Loftus, J. R. & Taylor, J. E. Selective inference with unknown variance via the square-root LASSO. *arXiv preprint 1504.08031* (2015) (cit. on p. 153).
117. Loftus, J. R. Selective inference after cross-validation. *arXiv preprint 1511.08866* (2015) (cit. on p. 153).
118. Harris, X. T., Panigrahi, S., Markovic, J., Bi, N. & Taylor, J. Selective sampling after solving a convex problem. *arXiv preprint 1609.05609* (2016) (cit. on p. 153).
119. Taylor, J. & Tibshirani, R. Post-selection inference for L₁-penalized likelihood models. *Canadian Journal of Statistics* (2017) (cit. on p. 153).
120. Markovic, J., Xia, L. & Taylor, J. E. Comparison of prediction errors: adaptive P-values after cross-validation. *arXiv preprint 1703.06559* (2017) (cit. on p. 153).
121. Wolf, E., Llinares-López, F., Borgwardt, K. & Hofmann, T. *Accounting for heterogeneous effect signs in significant pattern mining* Bachelor Thesis (ETH Zürich, 2017) (cit. on p. 154).
122. Pati, Y. C., Rezaifar, R. & Krishnaprasad, P. S. *Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition* in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers* **1** (1993), 40–44 (cit. on p. 156).

123. Tatti, N. *Itemsets for real-valued datasets* in *2013 IEEE 13th International Conference on Data Mining* (2013), 717–726 (cit. on pp. 157–159).
124. Sugiyama, M. & Borgwardt, K. M. Significant pattern mining on continuous variables. *arXiv preprint 1702.08694* (2017) (cit. on pp. 158–159).
125. Woolf, B. The log likelihood ratio test (the G-test). *Annals of Human Genetics* **21**, 397–409 (1957) (cit. on p. 159).
126. Saraswat, V. A., Kambadur, P., Kodali, S., Grove, D. & Krishnamoorthy, S. *Lifeline-based global load balancing* in *Proceedings of the 16th ACM Symposium on Principles and Practice of Parallel Programming* (2011), 201–212 (cit. on p. 161).
127. Terada, A., Kim, H. & Sese, J. *High-speed Westfall-Young permutation procedure for genome-wide association studies* in *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics* (2015), 17–26 (cit. on p. 161).
128. Pasaniuc, B. *et al.* Fast and accurate imputation of summary statistics enhances evidence of functional enrichment. *Bioinformatics* **30**, 2906–2914 (2014) (cit. on p. 161).
129. Hommel, G. & Krummenauer, F. Improvements and modifications of Tarone’s multiple test procedure for discrete data. *Biometrics* **54**, 673–681 (1998) (cit. on p. 162).
130. Roth, A. J. Multiple comparison procedures for discrete test statistics. *Journal of Statistical Planning and Inference* **82**, 101–117 (1999) (cit. on p. 162).
131. Pierson, E., GTEx Consortium, Koller, D., Battle, A. & Mostafavi, S. Sharing and specificity of co-expression networks across 35 human tissues. *PLOS Computational Biology* **11**, 1–19 (2015) (cit. on p. 163).
132. Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y. & Morishima, K. KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research* **45**, D353–D361 (2017) (cit. on p. 163).
133. Li, T. *et al.* A scored human protein-protein interaction network to catalyze genomic interpretation. *Nature Methods* **14**, 61–64 (2017) (cit. on p. 163).
134. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015) (cit. on p. 163).
135. Terada, A., Yamada, R., Tsuda, K. & Sese, J. LAMPLINK: detection of statistically significant SNP combinations from GWAS data. *Bioinformatics* **32**, 3513–3515 (2016) (cit. on p. 191).
136. Purcell, S. *et al.* PLINK: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics* **81**, 559–575 (2007) (cit. on p. 191).

