Automated Machine Learning on high-dimensional biological data

Zacharias Papadovasilakis

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Bioinformatics

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Supervisor: Prof. *Ioannis Tsamardinos*

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 617393

University of Crete
School of Medicine

Automated Machine Learning on high-dimensional biological data

Thesis submitted by
Zacharias Papadovasilakis
in partial fulfillment of the requirements for the
Masters' of Science degree in Bioinformatics

THESIS APPROVAL

Author: _____
Zacharias Papadovasilakis

Committee approvals: _____
Ioannis Tsamardinos
Professor, Thesis Supervisor

_____
Panayiota Poirazi
Research Director, Committee Member

_____
Georgios Potamias
Principal Researcher, Committee Member

Heraklion, December 2018

Abstract

Since their first application, Genome Wide Association Studies have evolved significantly and provided useful insight on medical diagnostics. Their main aim is to establish a connection between a variety of traits such as, human diseases or protein concentration levels, and the genetic background (usually via point mutations) of a given species. Problems of this type suffer from several issues primarily caused by high dimensionality (millions of Single Nucleotide Polymorhisms), low sample size, need of multiple testing correction and taking into account population structure.

In this thesis, we address current GWAS methodological issues utilizing a feature selection method, termed *generalized Orthogonal Matching Pursuit* (**gOMP**). *gOMP* offers a variety of advantageous characteristics such as **a)** computational efficiency and scalability to number of features, **b)** adaptability to any type of outcome variable (e.g. binary, continuous, time-to-event etc) and **c)** simplicity in terms of implementation. *gOMP* can also be fully integrated into **JAD Bio's™** automated machine learning pipeline which ensures methodological correctness in terms of proper model-building procedure and unbiased predictive performance estimation. On top of that, we extend *gOMP's* functionality by **a)** parallelizing its operation feature-wise and **b)** identifying features that are statistically equivalent to the already selected ones. Regarding equivalent features, we argue that the produced multiple solutions are able to capture and correct the underlying population structure. In order to evaluate *gOMP's* performance, we extensively compare it with *QTCAT* over a series of simulated datasets. Additionally, we apply *gOMP* to real human-disease datasets. As a result, *gOMP* proves to be a highly efficient method for genomic datasets in terms of performance, retrieval of associated features and computational cost.

Περίληψη

Από την πρώτη τους εφαρμογή, οι Genome Wide Association (GWA) μελέτες έχουν παρουσιάσει σημαντική εξέλιξη και έχουν προσφέρει πολύτιμη βοήθεια στη διαγνωστική ιατρική. Κύριος στόχος τους αποτελεί η δημιουργία σύνδεσης ανάμεσα σε ένα σύνολο χαρακτηριστικών όπως ανθρώπινων ασθενειών, ή επίπεδο πρωτεϊνικών συγκεντρώσεων, και στο γενετικό υπόβαθρο (συνήθως μέσω σημειακών μεταλλάξεων) ενός συγκεκριμένου βιολογικού είδους. Ερωτήματα τέτοιας μορφής είναι συχνά επιρρεπή σε προβλήματα που προκύπτουν κυρίως από τον υψηλό αριθμό διαστάσεων (εκατομμύρια καταγεγγραμμένες σημειακές μεταλλάξεις), τον χαμηλό αριθμό δειγμάτων, την ανάγκη για διόρθωση στον έλεγχο πολλαπλών υποθέσεων καθώς και την ανάγκη να ληφθεί υπόψιν η πληθυσμιακή δομή των δειγμάτων.

Στη συγκεκριμένη διπλωματική εργασία, αντιμετωπίζουμε τα τρέχοντα μεθοδολογικά προβλήματα των *GWA* αναλύσεων χρησιμοποιώντας μία μέθοδο επιλογής μεταβλητών, ονομαζόμενη *generalized Orthogonal Matching Pursuit*-**gOMP**. Ο *gOMP* προσφέρει πληθώρα ευνοϊκών χαρακτηριστικών όπως **α)** υπολογιστική ταχύτητα και επεκτασιμότητα σε οποιοδήποτε αριθμό μεταβλητών, **β)** προσαρμοστικότητα σε οποιοδήποτε τύπο εξαρτημένης μεταβλητής (π.χ. δυαδική, συνεχής, time-to-event κ.α.) και **γ)** απλότητα ως προς την υπολοίησή του. Επίσης, ο *gOMP* είναι σε θέση να ενσωματωθεί πλήρως με το αυτοματοποιημένο σύστημα μηχανικής μάθησης **JAD Bio's**™ το οποίο εξασφαλίζει μεθοδολογική ορθότητα σχετικά με τη διαδικασία δημιουργίας των στατιστικών μοντέλων, καθώς και την αμερόληπτη εκτίμηση της προβλεπτικής επίδοσης. Επιπροσθέτως, επεκτείνουμε τα τεχνικά χαρακτηριστικά του *gOMP* μέσω παραλληλοποίησης της λειτουργίας του ως προς τον αριθμό των μεταβλητών, καθώς και μέσω της προσθήκης της δυνατότητας εύρεσης πολλαπλών μεταβλητών, στατιστικά ισοδύναμων των ήδη επιλεγμένων. Σχετικά με τις ισοδύναμες υπογραφές, υποστηρίζουμε ότι μέσω αυτών είναι δυνατή η αποτύπωση και η διόρθωση των φαινομένων που πηγάζουν από την πληθυσμιακή δομή. Ως προς την αξιολόγηση της επίδοσης του, επιχειρείται μία εκτενής συγκρισή ανάμεσα στο *gOMP* και στο *QTCAT* πάνω σε προσομοιωμένα δεδομένα. Στη συνέχεια, ο *gOMP* εφαρμόζεται και σε πραγματικά δεδομένα που αφορούν σε ανθρώπινες ασθένειες. Ως αποτέλεσμα, ο *gOMP* αποδεικνύεται μία ισχυρή μέθοδος ανάλυσης γενομικών δεδομένων όσον αφορά την επίδοση, την εύρεση των συσχετισμένων με το φαινότυπο μεταβλητών καθώς και ως προς την υπολογιστική πολυπλοκότητα (χρόνου εκτέλεσης).

Acknowledgements

First and foremost I would like to thank my supervisor prof. Ioannis Tsamardinos for his support throughout my program, as well as for giving me the opportunity to be engaged in a such interesting scientific field.

I would also like to thank Dr. Klio Lakiotaki, Dr. Michail Tsagris and Paul Charonyktakis for their invaluable cooperation, comments and especially long discussions.

Mostly, I would like to thank my family and friends for their unconditional love and support throughout all these years.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Throughout the history of mankind, predicting events and phenomena has been intertwined with societies' welfare and fueling of technological advancements. A valid prediction springs from meticulous observations, or data gathering, and an application of a systematic methodology for inferring from these observations.

Hardware improvements regarding processing power and memory efficiency, along with the development of robust machine learning and applied statistics methods, have been the cornerstone upon which the emerging scientific field of *Data Analysis* has been built.

Biology and socioeconomics are representative fields where each phenomenon is associated by an enormous, feature-wise, dimensionality, with a great degree of complexity between feature interactions, rendering them ideal candidates for machine learning applications.

1.1 Motivation

A common task in supervised machine learning that has been studied for decades is *Feature Selection* (FS), also known as variable or attribute selection. FS can be defined as the identification of a minimal-sized subset of features that maximally predict an outcome, or target, variable of interest (Pantazis et al., 2017; Borboudakis and Tsamardinos, 2017), in other words, removing irrelevant features that are not associated with the target variable.

In many scientific fields, especially in bioinformatics, which incorporates machine learning techniques, FS proves to be significantly helpful in numerous ways; It can reduce the computing, storing and/or measuring cost of variables, e.g. in medical diagnostics. Parsimonious predictive models are easier to interpret, conceptualize and inspect. In many cases, it leads to more accurate models by removing the inherent noise in high-dimensional problems, restricting the so called *curse of dimensionality*. More importantly, FS is primarily employed for knowledge discovery by retaining the features able to describe the data generation mechanisms; in causality, FS is often the first step in identifying causal relations among features.

(Tsamardinos et al., 2003). In such domains, it is often the case that multiple, equivalent solutions to the feature selection problem do exist due to natural fail-safe mechanisms, thus identification of all the possible feature subsets can provide a much clearer picture of the problem at hand.

Biological and medical data are mostly comprised by a small sample size and a considerably large feature size, often called high-dimensional, or "small n - large p" problems. This characteristic creates 2 main difficulties in analysis:

1. Need for robust statistics, insusceptible to low sample size in terms of inference ability.

2. Algorithmic techniques that are able to scale up to a such high feature size.

On top of these, any reported predictions or models produced must be accurate, consistent and as much parsimonious as possible, since controlling *Type I* errors in biology is gravely important. To date, the amount of different analysis pipelines is too large, in terms of combinations of data preprocessing, algorithms used for modelling and tuning of their respective hyper-parameters. The above issues can easily be tackled by an objective systematic methodology, basically an automated machine learning pipeline (a main component of this thesis), that ensures correctness and prevents overfitting.

1.2 Contribution

In this thesis, we utilize a modified, generalized orthogonal matching pursuit (**gOMP**) algorithm for feature selection on genomic variant (SNP) data, integrated with the **JAD Bio™** automated machine learning pipeline that includes all the necessary analysis steps, i.e. data preprocessing, tuning of feature selection hyper-parameters, selection of modelling algorithms and performance bias correction.

gOMP accepts numerous types of target variables, e.g. continuous, binary, time-to-event to name a few, and scales up to thousands, or millions variables [1]. We also extend gOMP to identify multiple solutions to the feature selection problem. By doing so, we achieve to produce signatures of equal performance and provide, simultaneously, a population structure correction method.

The above work synthesizes an automated and complete tool for *GWA* analyses, able to systematically identify additive effects in high-dimensional data and detect bibliographically known associations, as well as variants previously unknown to biology specialists.

1.3 Outline

The rest of the thesis is organized as follows:

[1] A typical genome-wide dataset contains millions of polymorphisms.

Chapter 2 surveys existing work on feature selection with population structure correction methods, when dealing with such high-dimensional problems.

In Chapter 3 we describe analytically the methods used for analyzing genomic datasets, under two major test categories: with phenotype-simulated data and with real, human-disease data acquired from *European Genome and phenome Archive* (EGA). Regarding tests with simulated data, we empirically evaluate gOMP by comparing it to a state-of-the-art algorithm, QTCAT, on exactly the same grounds.

In Chapter 4 we present the results produced from simulation and real-dataset studies, in order to assess our method's performance end evaluate newly identified loci.

In Chapter 5 we comment on the results, explore the advantages and disadvantages of both methods and contemplate on future work and directions.

Chapter 2

Literature Review

As a result to the recent improvements, in terms of decreasing monetary cost and processing time, in high-throughput genotyping methods, the availability of polymorphism data has increased dramatically, calling for suitable statistical-analysis methods. During those years, *Genome-Wide-Association-Studies* (GWAS) have attracted substantial research interest and become the default analysis method for such data. GWAS belong to the category of observational studies where an association between genetic variants (such as *Single Nucleotide Polymorphisms* (SNPs) or *insertions-deletions* (indels)), identified via genotyping, or sequencing across the whole genome of a given species, and a phenotypic trait, e.g. disease status or biochemical concentration levels, is attempted. Below we provide examples of biological questions answered by genome-wide association studies.

Analysis	Purpose	Discoveries
GWAS	detecting trait-SNP associations	~10,000 robust associations with diseases and disorders, quantitative traits, and genomic traits
Genome-wide CNV analysis	detecting trait-CNV associations	hundreds of associations with diseases and disorders
Genome-wide assessment of LD	quantifying genome architecture	large variation in LD in the genome
Estimation of SNP heritability[a]	genetic architecture	large proportion of genetic variation captured by common SNPs
Estimation of genetic correlation[a]	detecting and quantifying pleiotropy	pleiotropy is ubiquitous
Polygenic risk scores[a]	detecting pleiotropy; validating GWAS discoveries	out-of-sample prediction works as expected; detection of novel trait associations
Mendelian randomization[a]	testing causal relationships	replication of known causal relationships; empirical evidence of observational associations that are not causal
Population differences in allele frequencies	reconstructing human population history; detecting selection	genetic structure can mimic geographical structure; evidence of natural selection
Trait GWAS with -omics GWAS[a]	fine-mapping; detecting target genes; function	two-thirds of GWAS-associated loci implicate a gene that is not the nearest gene to the most associated SNP

[a]These analyses can be performed with GWAS summary statistics.

Table 2.1: The role of GWAS SNP arrays in human genetic discoveries, (Visscher et al., 2017)

2.1 Feature Selection

As discussed in Motivation, associating genetic variants, such as SNPs, to a specific phenotypic trait, is practically an FS problem. Not until recently, GWAS tackled this problem from a univariate association perspective, which is a naive approach in identifying causative variants, often producing high false-positive rate (Type I error) (Waldmann et al., 2013). Moreover, univariate associations are bound to fail when additive effects come into play, i.e. numerous SNPs that jointly explain the phenotypic variation. This behaviour is often deteriorated due to the typical size ratio of such datasets, i.e. millions of predictive variables to thousands of samples, often called as a "small n - large p" problem. As a result, most recent, state-of-the-art, FS methods use a multivariate approach to such biological problems, correcting individual SNP's association via conditional information.

Another key characteristic of genome-wide datasets is the inherent population structure, i.e. linkage disequilibrium between genomic regions that are physically unlinked (Klasen et al., 2016). Confounding by population structure leads to correlations between such regions and inflation of corresponding test statistics (Segura et al., 2012). Many methods have been proposed to account for such spurious associations that will be discussed briefly in the following subsections.

Finally, to date, the vast majority of literature handles these analyses from a linear model-fitting perspective, since an exhaustive search of all possible combinations of non-linear relationships between such a large number of predictor variables, is prohibitive. Due to linear interpretations, identification of single SNPs that are associated non-linearly, e.g. quadratically, cosine etc, with the phenotype, is not possible, especially when epistatic phenomena (interaction between SNPs) produce the corresponding phenotype.

2.1.1 Weak Population Structure

LASSO

In the presence of weak population structure, e.g. *arabidopsis thaliana* plants of one race, there is no need in accounting for linkage disequilibrium between non-neighboring genomic regions. Under this context, *Least Absolute Shrinkage & Selection Operator* (LASSO) (Tibshirani, 1996) is perhaps the most popular and cited algorithm for feature selection. LASSO belongs to the category of penalized regression algorithms, where the calculations of the multivariate regression model's coefficients and feature selection are carried out simultaneously. This is accomplished by imposing a penalty on the sum of predictor variables' coefficients formulating a minimization problem:

$$\hat{\beta}_0, \hat{\beta} = argmin\left[\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j X_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right] \tag{2.1}$$

The first term of the above equation calculates the coefficient values, while the second term is an $l_1 - norm$ penalized least-squares criterion, wherein penalty, or

regularization, parameter λ controls the amount of shrinkage imposed on coefficients' values [1].

Ridge Regression

On the other hand, *Ridge Regression* (RR) (Hoerl and Kennard, 1970), calculates the regression coefficients through an $l_2 - norm$ penalized least-squares criterion: $\lambda \sum_{j=1}^{p} \beta_j^2$. RR deals with the shortcomings of LASSO when predictor variables are correlated, i.e. correlated predictor variables borrow strength from each other (Waldmann et al., 2013), something that is often the case in a genome-wide dataset with or without strong population structure (linkage between neighboring SNPs or LD between distant regions, respectively).

Elastic Net

In practice, a hybrid method that incorporates LASSO and RR, termed *Elastic Net* (EN), has proven to minimize the drawbacks and maximize the advantages of both methods at the cost of tuning more parameters:

$$\hat{\beta}_0, \hat{\beta} = argmin \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \left[(1-\alpha)\beta_j^2 + \alpha|\beta_j| \right] \right\} \quad (2.2)$$

Here, α represents the weighted penalty from each method, e.g. $\alpha = 0$ results in pure Ridge Regression, while $alpha = 1$ in pure LASSO.

Orthogonal Matching Pursuit

Orthogonal Matching Pursuit (OMP) (Pati et al., 1993) is a greedy forward-search algorithm behaving similarly to the forward regression method. OMP bases its selection strategy on correlation between predictor variables and consecutive model-fitting residuals. Specifically, once a variable is selected, the next one is determined by the maximum correlation of a predictor variable and the residuals of the previous fitted model, i.e the correlation after removing the effect of all the previously selected variables. OMP and LASSO share theoretical properties and due to the fact that they are both residual-based algorithms (substantially low number of fitted regression models) renders them computationally highly efficient feature selection methods.

2.1.2 Strong Population Structure

In the presence of strong population structure, some methods have been developed that try to reverse the effect of such spurious correlations and identify the truly associated with the phenotype variables. Perhaps the most popular and simple solution that has been shown to perform well in plants, animals and humans, comes from the application of *Linear Mixed Models* (LMM). LMMs are ideally used

[1] Through this penalty criterion, the coefficients of non-informative independent variables are shrunk towards zero, thus performing feature selection by discarding such features.

when data structure is hierarchically organized, e.g. different measurements from distinct groups. In this case, distinct groups are reflected by SNP sets that explain the relatedness in population groups (population structure). A simple linear mixed model is expressed as:

$$y = \mathbf{X}\beta + \mathbf{Z}u + \epsilon \tag{2.3}$$

where,

- $\mathbf{X}$, $n \times p$ data matrix of n samples and p predictors,
- β, $p \times 1$ fixed-effects regression coefficients,
- $\mathbf{Z}$, $n \times k$ design matrix of k random effects (distinct groups) and
- u, $k \times 1$ random-effects regression coefficients.

Segura et al. (2012) proposed a multivariate Linear Mixed Model approach for analyzing GWA datasets, termed Multi-locus Mixed Model (MLMM), claiming superior performance over univariate linear and univariate mixed model approaches, while they thoroughly tested it on simulated and real (*arabidopsis thaliana* and human) data. In forward phase, MLMM includes features behaving much like forward stepwise regression, and approximates the genetic (design matrix) and error variance at each inclusion step. During backward phase, features that are most likely falsely identified as causative, are removed, resulting in a typical forward-backward stepwise with mixed-effects regression algorithm.

Waldmann et al. (2013) utilized a different population structure correction strategy based on spectral graph theory. According to this techinque, a number of the most informative eigenvectors from the genomic variants is used as fixed covariates prior to initializing a feature selection procedure. In their work they extensively studied Elastic Net and tuned the regularization parameter λ along with the weight parameter α across a $10 - fold$ cross validation, on simulated and real datasets. Regarding a specific simulated dataset, *QTLMAS 2010*, they tested how their proposed methodology performed with and without the population structure correction, summarized below:

		Lasso	EN09	EN075	EN05	EN03	EN01	EN005	fdr
No pop. struct. corr.	Selected SNPs	161	176	168	219	232	326	454	78
	minMSE + 1SE	0.2825	0.3082	0.3822	0.5331	0.9087	2.6208	4.8283	–
Pop. struct. corr.	Selected SNPs	82	87	87	92	98	161	240	134
	minMSE + 1SE	0.2421	0.2594	0.3114	0.4673	0.7751	2.1707	4.0467	–

Table 2.2: Results from the analysis of the simulated QTLMAS 2010 workshop data with and without correction for population structure (using eigenvectors from spectral graph analyses). The simulated pedigree consists of 3226 individuals from 5 generations. The continuous trait was controlled by 37 QTLs that had 364 SNPs with $r^2 > 0.1$. The stopping criteria for λ were obtained as the average of ten $10 - fold$ cross validation runs at minimum MSE plus 1 standard error. The values of the elastic net (EN) refers to the penalty weight α (e.g., EN005 is elastic net with $\alpha = 0.05$). FDR refers to the SNPs selected by the single marker regression local false discovery rate method, (Visscher et al., 2017)

Klasen et al. (2016) proposed *Quantitative Trait Cluster Assosiation Test* (QTCAT), an alternative method for population structure correction: initially, a hierarchical clustering based on the pairwise correlations between all markers is performed and since testing for all markers is computationally expensive, an approximate greedy method is implemented instead. Next, this generated hierarchical structure is used in the association with the phenotype testing procedure. Starting from the tree's root, where all variants (covariates) are joined, the algorithm moves to deeper sub-clusters only if the inference testing yields statistically significant results. In many cases, the algorithm will return before reaching a leaf, i.e. before a single-marker (SNP) is tested. Finally, the samples are randomly split into 2 disjoint groups, B times where, from group I the most representative covariates are extracted via LASSO [2] and tested for significance on group II. In our work, we chose QTCAT to be compared to our proposed method *generalized Orthogonal Matching Pursuit* (gOMP), due to several important reasons such as, dataset and code availability, methodologically correct simulation strategy, accounting for population structure and superiority over linear mixed models.

2.2 Multiple Feature Selection

To date, most of feature selection methods, return a single subset of predictive features. Lagani et al. (2017) state that *"it is often the case that multiple feature subsets are approximately equally predictive for a given task"*. This is especially true in biology, where natural selection recruits redundancy as a "backup plan" to shocks and adverse events. The SES algorithm (Tsamardinos et al., 2012; Lagani et al., 2017) belongs to the class of constraint-based, feature selection algorithms (Tsamardinos et al., 2006), a class of algorithms that ground their root

[2]Penalty parameter λ is tuned through a 10-fold cross-validation.

in the theory of Causal Analysis (Spirtes et al., 2000). Constraint-based algorithms have recently proven to be able to retrieve highly predictive signatures (Aliferis et al., 2010). From an algorithmic point of view, given a data set D defined over a set of n variables / predictors V and a target variable T (a.k.a. outcome), constraint-based feature selection methods repetitively apply a statistical test of conditional independence in order to identify the subset of variables that can not be made independent by the outcome given any other subset of variables in V. We denote with $ind(X, T|W)$ any statistical test able to provide a p value $p_{XT.W}$ for assessing the null hypothesis that the variables X and T are conditionally independent given a set of variables W. Depending on the nature of the variables involved in the test (e.g., categorical, continuous, censored) the most appropriate conditional independence test must be chosen. Finally, it is worthwhile to note that under some additional assumptions, constraint-based methods have the interesting property of uncovering (part of) the causal mechanism that produced the data at hand. The SES algorithm implements an additional heuristic in order to retrieve multiple sets of features that are equally predictive.

In our work, we borrow SES properties and adjust them to a residual-based algorithm, OMP. On top of that, we argue that multiple solutions in genomic datasets are closely related to LD and population structure, thus through this modification we achieve simultaneously a 3-part goal:

- Feature selection.
- Account for population structure.
- Identification of equally predictive signatures.

Chapter 3

Methods

In this section we will describe in detail the statistical and machine learning methods used throughout the analyses of this work, as well as the design for the experimental validation through simulated datasets.

In order to produce a predictive final model and evaluate its performance robustly, we utilize a fully automated supervised machine learning protocol, named **JAD** (Just Add Data). Under this protocol a complete analysis is carried out, ensuring that tasks such as data preprocessing, feature selection, model selection and performance estimation avoid common methodological errors and the pitfalls of overfitting (Borboudakis et al., 2017; Orfanoudaki et al., 2017).

With regards to feature selection we utilize the *generalized Orthogonal Mathcing Pursuit* algorithm (Tsagris et al.), a variant of *Orthogonal Matching Pursuit*, which is scalable to high-dimensional problems, where this is the case with genome-wide SNP datasets. On top of that, we propose a method for statistically equivalent features discovery, similar to (Tsamardinos et al., 2012; Lagani et al., 2017).

For the experimental validation subsection, a simulation method identical to (Klasen et al., 2016) is used in order to generate continuous target variables for a given genotypic dataset acquired from easyGWAS platform (Grimm et al., 2017). This serves as a two-fold evaluation procedure:

- Establish a ground truth in terms of maximum predictive performance (reflected by heritability) and selected features associated with the phenotype.
- Comparison with the published feature selection method, QTCAT, in terms of predictive performance, SNP discovery and computational time.

3.1 Tools

3.1.1 Just Add Data (JAD) protocol

Every data analysis needs a suitable protocol implementation in order to produce generalizable, accurate results. For the purposes of simulated and real datasets

analyses, we relied heavily on **JAD Bio™** philosophy, a trademark of Gnosis DA. Between the dataset input and the results interpretation lie numerous steps of data handling along with the suitable algorithms and their hyper-parameters. **JAD Bio™** deals with all the in-between steps in an automated manner, ensuring methodological correctness (unbiased performance estimation and avoidance of overfitting). The main steps of this protocol along with some corresponding algorithms are listed below in sequential order:

1. **Data Partitioning**
   - hold-Out
   - k-Fold
   - Stratified variants of the above
2. **Data Preprocessing**
   - Standardization
   - Imputation
3. **Feature Selection**
   - SES
   - gOMP
4. **Modelling Algorithms (modellers)**
   - Support Vector Machines (SVM)
   - Random Forests (RF)
5. **Performance Estimation**
   - Bootstrap Bias Corrected Cross Validation (BBC-CV)

Table 3.1: Data handling steps

1) Data Partitioning

Partitioning the dataset in disjoint sets is vital for the learning procedure, model selection and estimation of performance on unseen data. For model selection, dataset is randomly split in 2[1] according to any of the methods described below, where one part acts as a training set for every configuration and each performance is calculated on the remaining set. Through this procedure, the best performing model is acquired. For performance estimation, the same reasoning

[1]Usually, a stratification of the splits, based on target's distribution, is used.

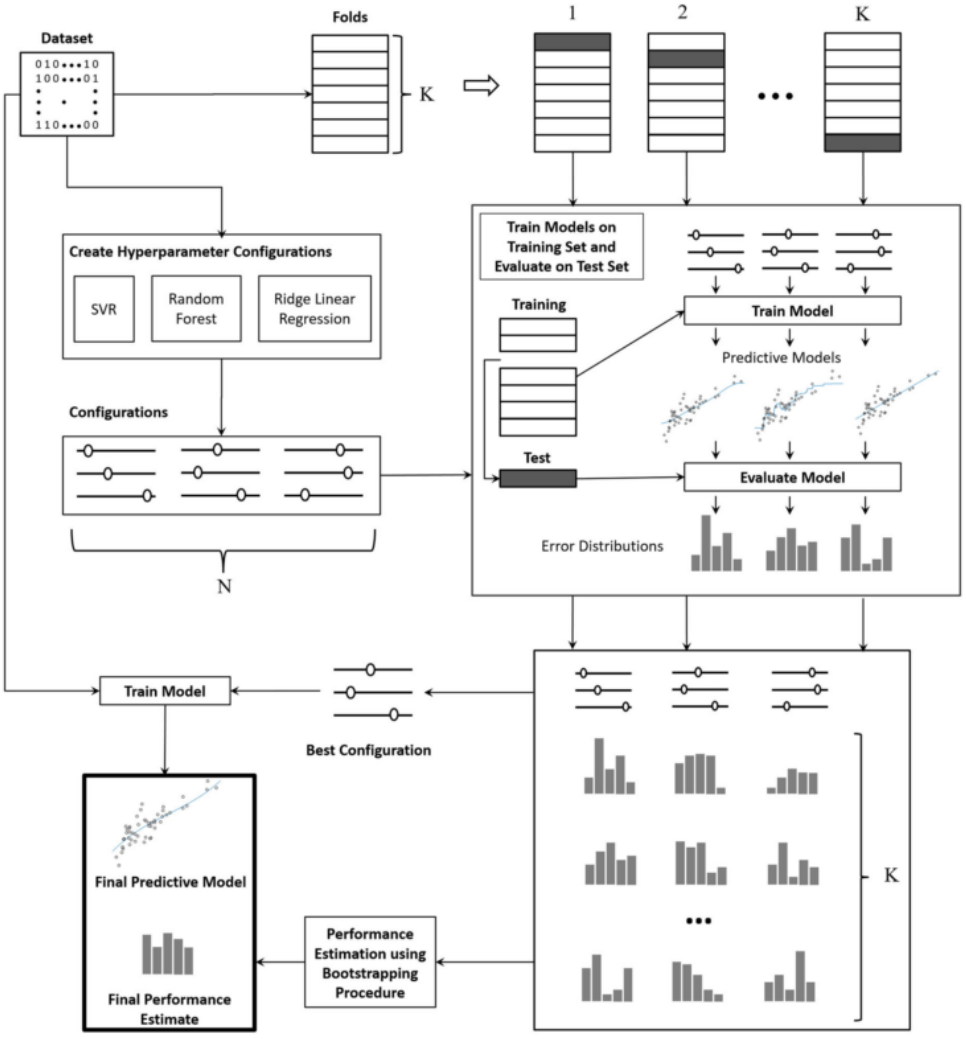


Figure 3.1: Flowchart of **JAD Bio**™ automated pipeline

can be applied in reporting the generalized performance on unseen data. In order to avoid methodological errors, a simultaneous model selection and performance estimation is required. One way to achieve this is by extending the partitioning to 3 disjoint sets, known as *Train-Validation-Test* protocol (see more at 5) Model Selection & Performance Estimation). Below we describe the two main partitioning methods depending on sample size.

- **hold-Out:** Split the dataset in 2 disjoint sets of arbitrary size, e.g. $20\%-80\%$ or $50\%-50\%$. A very simple partitioning method, particularly effective for large sample sizes.

- **k-Fold:** Split the dataset in k disjoint datasets of equal sample size. Repeatedly one set at a time acts as a validation set, while the union of the remaining $k-1$ sets, acts as a training set. *k-Fold*, or *cross-validation* (CV), can also be used for performance estimation by calculating the mean value of the performance achieved from every best model in k loops. This however will produce an optimistically biased estimation of the performance on unseen data, since validation and training sets are not disjoint as a whole[2]. Effective for small to medium sample sizes.

Stratification: When predicting a continuous or categorical target variable, value and class distribution respectively, should be taken into account. Through stratification we ensure that at any partitioning, train and test sets are represented by samples with similar distributions (can be applied in any partitioning method).

2) Data Preprocessing

Depending on the dataset type and/or the algorithms used in the pipeline, a preprocessing must be carried out, e.g.:

- **Standardization:** Basically a *z-score*[3] transformation, particularly important when variables have varying magnitude, e.g. height in cm and ratios between 0 and 1. Geometric algorithms such as *kNN* for modelling, or *gOMP* for feature selection could collapse if features are not standardized.

- **Imputation:** When dealing with real datasets, missing values is often the case. Excluding samples or variables that have at least one missing value could result in a particularly shrunk dataset with no statistical power. Inferring these values helps to overcome this problem, while common practices for imputation include, mean value, median etc.

3) Feature Selection

In section 2.1 the benefits and necessity of feature selection were thoroughly discussed. Here we refer the reader to the appropriate subsections which analyze the algorithms SES (mainly used in **JAD**) and OMP along with its proposed variant gOMP.

[2] For example during any loop, the $\frac{1}{k-1}$ of training samples would have been, at some point, members of a test set.

[3] $\frac{x-\mu}{\sigma}$, where x a value of a variable of a specific sample, μ the mean value of a variable across the samples and σ the standard deviation.

4) Modelling Algorithms (modellers)

- **SVM:** Support Vector Machines (Boser et al., 1992) can be used on classification or regression tasks. For classification, SVMs create a hyperplane that discriminates the 2 classes best. Imposing a penalty to a margin hyper-parameter (intuitively, the maximum distance allowed between 2 classes) gives control on overfitting and generalization. SVMs, through the use of kernel functions, can also handle non linear relations by transfering the data to higher dimensions.

- **RF:** Random Forests (Breiman, 2001) extend the idea of *Decision Trees* (DT) by repeatedly fitting trees using sampling with replacement (bootstrap aggregation) and as an ensemble method it averages the predictions or a majority voting takes place for regression or classification tasks respectively. This technique reduces performance variance without increasing bias, lacking however DT's interpretability.

5) Model Selection & Performance Estimation

Main goals of machine learning are the identification of the best configuration (sequence of statistical and algorithmic methods used, along with their hyper-parameters) in terms of predictive performance, as well as the accurate estimation of this performance on unseen data. In model selection, the identification of high-performing models is often called tuning and it refers to the algorithms used along with the specific corresponding hyper-parameters. These hyper-parameters affect the extent to which the algorithms are able to detect patterns in the data, the trade-off between overfitting and generalization, as well as the bias-variance trade-off (figure 3.2), to name a few. Here, **model** in Model Selection should not be confused with a modelling algorithm, as it refers to the collection of algorithms used in learning procedure for a given partitioning. For example, one **model**, or configuration, could be: imputation with median value → standardization → gOMP with $\Delta BIC = 6$ → SVM with linear kernel, $cost = 0.1$.

As discussed briefly in 1), a suitable data partitioning scheme is vital for model selection and performance estimation. In case of small or medium sample sizes, *Train-Validation-Test* protocol and *CV* will produce a biased performance estimation. To overcome this one could employ the *Nested Cross Validation* protocol, where an inner and an outer *CV* is utilized; First, the dataset D is partitioned into k_{outer} stratified folds ($i = 1 : k_{outer}$), while each $D \setminus D_i$ set is partitioned into k_{inner} stratified folds ($j = 1 : k_{inner}$). Model selection is performed in every i^{th} iteration on k_{inner} folds and the performance of the best model is calculated. Next, performance estimation is calculated from the k_{outer} best-performing models. This protocol produces unbiased estimation, but since it is computationally expensive, it is unsuitable for large sample sizes. **JAD Bio™** employs a bias-correction method (Tsamardinos et al., 2018b), rendering outer Cross-Validation unnecessary. The method is described below.

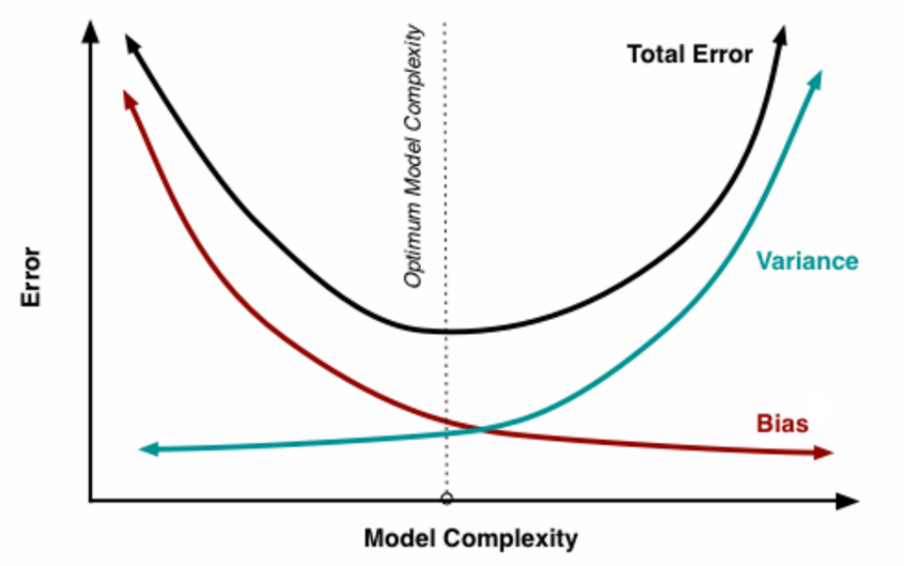


Figure 3.2: Prediction error as a function of model complexity. Blue line represents the variance-model complexity dependance, while the red line the bias. Dotted vertical line depicts the optimized bias-variance trade-off in terms of minimum total error.(*source*)

- **BBC-CV:** By the end of a *Cross Validation* training protocol, all prediction performances of the k disjoint validation sets are pooled together in a matrix Π of N rows (number of samples) and M columns (number of configurations)[4]. By sampling with replacement (bootstraping) N rows of Π, the best configuration is chosen and its performance is calculated on the out-of-sample predictions; that is, on samples that were not sampled during the bootstrap[5]. The above procedure is repeated B times (approximately $\times 100$) and the average corrected performance is calculated. This method underestimates performance, meaning that reported performance will be on average smaller than the actual on a test set. Moreover, the reported performance will have smaller bias and variance, similar to *source*, but with substantially smaller computational overhead, as no new models are fitted or trained during this method (Tsamardinos et al., 2018b).

3.1.2 Generalized Orthogonal Matching Pursuit algorithm

In the established version of the *OMP* algorithm (section 2.1.2), the norm-based stopping criterion is arbitrarily chosen and cannot be accurately defined without multiple runs and manual evaluation. On the other hand, **gOMP**, makes use of the *Bayesian Information Criterion* (BIC), which is preferred due to its flexibility and

[4]Each column contains the performance of j^{th} configuration or model across all samples.

[5]On average, the bootstrapped set will contain 63.2% of the original samples, while the rest 36.8% will be random copies of them.

the ability to quantify the model's fit quality objectively. Regarding flexibility, *"use of this stopping criterion, generalizes OMP's functionality on accepting numerous types of outcome variables, including multi-class, survival, left censored, counts and proportions to name a few, thus being able to handle various regression models"* (Tsagris et al.). On Algorithm 1 we provide the pseudocode of *gOMP*.

Forward Phase

Since *OMP* belongs to the category of residual-based algorithms, a *z-score* transformation is vital at the begining of the algorithm, as a preprocessing step. In order to select the first variable, a null model is fitted and its residuals[6] and null BIC are calculated. Next the correlation between all variables and the residuals is computed and the variable with the maximum absolute[7] correlation is chosen. Once the first selected variable is defined, the algorithm enters an iterative procedure; Every time a new variable is chosen for possible inclusion (candidate variable), the model including the candidate variable is fitted and current residuals and BIC are calculated. If the decrease in BIC between the current model and the previous model (without the candidate feature) is above a predefined threshold value, the variable is selected permanently, otherwise the process stops, excluding the currently selected variable. When a variable is permanently added to the selected variables set, the correlation between current residuals and all the remaining variables is updated. Again, the variable with the maximum absolute correlation enters the candidate set and the process begins a new iteration.

Backward Phase

When no new variable can enter the selected variables set, the algorithm can stop. In order to check for false positive variables a backward step is implemented. Given a set of selected variables, we start by removing one variable at a time and calculate the corresponding *BIC* score. Lower BIC score, corresponds to models that performed better even though a variable was removed. If the difference between the full model's BIC and the lowest BIC of a model with one removed variable is below a certain threshold, then this variable can be discarded indeed. The process continues until no further variable can be removed.

The above backward method, uses the same ΔBIC threshold for selection and discard of false positive variables. Alternatively, Zhang (2008) proposed a modified backward method based on a varying ΔBIC threshold; During *Forward Phase*, ΔBIC that allowed each variable to enter the selected variables set is stored. This data structure helps to keep the contribution of each selected variable to the full model, as well as the sequence in which each variable has entered the set.

[6]The residuals of a null model are basically the difference between the target variable y and its mean value $\bar{y}$.

[7]A high negative correlation between a variable and the residuals indicates reversely proportional relation.

Algorithm 1 gOMP

Input: Target variable $\mathbf{y}$, $n \times p$ data matrix $\mathbf{X}$, a selection threshold value tol_S and an equivalent threshold value α_{equiv}.

Output: A list of selected features $\mathbf{S}$ and a list of sets of equivalent features $\mathbf{E}$.

Forward Phase

$S = \emptyset$ // *Set of selected features*

$E = \emptyset$ // *List of sets of equivalent features*

$F = 1 : p$ // *Set of remaining variables to be considered for inclusion*

// *Initialization step*

$\mathbf{X} = zscore(\mathbf{X})$ // *Standardize data*

$[residuals, BIC_0] = fit(\mathbf{y}, \emptyset)$ // *Calculate residuals & BIC for null model*

// *Calculate r for each remaining variable against the residuals*

$r = corr(residuals, \mathbf{X}(:, F))$

$s^* = \arg\max_{j \in F}(|r_j|)$ // *Select variable that maximizes absolute r*

// *Update*

$S = S \cup s^*$

$F = F \setminus s^*$

// ***Main Loop***

while $R \neq \emptyset$ **do**

$[E, F] = equivalentSearch(residuals, s^*, F, \mathbf{X}, \alpha_{equiv})$

$F = F \setminus E_{s^*}$ // *Update remaining variables*

$[residuals, BIC_1] = fit(\mathbf{y}, \mathbf{S})$ // *Calculate residuals & BIC for current model.*

$\Delta BIC = BIC_0 - BIC_1$

if $\Delta BIC < tol_S$ **then**

$S = S \setminus s^*$

break

end if

// ***Calculate** r **for each remaining variable against the residuals***

$r = corr(residuals, \mathbf{X}(:, F))$

$s^* = \arg\max_{j \in F}(|r_j|)$ // ***Select variable that maximizes absolute** r*

// ***Update***

$S = S \cup s^*$

$F = F \setminus s^*$

$BIC_0 = BIC_1$

end while

Algorithm 1 gOMP (cont'd)

Backward Phase

// Set difference in BIC to infinity
$\Delta BIC = \infty$

// Initialize removed variable
$v^* = \emptyset$

while $\Delta BIC > tol$ **do**
  $S = S \setminus v^*$
  $BIC_S = BIC_{full}$ *// BIC of full model (every variable selected so far)*
  for $v \in S$ **do**
    $S' = S \setminus v$
    $BIC_v = BIC_{S'}$ *// BIC when v removed*
  end for
  // Find variable which minimizes BIC, if removed
  $v^* = \arg\min_{v \in S}(BIC)$
  $BIC^* = \min(BIC)$
  // Calculate BIC difference
  $\Delta BIC = BIC_S - BIC^*$
end while

During *Backward Phase*, tolerance value changes to the corresponding ΔBIC_i multiplied by 0.5, that is, half the ΔBIC_i score achieved at i^{th} iteration, when variable i entered the set.

Caching

In a machine learning pipeline, it is often desired to tune the tolerance (selection) hyper-parameter of *gOMP* in order to obtain the best-performing set of variables. Storing of ΔBIC data structure contributes to the substantial reduction of computational overhead of consequent runs of the algorithm (different values of ΔBIC). Since *gOMP* includes the candidate features in descending order of ΔBIC, only one run of the algorithm, with the minimum desired tolerance, is required; In a consecutive run, we retain only the variables up to which the first occurrence of a ΔBIC lower than the current threshold is met. In figure 3.3 we present a graphical representation of ΔBIC caching structure.

Statistically equivalent features

As discussed in section 2.2.1, *SES* is a powerful method for statistically equivalent feature discovery. In order to extend *gOMP* functionality, we applied a similar strategy to the equivalence problem, addressed by the *equivalentSearch* function.

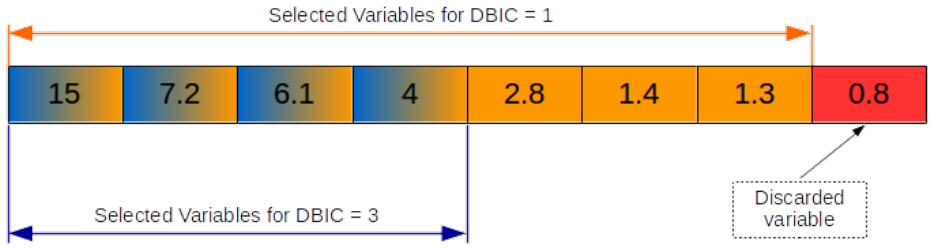


Figure 3.3: ΔBIC caching structure for tolerance hyper-parameters of 1 and 3. Each box contains the corresponding BIC score when a variable is considered for inclusion. First, *gOMP* runs with $tol_S = 1$ and 7 features are selected, while 8^{th} variable is discarded. Next, for $tol_S = 3$ we retain only the 4 first features of the first run.

After a candidate feature for inclusion (s^*) is defined, the partial correlations between the residuals, r and interchangeably variable s^* and the remaining variables F, are calculated like below:

- Calculate all partial correlation between r and $F_i \in F$, given s^*: $\boxed{\hat{\rho}_{F_i}(r, F_i|s^*)}$
- Calculate all partial correlation between r and s^*, given $F_i \in F$: $\boxed{\hat{\rho}_{s^*}(r, s^*|F_i)}$
- Compute the *p-values* of zero-correlation hypothesis for both partial correlation vectors.
- Identify as equivalent features those that both *p-values* are above an equivalency threshold, a_{equiv}.

Algorithm 2 equivalentSearch

Input: Residual vector **residuals** of previous fitting model, currently selected variable s^*, remaining variables R, $n \times p$ data matrix $\mathbf{X}$, list of sets of equivalent features $\mathbf{E}$ and an equivalent threshold value α_{equiv}.

Output: List of sets of equivalent features $\mathbf{E}$ and remaining variables R.

$\mathbf{X_{s^*}}$ // *Data vector of currently selected variable.*
$\mathbf{X_R}$ // *Data matrix of remaining variables.*

//***Calculate partial correlations***
$pvC_1 = pcorr\,(\mathbf{residuals}, \mathbf{X_R}|\mathbf{X_{s^*}})$ // *p-value vector conditioned on selected variable.*
$pvC_2 = pcorr\,(\mathbf{residuals}, \mathbf{X_{s^*}}|\mathbf{X_R})$ // *p-value vector conditioned on remaining variables.*

$E_{s^*} = (pvC_1 \wedge pvC_2) > \alpha_{equiv}$ // *Assign equivalences for* s^*.
$R = R \setminus E_{s^*}$ // *Update remaining variables*

Distributed version

GWAS analysis is by default a high-dimensional problem, where the dataset consists of hundred of thousands, or even millions, of variables, here SNPs. Most conventional feature selection methods are unable to handle that kind of memory overload and processing requirements. Even if the computations are carried out in a high-performance-computing (*HPC*) cluster, the fact that parallelization is not exploited, renders these methods inefficient for that kind of analyses. Here we propose a distributed version of *gOMP* scalable to high-dimensional datasets, operating exactly as *gOMP* and producing identical results.

Parallelization of *gOMP* stems from the fact that at every iteration of the algorithm, the most correlated with the residuals variable enters the candidate set. This selection criterion allows the segmentation of the dataset feature-wise by running *gOMP* in C seperate chunks, storing the most correlated variable in each chunk and selecting the one with the highest correlation coefficient across all chunks. The number of chunks, C, and the sequence of chunk processing is independent of the final selected variables, thus parallelization is only limited by each computing unit's resources (in fact a high-dimensional problem, e.g. with 2×10^3 samples and 10^6 features, is solvable in an ordinary home-PC). Below we present the pseudocode for distributed *gOMP*.

Algorithm 3 Distributed gOMP

Input: Target variable $\mathbf{y}$, $n \times p$ data matrix $\mathbf{X}$, a selection threshold value $\mathbf{tol_S}$, equivalent threshold value α_{equiv}, and number of chunks C.

Output: A list of selected features $\mathbf{S}$ and a list of sets of equivalent features $\mathbf{E}$.

// ***Forward Phase***
$S = \emptyset$// *Set of selected features*
$F = 1 : p$ // *Set of remaining variables to be considered for inclusion*
$E = \emptyset$// *List of sets of equivalent features*

while S changes **do**
  r // *List of correlation coefficient vectors*
  for $i = 1 : C$ **do** // *Parallelization*
    if 1^{st} run **then**
      $F_i = 1 : p_i$ // *Set of remaining variables for current chunk*

      // *Initialization step*
      $\mathbf{X_i} = zscore(\mathbf{X_i})$ // *Standardize data*
      $[residuals_i, BIC_{i1}] = fit(\mathbf{y}, \emptyset)$ // *Calculate residuals & BIC for null model*

      // *Calculate r for each remaining variable against the residuals*
      $r_i = corr(residuals_i, \mathbf{X_i}(:, F_i))$
      $s_i^* = \underset{j \in F_i}{\arg\max}(|r_{ij}|)$ // *Select variable that maximizes absolute* r_i

Algorithm 3 Distributed gOMP (cont'd)

else
$F_i = F_i \setminus S \cup E$
$E_i = equivalentSearch(residuals_i, s^*, F_i, \mathbf{X_i}, \alpha_{equiv})$
$[residuals_i, BIC_{i1}] = fit(\mathbf{y}, S)$ *// Calculate residuals & BIC for current model*
$\Delta BIC = BIC_0 - BIC_1$
if $\Delta BIC < tol_S$ **then**
$S = S \setminus s^*$
break
end if
*// **Calculate** r **for each remaining variable against the residuals***
$r_i = corr(residuals_i, \mathbf{X_i}(:, F_i))$
$s_i^* = \arg\max_{j \in F_i}(|r_{ij}|)$ *// **Select variable that maximizes absolute** r_i*
end if
end for

*// **Update***
$C^* = \arg\max_{i \in C}(r_i.)$ *// **Select chunk that maximizes** r*
$s^* = s^*_{C^*}$ *// **Choose best variable across chunks***
$S = S \cup s^*$
$E = \bigcup_{i \in C} E_i$ // **Unify equivalences**
$BIC_0 = BIC_{C^*1}$
end while
*// **Backward Phase** // **Set difference in BIC to infinity***
$\Delta BIC = \infty$
*// **Initialize removed variable***
$v^* = \emptyset$
while $\Delta BIC > tol$ **do**
$S = S \setminus v^*$
$BIC_S = BIC_{full}$ *// **BIC of full model (every variable selected so far)***
for $v \in S$ **do**
$S' = S \setminus v$
$BIC_v = BIC_{S'}$ *// **BIC when** v **removed***
end for
*// **Find variable which minimizes BIC, if removed***
$v^* = \arg\min_{v \in S}(BIC)$
$BIC^* = \min(BIC)$
*// **Calculate BIC difference***
$\Delta BIC = BIC_S - BIC^*$
end while

3.2 Experimental Evaluation

In this section we describe the experimental setup used for evaluating the *gOMP* feature selection method through simulation studies and real datasets. Specifically in simulated datasets, subsection 3.2.1, we compare *gOMP* against *QTCAT* on exactly the same background: simulated phenotype, sample splitting for training and testing, modelling algorithms, to name a few. In subsection 3.2.2, we utilize *gOMP* for SNP discovery and evaluate their predictive performance on real, human-disease datasets.

3.2.1 Simulated Datasets

As mentioned before, the simulation procedure which generates the phenotype is identical to (Klasen et al., 2016). Here, we will highlight the key points of the simulation strategy, as well as the comparison protocol for **gOMP** and **QTCAT**.

The dataset used is acquired from the *easyGWAS* platform, available here and it consists of 1,307 genotyped samples of the species *Arabidopsis thaliana*. The simulation strategy exploits the real genetic profile of the samples in order to account for the underlying complicated mechanisms, such as heritability, in sets of populations. Alternatively, one could simulate the genotype as well, but that would require using models of mutation rate, crossover, etc, or even defining gene-rich regions, in order to produce realistic genetic profiles. Using real, genotyped data, overcomes this barrier and is only limited by the maximum number of samples used, in this case 1,307 which is statistically adequate. In Table 3.2 we provide basic information on this dataset.

Species	Arabidopsis thaliana
Dataset Name	AtPolyDB (call method 75, Horton et al.)
Dataset Build	TAIR9
# Samples	1,307
# Chromosomes	5
# SNPs	214,051
# SNPs in Gene Regions	28,496
Dataset Homozygous	Yes

Table 3.2: Simulation dataset overview

Firstly, a SNP "pool" is created from these SNPs that belong to previously known gene regions. This is to ensure that selected SNPs originate from areas that could affect the phenotype in a biologically realistic way. Next, probabilities dictated by a statistical distribution, are assigned on each SNP position, using Gaussian (normal), or gamma[8] probability density function. Given a probability

[8]For example, if gamma *pdf* is used, neighboring SNPs of a specific region have higher probability to be chosen as associative.

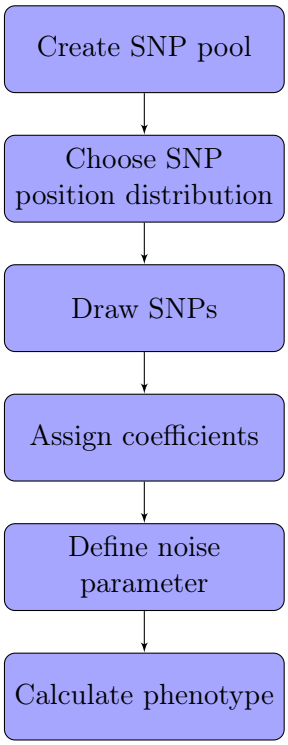


Figure 3.4: Simulation flowchart

distribution, a predefined number of SNPs is drawn randomly acting as independent variables affecting the phenotype (dependent variable). The linear model's coefficient for each associative selected SNP is chosen at random from a Gaussian distribution. Finally, in order to avoid producing a deterministic phenotype relationship, a statistical noise parameter is added which simulates a random enviromental effect and reflects the heritability of a set of SNPs. The magnitude of the noise parameter is such that it tends to match the unexplained variance of the linear model. In other words, the coefficient of determination, R^2 of this linear model, approaches the user-defined parameter, h^2. The simulated phenotype is continuous, resulting in a regression problem. In figure 3.1 we present a flowchart of the simulation strategy.

In order to provide a fair comparison between *gOMP* and *QTCAT* (as feature selection methods), a common pipeline is set up. This is performed in terms of **1)** parameters in simulation scenarios, **2)** splitting of samples in train-validation-test sets, as well as in **3)** modelling algorithms and their hyper-parameters. Regarding feature selection hyper-parameters, **gOMP** and **QTCAT** require only one; **4)** a selection threshold. As discussed previously, gOMP and QTCAT use ΔBIC and $p - value$ respectively, thus a consistent transformation between these two is required. Below we provide further insight on these aspects of the automated simulation pipeline.

1. **Simulation Scenarios**
   For phenotype simulation, 3 parameters are required; **1)** SNP position distribution (*Gaussian* or *gamma*), **2)** Number of associated SNPs and **3)** heritability, h^2 parameter. For the purposes of this evaluation, we concluded on 4 scenarios based on the aforementioned parameters:

   - **Distribution**: Gaussian, **Number of SNPs**: 20, $\mathbf{h^2}$: 0.7
   - **Distribution**: Gamma, **Number of SNPs**: 20, $\mathbf{h^2}$: 0.7
   - **Distribution**: Gaussian, **Number of SNPs**: 50, $\mathbf{h^2}$: 0.7
   - **Distribution**: Gaussian, **Number of SNPs**: 150, $\mathbf{h^2}$: 0.4

   Each scenario is repeated 50 times, producing 50 different phenotypes and sets of associated SNPs, creating enough simulation instances for statistical evaluation.

2. **Sample splitting**
   As discussed in Model Selection & Performance Estimation, a common tactic in machine learning regarding model selection and performance estimation is splitting the dataset in 3 disjoint sets for training, validation and testing on unseen data. Here, for each repetition and scenario, we choose to hold out a percentage of 10% of the samples which will be used to test the final model produced by the automated pipeline. Furthermore, a 50% of the remaining samples will be used for training (feature selection and modelling), while the rest 50% will be used to estimate the performance of the training procedure. Each feature selection algorithm, **gOMP** and **QTCAT** will be trained, validated and finally tested on exactly the same sample splits.

3. **Selection threshold**
   Both feature selection methods incorporate a parameter which controls the number of features selected. **QTCAT** uses a *p-value*, where higher values indicate larger signature size and less control over false positive features. On the other hand, **gOMP** uses ΔBIC, the relative drop of BIC scores of two successive statistical models. Here, larger values of ΔBIC correspond to stricter selection criteria, thus *p-value* and ΔBIC are reversely proportional. Generally, BIC form is given by:

$$BIC = -2ln(\hat{L}) + k \cdot ln(n) \tag{3.1}$$

   where $\hat{L}$ the maximized value of the likelihood function of the model, while for regression tasks:

$$BIC = n \cdot ln(\frac{2\pi \cdot \sum e_i}{n}) + n + k \cdot ln(n) \tag{3.2}$$

   where:

- n: Number of samples.
- e_i: The residuals per sample of current model.
- k: Difference in number of independent variables in comparing models[9].

Residuals cannot be known beforehand, so this type of BIC obstructs the determination of threshold parameters for **gOMP**, however when sample size is sufficiently large BIC can be rewritten as:

$$BIC = X^2_{1-\alpha,df} + k \cdot ln(n) \tag{3.3}$$

and for $df = 1$ the formula reduces to:

$$BIC = X^2_{1-\alpha,1} + ln(n) \tag{3.4}$$

where $X^2_{1-\alpha,1}$ the statistic of a chi-square distribution with $1-\alpha$ confidence level and 1 degree of freedom. This formula depends only on sample size and a predefined significance level, allowing a direct comparison with **QTCAT**. For the purposes of these scenarios we used 10 *p-values* logarithmically spaced between the range 10^{-6} and 0.8 (as in Borboudakis (2018)) for **QTCAT** and the corresponding BIC values for **gOMP**.

4. **Modellers**
From the modelling algorithms discussed in 3.1.1, we will use only *Random Forests* for the simulation analysis, since we are interested more in a relative comparison between the two feature selection methods and an exhaustive search of the universally best modelling method is not of grave importance. For the same reason, along with computational time speedup, we limit the number of hyper-parameters of RFs to:

   - *minLeafSize*: $[5, 10]$
   - *numPredictorsToSample*: $[0.5, 1, 1.5]$

   producing a total of 6 modelling configurations.

Taking all the above into account, in a single repetition and simulation scenario, a total number of 60 (10 FS configurations $\times$ 6 modelling configurations) configurations will be trained for each of the feature selection methods.

3.2.2 Real Datasets

Simulation studies help to prove, or at least provide insight, in *gOMP's* theoretical properties, which will be used to rely on when analyzing real data without

[9]When calculating ΔBIC, only one extra variable is examined for inclusion, thus if a previous model consists of p independent variables, the current one will have $p+1$, resulting in $df = (p+1) - p = 1$.

knowledge of the underlying mechanism that generates the phenotype (retrieval of causative SNPs).

For the purposes of Real-data analyses we downloaded several disease (case) datasets from European Genome-phenome Archive. Specifically, control datasets where downloaded seperately and each analysis consists of a triple merging between 2 different control datasets and 1 case dataset. We also provide one additional case-control dataset from an online data-analysis challenge (DREAM challenge). Below we present the datasets used:

Name	***Code***	***Platform***	***Samples***	***Status***
58C	EGAD00000000022	Illumina 1.2M	3,000	control
NBS	EGAD00000000024	Illumina 1.2M	3,000	control
Ulcerative Colitis	EGAD00000000025	Affymetrix 6.0	2,869	case
Parkinson's	EGAD00000000057	Illumina 610K Quad	1,705	case
Multiple Sclerosis	EGAD00000000120	Human670 - QuadCustom v1	11,375	case
Psoriasis	EGAD00010000124	Illumina 670K - Illuminus	2,622	case
Ankylosing Spondylitis	EGAD00010000150	Illumina 670K - Illuminus	2,005	case
Schizophrenia	EGAD00010000262	Affyemtrix 6.0 - CHIAMO	3,019	case
Pharmacoge-nomic Response to Statins	EGAD00010000282	Affymetrix 6.0 - CHIAMO	4,134	case
Barretts Oesophagus	EGAD00010000506	Illumina 670K - Illuminus	1,091	case
Reumatoid Arthritis	DREAM challenge	Batches from several platforms	2,022	case & control

Table 3.3: Real datasets overview

For each genotypic dataset we filtered out SNPs and samples that had missing values over 5%, in that order. The remaining missing values were imputed through replacement of the most frequent, column-wise or row-wise respectively, value. At this point we have to highlight that no additional filter has been applied, i.e. discard of SNPs with minor-allele frequency below a certain threshold, or SNPs that deviate from Hardy-Weinberg equilibrium. Finally, we chose to split each dataset in 2 disjoint sample sets and apply the aforementioned pipeline in each split independently, while the remaining set should act as a test set. On top of that, we repeated each sample-splitting scheme a total of 3 times.

3.3 Performance of Multiple Solutions

In paragraph Statistically equivalent features and in Algorithm equivalentSearch we explained the strategy on identifying equivalent features to already selected ones. This procedure produces a list of equivalent features on each member of the reference signature, visualized below in figure 3.5:

$$\{f_5, f_2, f_{16}, f_9\} \quad \text{Reference Signature}$$

$$\left\{\begin{bmatrix} f_5 \\ f_{25} \\ f_{10} \end{bmatrix}, [f_2], \begin{bmatrix} f_{16} \\ f_{100} \end{bmatrix}, [f_9]\right\} \quad \text{Multiple Solutions}$$

Figure 3.5: An arbitrary example of multiple solutions data structure. $\mathbf{f_i}$ entries denote the i^{th} feature of a given dataset, while each reference signature's feature equivalent list of features, appears in the respective position of *Multiple Solutions'* data structure.

Although, equivalent features for each selected variable are computed through a meticulous process, formulating alternative signatures could, theoretically, produce inconsistencies in terms of predictive performance. Since, statistical error resides in every determination of equivalent features for a specific variable of the reference signature, this error is propagated and magnified when formulating alternative signatures comprised by many equivalent features.

In order to assess equivalent combinations of equivalent features, we use a further post-processing filtering of every possible such combination. To do so, we propose a test statistic suitable for comparing the performances between equivalent signatures and the reference one. Below we describe this method.

3.3.1 Proposed Equivalence test-statistic

Quang H. Vuong (1989) proposed a suitable variance test for model selection, i.e testing statistical models that fit the data equally well, based on the variance of the log likelihood-ratio between both models, which is zero under the null hypothesis. Similarly, Borboudakis (2018) extended this idea to testing performance equivalence using permutation-based techniques. In any case, the values of the sample-sets used for the equivalence test can be the actual performance or loss metric produced by the reference and the alternative signature computed for each sample, e.g. deviance for logistic regression, or mean-squared error for regression.

Here, we formulate a test statistic inspired by both studies, defined as:

$$t_{equiv} = \frac{1}{n}\sum_{i=1}^{n} ln\left(\frac{L_i^{ref}}{L_i^{alt}}\right) \tag{3.5}$$

where:

- n: Number of samples.
- L_i^{ref}, L_i^{alt}: The loss metric for each sample in reference and alternative signature respectively.

Under the null hypothesis, this test-statistic will be zero, meaning that signatures are identical in terms of predictive performance, while due to log-ratio, we expect it to be distributed symmetrically. In order to determine the theoretical distribution of the statistic, we permute B times, e.g. 1000, the nominator and denominator of the log-ratio on a randomly selected number of samples and calculate the corresponding statistic, $_P$. The above procedure results in Bt_{Pi} values of the underlying distribution. The *p-value* of the observed statistic, T_{obs} is extracted by calculating the number of times that $|T_{obs}| > |t_{Pi}|, for\ i = 1 : B$. The resulting distribution is indeed symmetrical as shown below.

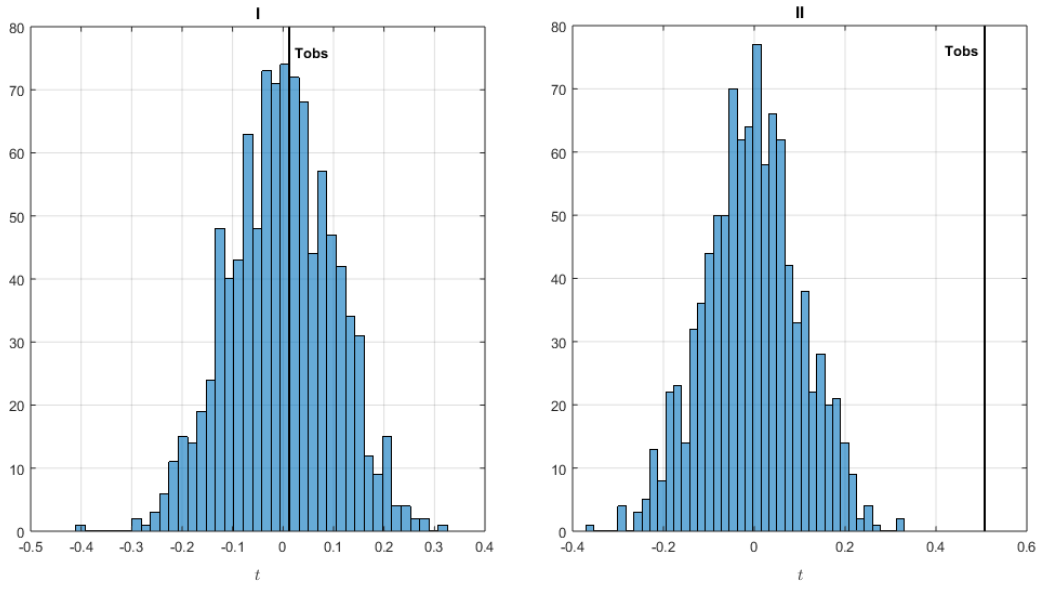


Figure 3.6: Distribution of proposed statistic acquired from sign permutations. I) Failure of rejection of the null hypothesis of equivalent signatures, II) Rejection of null hypothesis. Figure I depicts the statistic of a signature produced through simulation studies, while figure II, when “contaminating" signature I with random features.

3.4 Population Structure

In any GWA study population structure, i.e. correlation due to linkage disequilibrium between causative loci and physically unlinked regions, is one of the most frequent root causes of falsely associated SNPs with the phenotype. Klasen et al. (2016) demonstrated that *QTCAT* is insusceptible to such structure by applying their method to a dataset with inherent strong population structure (Horton et al.). On top of that, their method performed better in comparison with *Linear Mixed Models* (LMM) that take into account each population's variance and is the default solution to such problems.

In order to assess *gOMP's* ability of filtering out, or disregarding such features, we use a specific strategy which tries to evaluate if a selection bias occurs during *gOMP's* run. This strategy is as follows:

1. Perform a *Principal Component Analysis* on the same dataset used in simulations studies (strong population structure) and project the samples on the first two eigenvectors with the larger eigenvalues[10].

2. Use gaussian mixture models to define the 2 most representative hyper-population clusters (A and B). These two clusters will be comprised by populations that are most relative with each other internally, and most differentiated externally.

3. Implement a double Train-Validation-Test protocol, where at first population cluster A serves as the $Train - Validation$ set while population cluster B as the $Test$ set. Next, cluster A and B exchange set properties.

4. Calculate performance in terms of prediction and selected features for each protocol.

5. Repeat $3 - 4$ steps in order to create enough statistical samples of performance.

Theoretically, any feature selection that is insusceptible to population structure will yield similar results, within a statistical interval, in terms of prediction performance and selected variables on both population clusters. In any case, accurate, or underestimated performance estimation is of outmost importance, i.e. prediction model and performance do not break down when testing on unseen data.

[10] Magnitude of eigenvalues is equivalent to explained variance of principal components, i.e. $SS_{Ei} = \frac{EigenValue_i}{\sum_{n=1}^{p} EigenValue}$.

Chapter 4

Results

In this section we evaluate *gOMP* performance on simulated datasets and on real human-disease datasets acquired from EGA. Every analysis uses **JAD Bio's™** automated machine learning pipeline. In section 4.1 we present the simulation studies along with the comparison of *gOMP* with *QTCAT*. In section 4.2 the results of SNP discovery on real datasets are presented.

4.1 Simulated Datasets

In section 3.2.1 we described in detail the simulation strategy that will be followed. All genotypic variation across *Arabidopsis thaliana* samples remains identical (SNP values are exactly as genotyped), where only the phenotype affected by a group of SNPs is simulated. The presentation of the results on simulated datasets is organized as follows: In section 4.1.1 we provide an extensive comparison between *gOMP* and *QTCAT*. Comparison metrics include predictive performance, *True Positive Rate*(TPR) against *False Discovery Rate*(FDR), as well as computational time for each method. In section 4.1.2 we evaluate the effect of population structure on *gOMP* experimentally, while trying to quantify selection bias across different populations. Finally, in 4.1.3 we test the null hypothesis of the equivalency between the reported multiple signatures and the reference signature using a suitable test statistic.

4.1.1 gOMP - QTCAT comparison

Quantitative Trait Association Test (*QTCAT*) was considered as the most suitable feature selection method for comparison with *gOMP* since it fulfilled several criteria:

- Dataset availability.
- Open source code, available in GitHub, with reproducible test cases.
- Methodologically correct simulation strategy.

- Insusceptibility to population structure.
- Superiority over regression with *Linear Mixed Models*.
- Easy integration with **JAD Bio's™** pipeline.

As discussed in Simulation scenarios, 3 parameters are required for any scenario; **1)** SNP position distribution (*Gaussian* or *gamma*), **2)** Number of associated SNPs and **3)** heritability, h^2 parameter. Below we will present the 4 different scenarios chosen for *gOMP-QTCAT* comparison.

In order to produce enough paired samples (performances for *gOMP* and *QTCAT*), we repeat each scenario 50 times, i.e. 50 different simulated phenotypes. During any repeat we calculate the R^2 value achieved when regressing the ground truth SNPs on the phenotype. Consequently, we subtract this value from the corresponding R^2 achieved by each feature selection method, resulting in a relative performance metric[1]. By plotting the distribution of these relative performances, models that performed better will lie near the *zero* value of $y-axis$ ($maximum-performance, MxP$ line), while the worst ones will reside around $-h^2$. The value of $-h^2$ is the minimum theoretical performance ($minimum-theoretical-performance, MnTP$ line) that any model can achieve prior to simulation of the phenotype. In practice after the simulation occurs, since h^2 is a statistical parameter, the minimum actual performance ($minimum-actual-performance, MnAP$ line) will vary around this parameter. Models that performed above the MxP line indicate that they identified SNPs which are associated with the random independent noise dictated by h^2 parameter and this is a statistical artifact which should not be taken into account. On the other hand, performance of models lower than the $MnAP$ line translates to identification of SNPs that systematically predict worse than the mean value of phenotype does[2]. When dealing with methods of similar performance, an important aspect is the variance of corresponding distributions, i.e methods that produce models of low variance in performance are always preferred due to their consistency. A final comparison metric is the calculation of the *p-value* of the paired *t-test*, i.e. when testing the null hypothesis of equal performances between *gOMP* and *QTCAT* method.

Regarding solely the identification of ground truth SNPs' performance, we store every reported signature across all repeats and every selection threshold. We remind that for *QTCAT* we used 10 *p-values* logarithmically spaced between 10^{-6} and 0.8, while for *gOMP* the equivalent ΔBIC scores ranging from 31 to 7.13 respectively. Next, we calculated the *true positive rate, (TPR)* and *false discovery rate, (FDR)*; TPR is the amount of correctly identified SNPs to the amount of ground truth SNPs, while FDR is the amount of false positive SNPs divided by

[1] We note that we use the value of the best configuration reported from the machine learning pipeline.

[2] Negative values of R^2 can occur indeed. From the formula: $R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$ this is apparent when predicted value $\hat{y}_i$ is systematically further from $\bar{y}$.

the amount of selected SNPs. A perfect identification of the ground truth SNPs will score $TPR = 1$ and $FDR = 0$, while the reverse is true for the worst identification case. Reasonably, in any feature selection method, an increase in TPR will be accompanied by an increase in FDR, resulting again in a *bias-variance* trade-off equivalent for feature selection. Below we present the performance plots regarding prediction accuracy and ground truth SNPs identification for each scenario.

Scenario 1: Gamma - 20 - 0.7

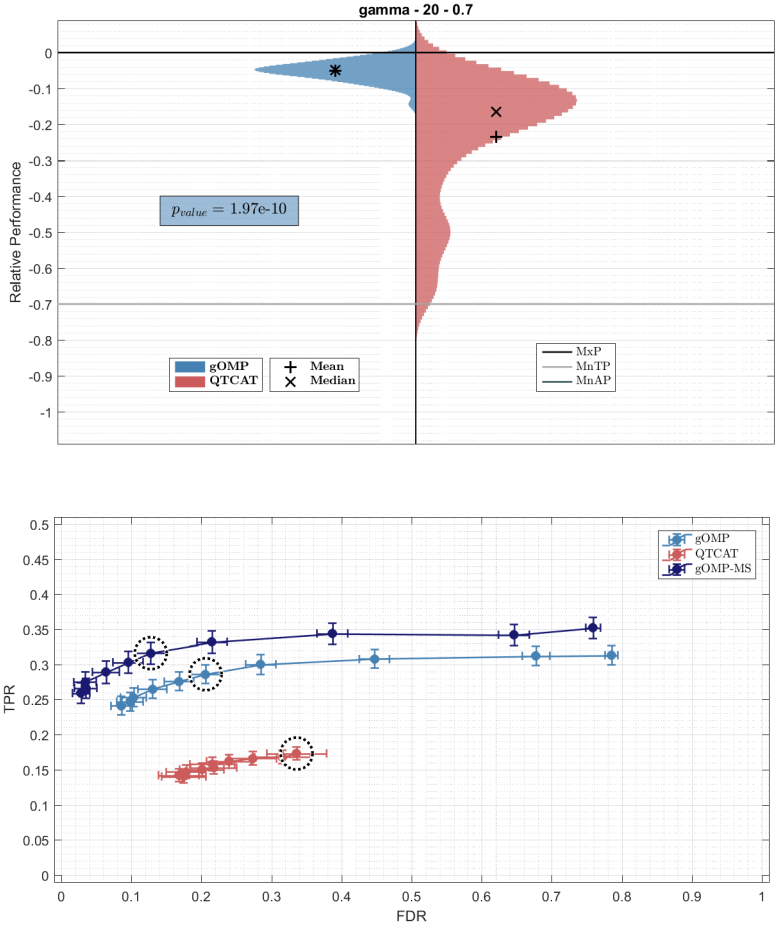
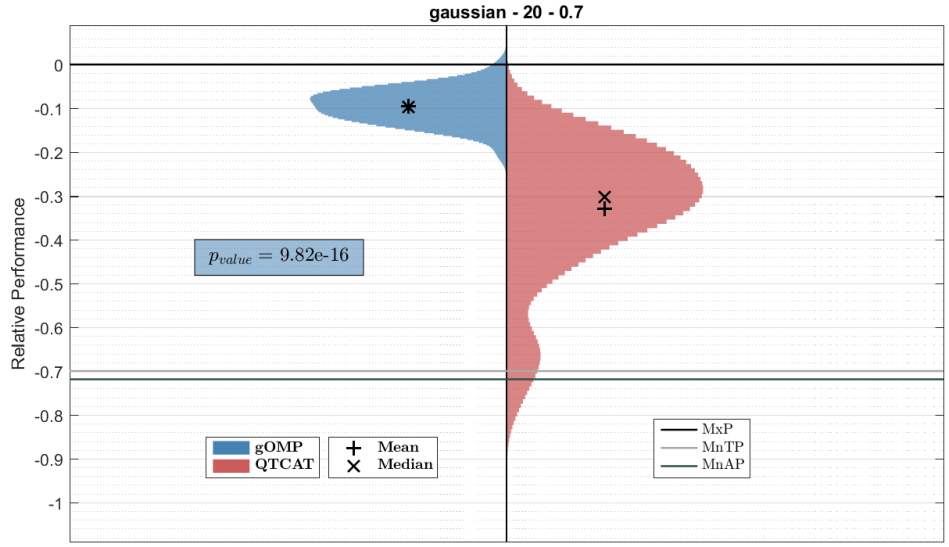
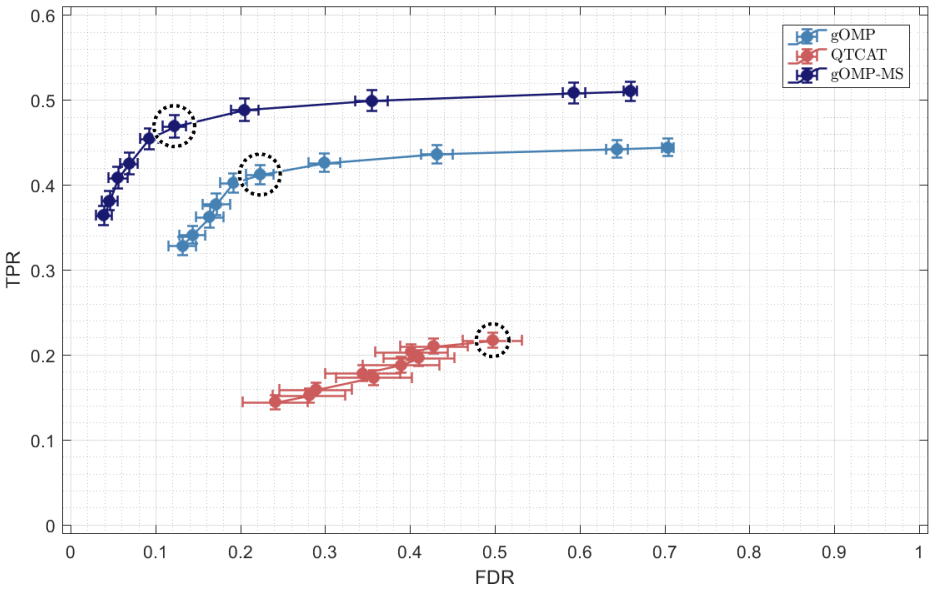


Figure 4.1: **I)** Relative performance distributions across 50 repeats for *gOMP* (light-blue) and *QTCAT* (red) at SCENARIO 1. Highlighted horizontal lines are maximum Performance (MxP line), minimum theoretical performance (MnTP line) and minimum average performance (MnAP line). The *p-value* of the paired *t-test* (null hypothesis of equal mean performances) is contained inside the colored box, while its color corresponds to the feature selection method with the higher average performance. **II)** Average TPR and FDR scores across 10 selection thresholds for *gOMP*, *QTCAT* and *gOMP-MS* (*gOMP* including the equivalent signatures or multiple solutions). The most frequent selection threshold of the best configuration is circled in dotted line, while circle radius is inversely proportional to this frequency.

Scenario 2: Gaussian - 20 - 0.7



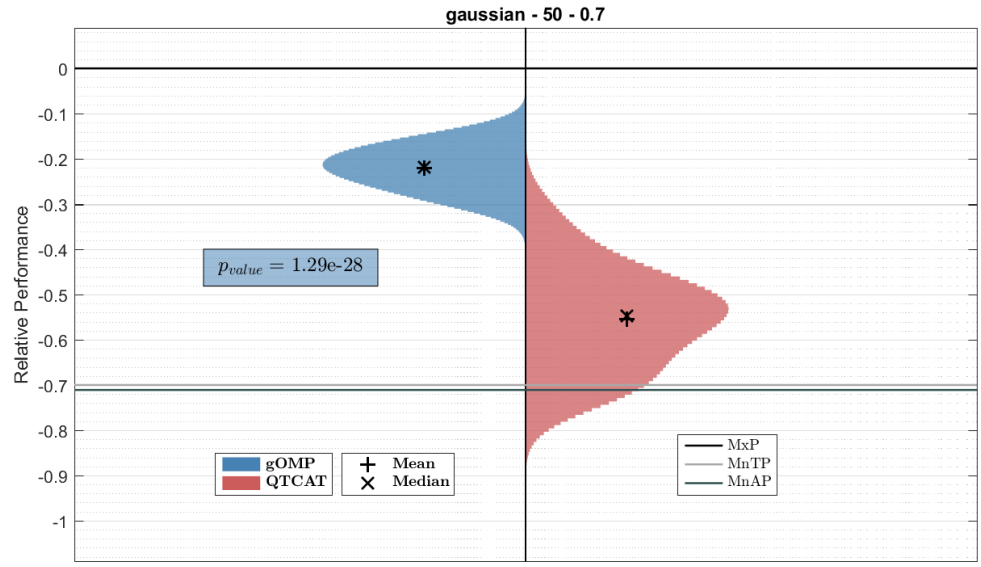
I)



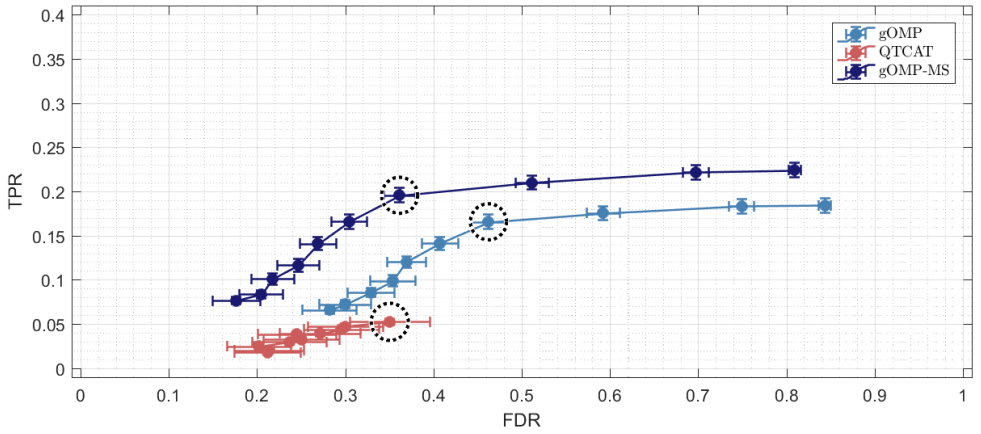
II)

Figure 4.2: **I)** Relative performance distributions across 50 repeats for *gOMP* (light-blue) and *QTCAT* (red) at SCENARIO 2. Highlighted horizontal lines are MxP line, MnTP line and MnAP line. The *p-value* of the paired *t-test* (null hypothesis of equal mean performances) is contained inside the colored box, while its color corresponds to the feature selection method with the higher average performance. **II)** Average TPR and FDR scores across 10 selection thresholds for *gOMP*, *QTCAT* and *gOMP-MS* (*gOMP* including the equivalent signatures or multiple solutions). The most frequent selection threshold of the best configuration is circled in dotted line, while circle radius is inversely proportional to this frequency.

Scenario 3: Gaussian - 50 - 0.7



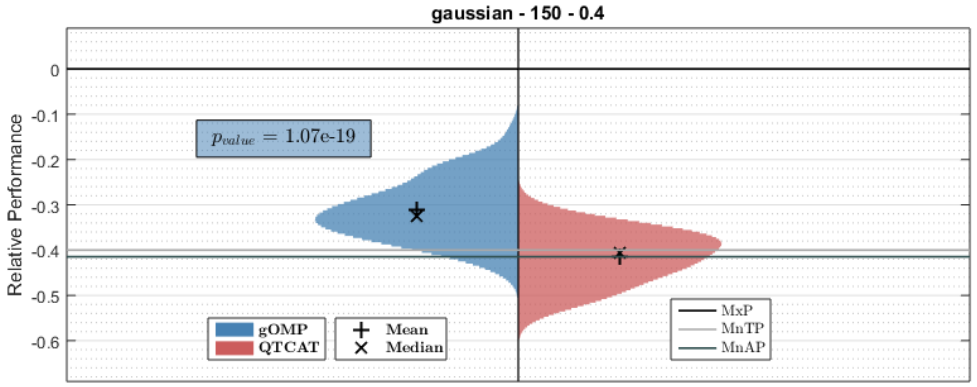
I)



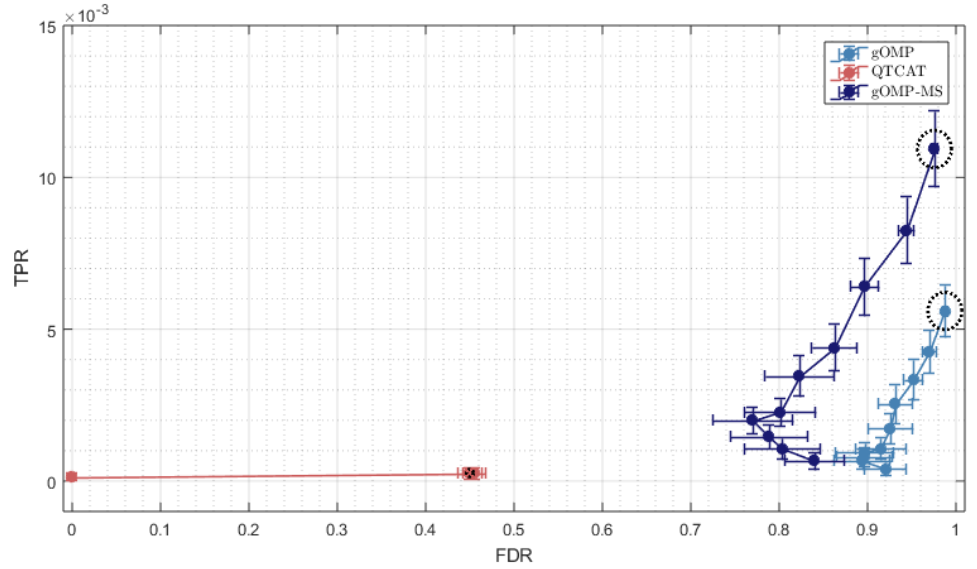
II)

Figure 4.3: **I)** Relative performance distributions across 50 repeats for *gOMP* (light-blue) and *QTCAT* (red) at SCENARIO 3. Highlighted horizontal lines are maximum Performance (MxP line), minimum theoretical performance (MnTP line) and minimum average performance (MnAP line). The *p-value* of the paired *t-test* (null hypothesis of equal performance) is inside the box, while its color corresponds to the feature selection method with the higher mean performance. **II)** Average TPR and FDR scores across 10 selection thresholds for *gOMP*, *QTCAT* and *gOMP-MS* (*gOMP* including the multiple solutions). The most frequent selection threshold of the best configuration is circled in dotted line, while circle radius is inversely proportional to this frequency.

Scenario 4: Gaussian - 150 - 0.4



I)



II)

Figure 4.4: **I)** Relative performance distributions across 50 repeats for *gOMP* (light-blue) and *QTCAT* (red) at SCENARIO 4. Highlighted horizontal lines are maximum Performance (MxP line), minimum theoretical performance (MnTP line) and minimum average performance (MnAP line). The *p-value* of the paired *t-test* (null hypothesis of equal performance) is inside the box, while its color corresponds to the feature selection method with the higher mean performance. **II)** Average TPR and FDR scores across 10 selection thresholds for *gOMP*, *QTCAT* and *gOMP-MS* (*gOMP* including the multiple solutions). The most frequent selection threshold of the best configuration is circled in dotted line, while circle radius is inversely proportional to this frequency.

Regarding predictive performance, *gOMP's* reference signature produces models that are statistically significantly more accurate than the corresponding models

of *QTCAT*, across all 4 simulation scenarios. More importantly, *gOMP's* distribution of predictive performance, acquired from these 50 repeats, is of smaller variance (distributions with tighter bounds), which is an indicative characteristic of a consistent model-producing methodology.

With respect to ground-truth signature retrieval, *gOMP* always detects more true positive features (higher TPR), regardless of the 10 different selection threshold values. This is the case for all 4 scenarios, deducing that *gOMP's* selection strategy is more efficient. As for falsely identified SNPs (FDR), in Scenario 1 ($gamma - 20 - 0.7$) *gOMP* achieves lower FDR for the most qualified (most frequent hyper-parameter value used in best model) selection threshold. The above is also true for Scenario 2 ($gaussian - 20 - 0.7$). However, in Scenario 3 ($gaussian - 50 - 0.7$), most frequent selection threshold produces higher FDR values, for *gOMP* compared to *QTCAT*. Finally, concerning Scenario 4 ($gaussian - 150 - 0.4$), both methods perform poorly, probably due to highly noisy data influenced by low heritability.

Computational Time

Another aspect of a feature selection algorithm's efficiency, apart from predictive performance and associated features' retrieval, is computational time. In computer science, *big O* notation ($O \sim$), or time complexity, is used to characterize algorithms according to their computational time, or memory requirements as a function of input parameters, p. In feature selection, the parameters p usually consist of sample size N, dimensionality size (number of features) F and number of selected features S. Generally, a linear $O(p)$ is ideal, i.e. $O(kN)$, $O(kF)$ or $O(kS)$, whereas quadratic, or higher, $O(p)$ indicates a rather computationally inefficient algorithm.

For the purposes of this thesis, we examined time complexity of *gOMP* and *QTCAT* in terms of sample and dimensionality size ($O \sim f(N, F)$) on a predefined simulation dataset with the following simulation parameters: distribution $= gamma$, number of SNPs $= 20$ and heritability $= 0.7$. Since the simulation occurs only for the phenotype, the maximum sample and dimensionality size are initially limited to the corresponding size of the original dataset, that is $1,307$ samples and $214,051$ features. Nonetheless, *QTCAT* requires a substantial amount of computational time to complete when feature size is maximum, approximately $6h$ for $1,307 \times 214,051$ dataset. For this reason we constrained feature size to a range of 10% to 60%, while sample size to a range of 10% to 100%.

To conclude, we have to note that *QTCAT* is implemented in **R**™, while *gOMP* in **Matlab R2016b**™, thus a comparison in absolute differences in computational time is not of interest, rather than the degree and coefficients of each polynomial function of time complexity ($O(p)$). All simulations and runs were performed in a machine with $4.2GHz$ processors (Inter(R)™ i7-7700) and $32GB$ of RAM, running on a $64bit$ Windows 10™ operating system.

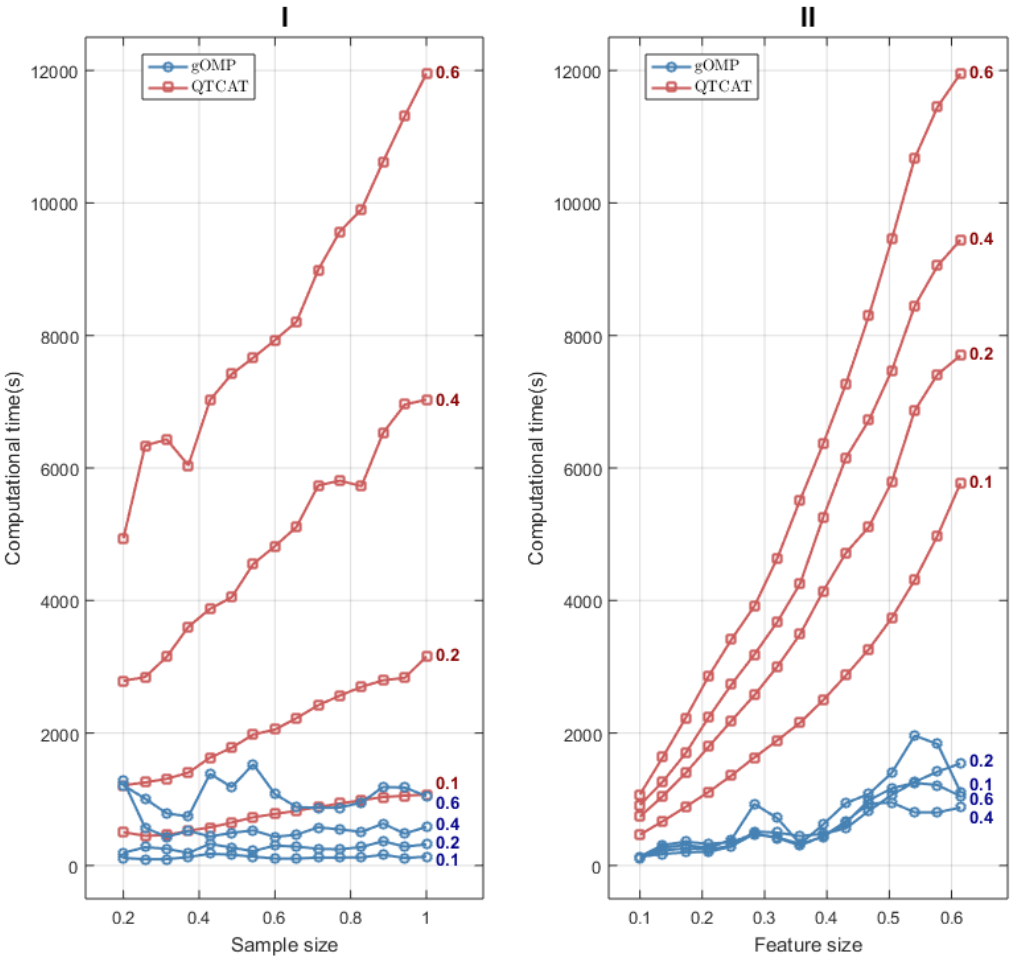


Figure 4.5: Computational time comparison between *QTCAT* and *gOMP*. Figure **I** depicts computational time, for each feature selection method, as a function of relative sample size (100% corresponds to 1,307 samples) for 4 different relative feature sizes (100% corresponds to 214,051 SNPs). Figure **II** depicts computational time as a function of relative feature size for 4 different relative sample sizes.

In terms of time complexity, *gOMP* is evidently more efficient regardless of sample or feature size. As figure 4.4I suggests, *gOMP's* computational time is invariant of sample size, i.e constant value dependent only on feature size, while *QTCAT's* is linearly dependent, with increasing slope as feature size gets larger. Regarding feature size (figure 4.4II), both methods have linearly dependent computational time, however again, the corresponding slopes are larger for *QTCAT*.

Since *gOMP* and *QTCAT* are implemented in different programming languages

(*MATLAB* and *R* respectively), the absolute time differences between these methods should not be taken into account, rather than the differences between the respective derivatives, which capture the inherent $big - O$ notation of each algorithm.

4.1.2 Effect of population structure

Horton et al. genotyped $1,307$ world-wide accessions of the plant *arabidopsis thaliana* and were able to detect and describe the global pattern of genetic variation for this species. This localized genetic variation, or population structure is apparent and shown below in figure 4.6, where the first 2 principal components are extracted from a *PCA* and each sample is colored based on geographical region.

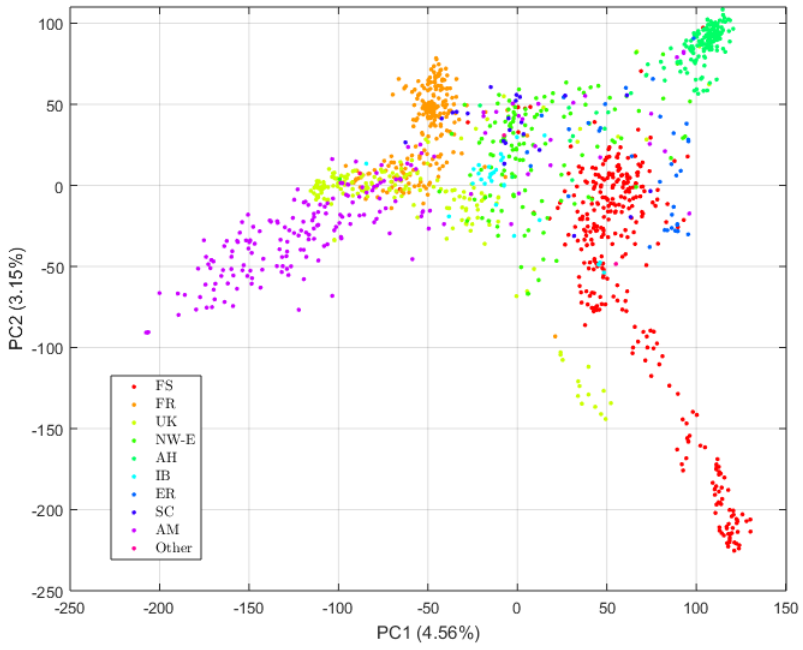


Figure 4.6: Inherent population structure in *arabidopsis thaliana* dataset

In section 3.4 we discussed an empirical solution in determining the effect of population structure generally on feature selection. We remind that we initiate by clustering the populations into 2 major hyper-groups with maximum differentiation between them. The null hypothesis states that, when training a feature selection method on *cluster A*, the performance estimation for this method will be statistically equal compared to the actual performance on unseen, test data, i.e on *cluster B* and vice versa. The above statement follows that population structure plays no role in selecting the true, associative with the phenotype, features. Inarguably, training sample size strongly affects the selection procedure, leading to

FS	324	Fennoscandia (Norway, Sweden & Finland)
FR	222	France
UK	187	British Isles
NW-E	129	North-West Europe (Belgium, Netherlands, Denmark, Germany & Poland)
AH	165	Austria-Hungary (Austria, Czech Republic & Romania)
IB	30	Iberia (Portugal & Spain)
ER	35	Eastern-Range (Estonia, Lithuania, Belarus, Ukraine, Georgia, Azerbaijan, Russia, Tajikistan, Kashmir & Kazakhstan)
SC	22	South-Central Europe (Switzerland & Italy)
AM	190	Americas (Canada & United States)
Other	3	–

Table 4.1: Population characteristics of *arabidopsis thaliana* by geographical region

statistically significant differences in performance estimation and test performance, but this will not pose any problem, if and only if, training procedure systematically underestimates the actual performance on unseen data, i.e. feature selection method identifies its inherent weakness due to low sample size, but over-performs when tested on data of larger sample size (under-promise, over-deliver).

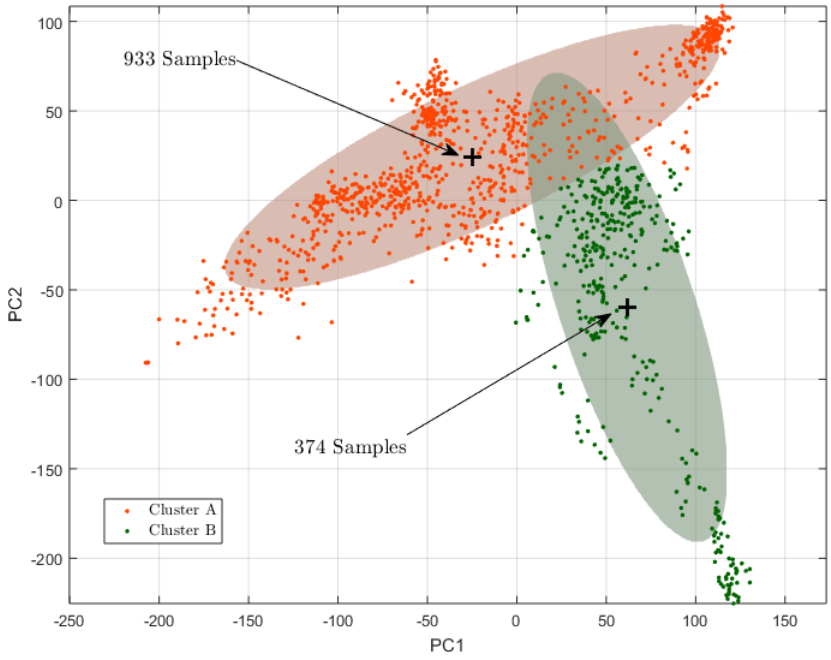


Figure 4.7: Clustering of populations in 2 major hyper-groups

After the clustering is performed, sample sizes for each hyper-group are: *cluster* A= 933 samples and *cluster* B= 374 samples (figure 4.5). For the purposes of this experiment we used the phenotypes produced by ***gamma - 20 - 0.7*** simulation scenario, across 40 repeats (for computational-time reasons). For every repeat we used 2 different training sets, *cluster* A and *cluster* B, while testing was applied on *cluster* B and *cluster* A respectively. For each training procedure we used, again, 10 ΔBIC values and 6 combinations of Random Forest hyper-parameters, resulting in 60 configurations per training set (120 total). On figure 4.6 we present the performance comparison between training set (estimation) and test set (actual) for the 2 *train-test* scenarios.

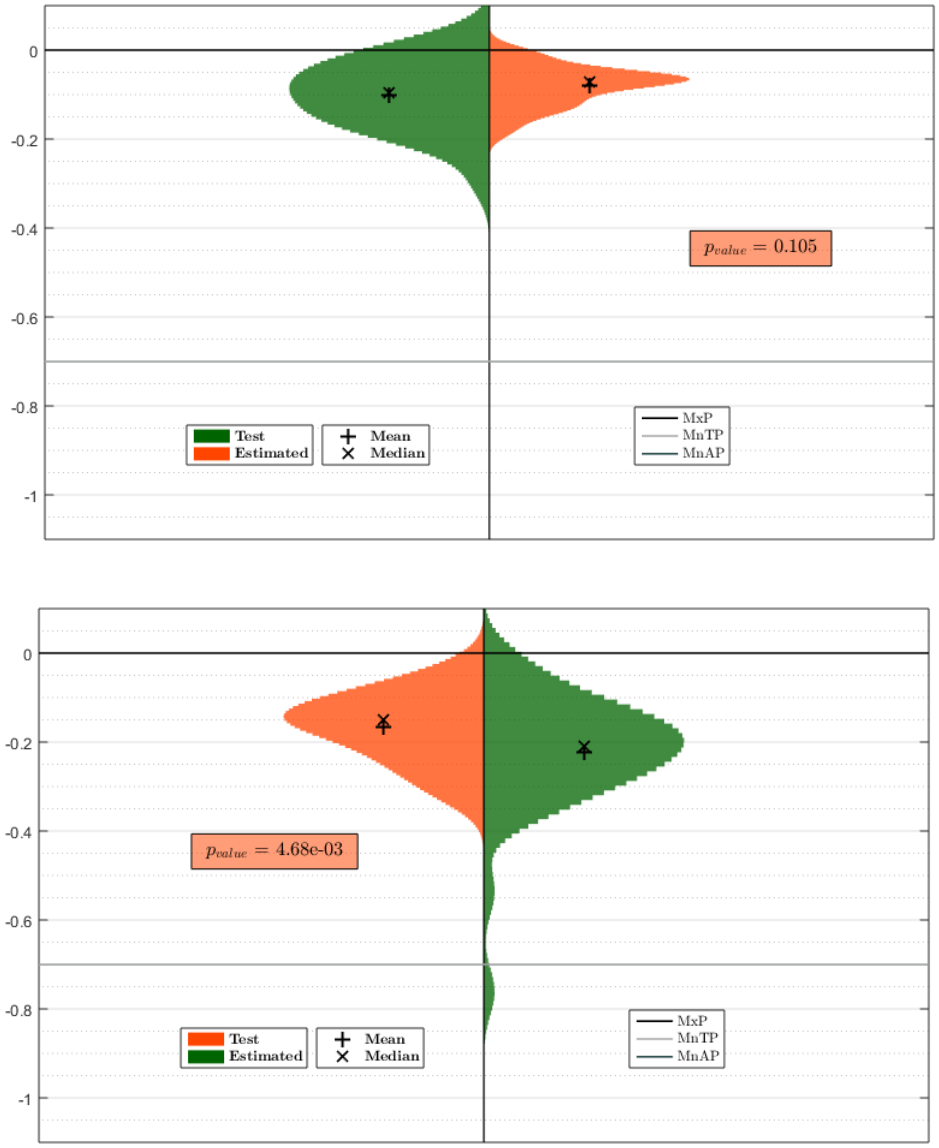


Figure 4.8: Performance distributions across 40 repeats between training set and test set. On the left lies the actual performance on the test set. On the right lies the estimated performance from the training set. Again, inside the box lies the *p-value* of the paired *t-test* of the null hypothesis that mean performances are equal and the background is colored accordingly to the distribution with the larger average performance. **I)** Training set: *Cluster A* (933 samples), Test set: *Cluster B* (374 samples). Estimated performance is slightly higher but with statistically insignificant difference. **II)** Training set: *Cluster B* (374 samples), Test set: *Cluster A* (933 samples). Estimated performance is lower with statistically significant difference, i.e training procedure systematically under-estimates the actual performance.

In figure 4.7I we notice that when the pipeline is trained with the larger hyper-population (933 samples), performance estimation is not statistically significantly different when testing the best reported model on a hyper-population with the higher possible genotypic variation. This evidences that the produced model is unaffected by population structure. On the other hand, when the pipeline is trained with the smaller hyper-population (374 samples, figure 4.7II), estimated performance is statistically significantly lower than the actual one, meaning that the pipeline systematically underestimates the performance on unseen data. This behaviour is attributed mostly on the low sample size of the second hyper-population set. Nevertheless, underestimation of performance is preferable over an optimistic estimation, in the context of prognostic expectations.

4.2 Real Datasets

As discussed in section 3.2.2, we analyzed 9 human-disease datasets, 8 acquired from *EGA* and the last from a *Dialogue for Reverse Engineering Assessments and Methods* (DREAM) challenge. These high-dimensional datasets fall into the category of "small n - large p" problems, where sample size is in the order of thousands, while feature space in the order of hundreds of thousands. Below an overview of the finalized datasets is presented, while table 4.3 contains a summary of the corresponding results. In figures 4.9 - 4.18 the performance evaluation across all 3 repeats and 2 splits is depicted.

Dataset	***sample size***	***feature size***	***P(T=case)***
Ulcerative Colitis	8,788	884,760	32.1%
Parkinson's	7,791	571,685	27.8%
Multiple Sclerosis	16,969	572,359	62.7%
Psoriasis	8,216	571,711	31,1%
Ankylosing Spondylitis	7,604	571,469	26.1%
Schizophrenia	9,017	884,755	21.3%
Pharmacogenomic Response to Statins	10,073	883,851	40.8%
Barrett's Oesophagus	7,602	570,638	26.1%
Reumatoid Arthritis	2,022	1,866,172	21.5%

Table 4.2: Real datasets overview

Dataset	*Signature size*	*# Signatures*	*Performance*	*rs Codes*
Ulcerative Colitis	25.333	159.333	0.733	Show
Parkinson's	1	152.5	0.998	Show
Multiple Sclerosis	3	1.333	0.952	Show
Psoriasis	1	118	1	Show
Ankylosing Spondylitis	1	125.666	0.999	Show
Schizophrenia	6.833	4.5	0.954	Show
Pharmacogenomic Response to Statins	1	2.5	0.999	Show
Barretts Oesophagus	1	123	1	Show
Reumatoid Arthritis	10.333	10,3907.667	0.506	Show

Table 4.3: Averaged Results on real data. rs Codes refer to the union of selected and equivalent SNPs across all repeats.

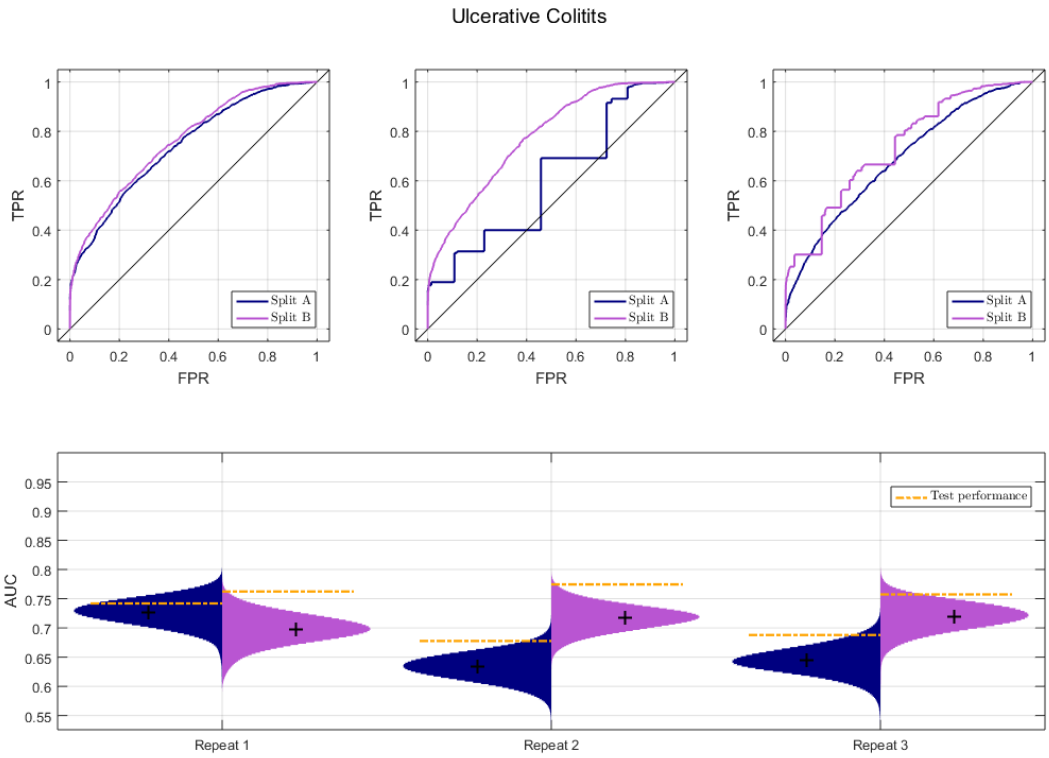


Figure 4.9: Performance on Ulcerative Colitis. First row of figures depicts the receiving operating characteristic (ROC) curves for each split (blue and magenta curves), across all repeats. Second row figure illustrates the estimated performance distributions as calculated by the bootstrap bias correction (BBC) method, while the yellow line concerns the actual performance on the test set.

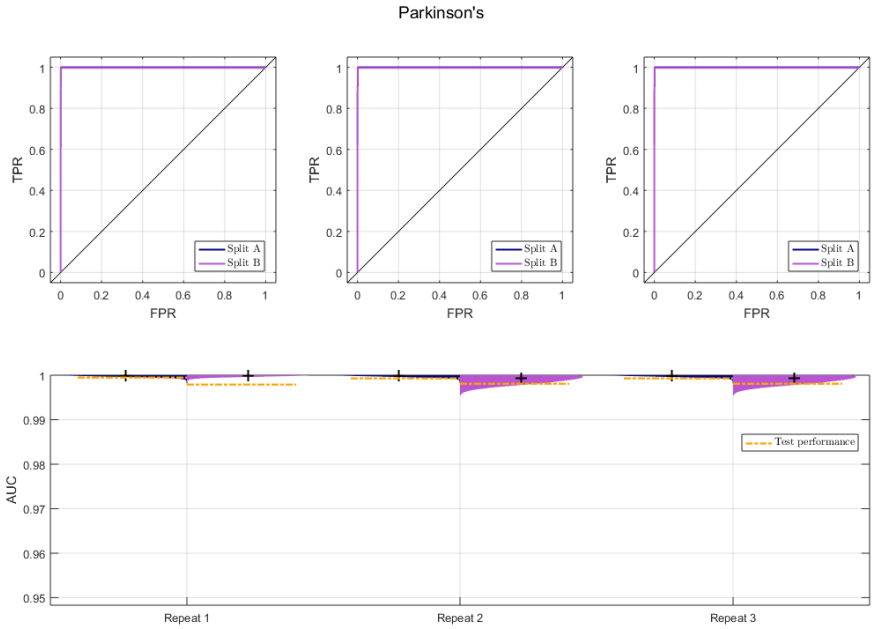


Figure 4.10: Performance on Parkinson's disease

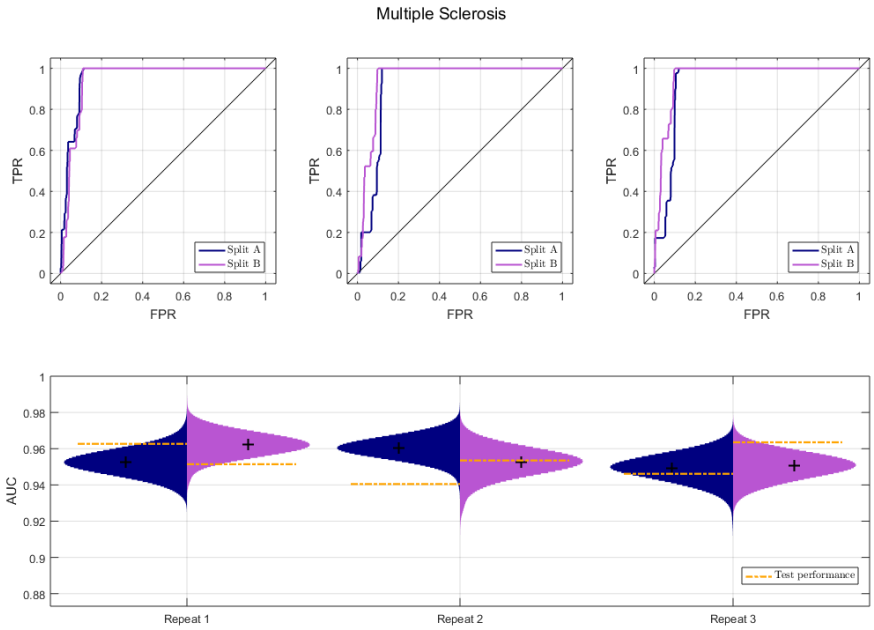


Figure 4.11: Performance on Multiple Sclerosis

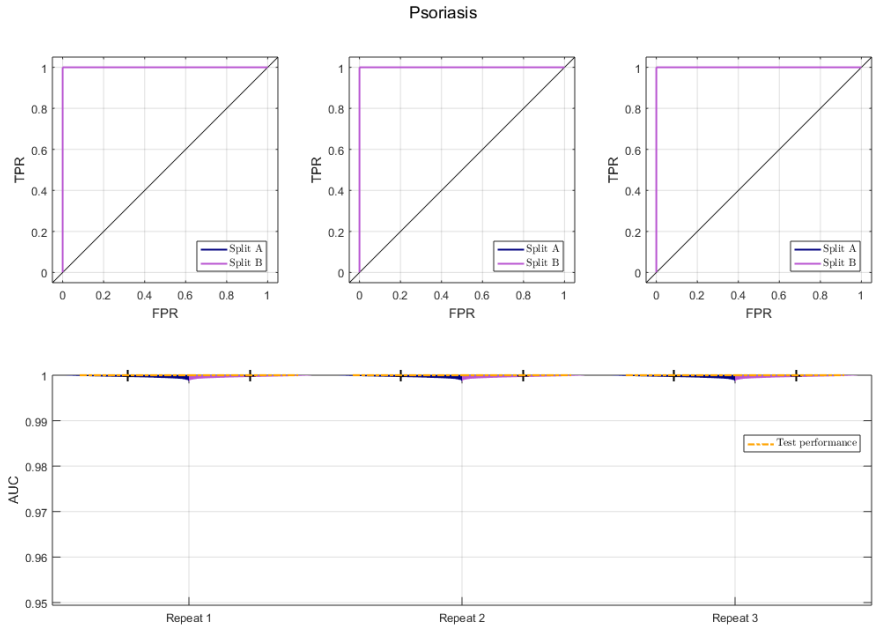


Figure 4.12: Performance on Psoriasis

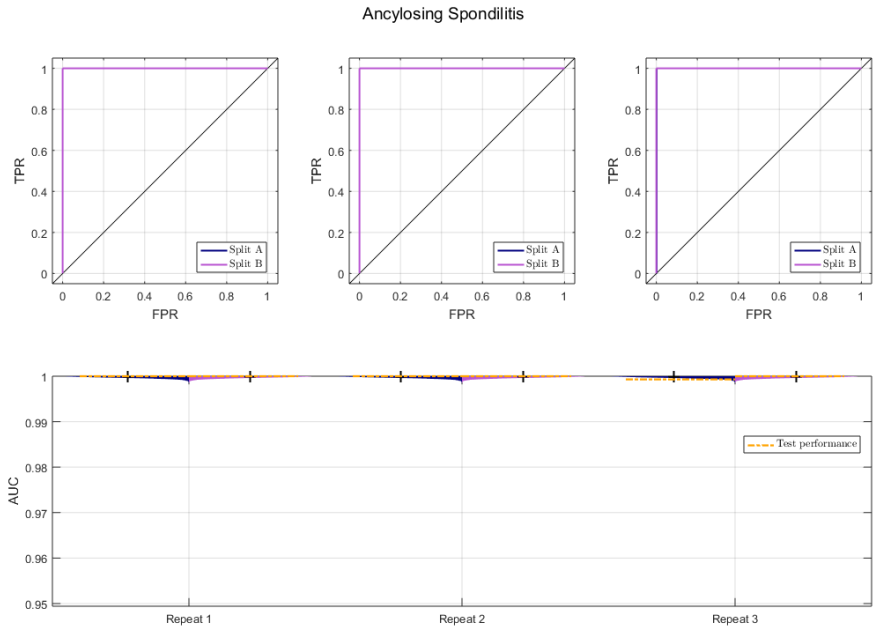


Figure 4.13: Performance on Ankylosing Spondilitis

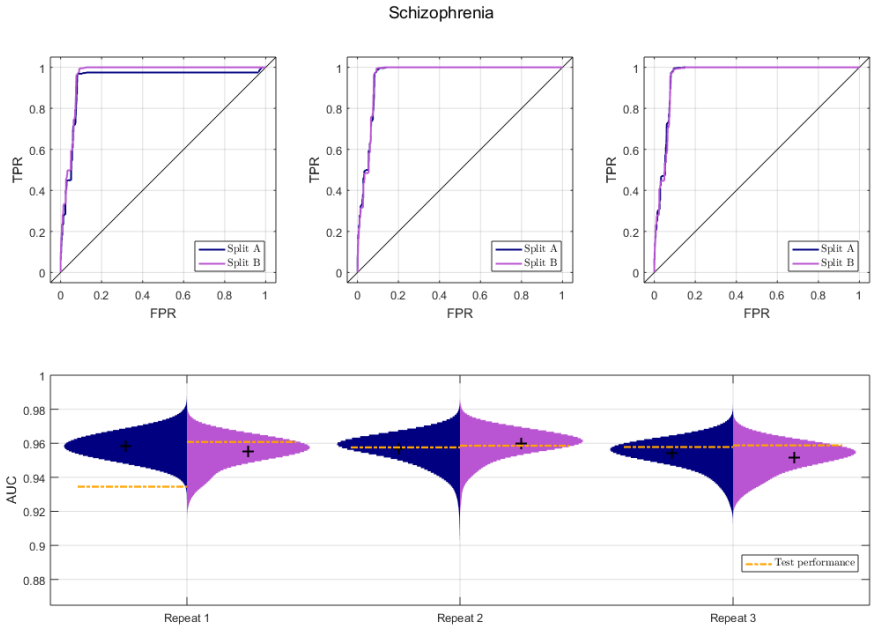


Figure 4.14: Performance on Schizophrenia

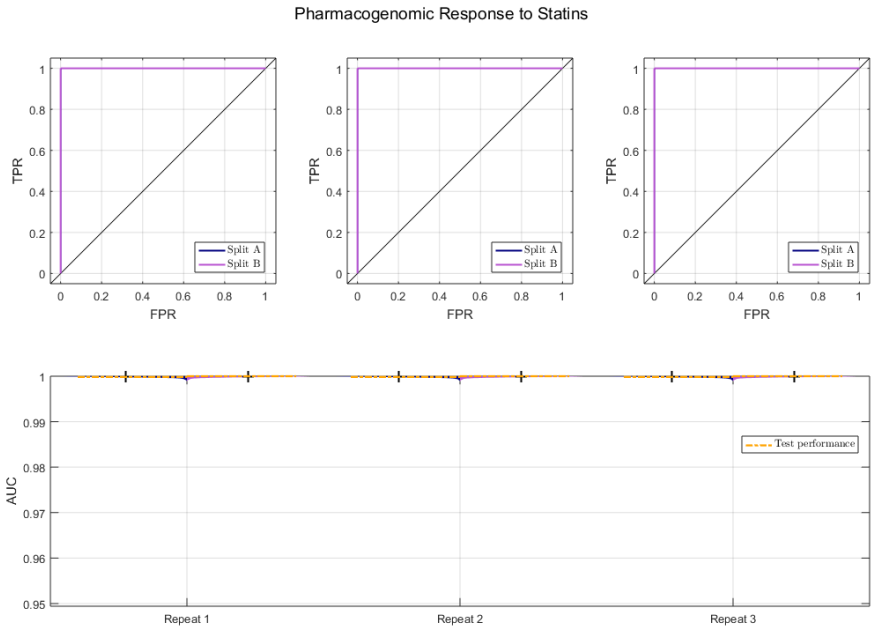


Figure 4.15: Performance on Pharmacogenomic Response to Statins

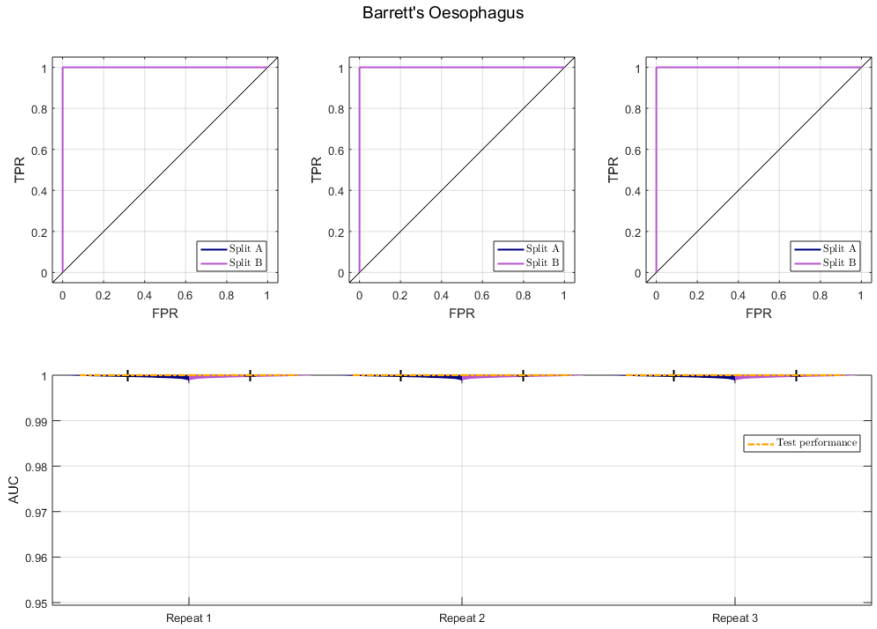


Figure 4.16: Performance on Barretts Oesophagus disease

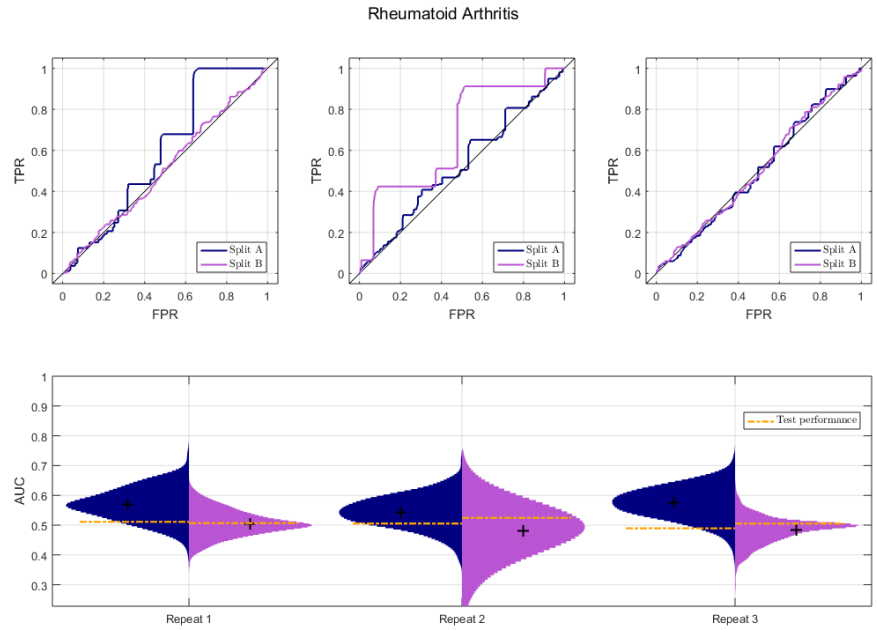


Figure 4.17: Performance on Reumatoid Arthritis disease

Excluding analyses for Ulcerative Colitis and especially Reumatoid arthritis, where the results were inconclusive, i.e. low area under the curve, close to random guessing, in all remaining datasets performance was exceptionally high. Particularly for datasets such as *Parkinson's*, *Psoriasis*, *Ancylosing Spondilitis*, *Barretts Oesophagus* and *Pharmacogenomic Response to Statins* the pipeline identified a unique SNP, with many equivalent SNPs that is highly associated with each corresponding phenotype. Especially for *Psoriasis* and *Barretts Oesophagus* the relationship between the causative SNPs and the phenotype is deterministic, meaning that a single SNP is sufficient for a 100% accurate prediction of disease status.

Finally, we have to note that a high percentage of identified loci concerns genomic areas of introns. Although introns are not expressed, i.e. do not affect directly the encoded aminoacid, they play an important role in alternative splicing, thus affecting the final form of the produced protein. Below we present the results specifically for Multiple Sclerosis as reported from Ensemble's Variant Effect Predictor tool.

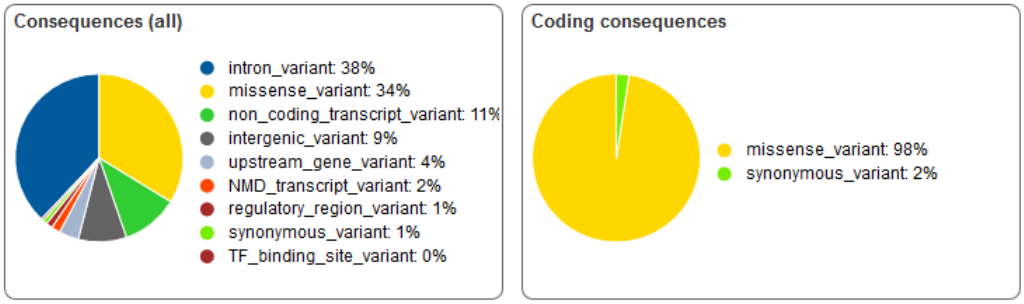


Figure 4.18: Ensemble's Variant Effect Predictor tool for Multiple Sclerosis SNPs. Intron variants along with non-coding variants make up to nearly 50% of reported SNPs (6 out of 12), while average performance of these 12 SNPs approaches 95% AUC.

Chapter 5

Discussion

In this chapter we will comment on the results in the same order as presented at the corresponding chapter. Furthermore, we will suggest future work regarding *gOMP's* features and usability.

5.1 gOMP - QTCAT comparison

In section 4.1.1 we described the comparison strategy and characteristics for each simulation scenario.

Regarding predictive performance, *gOMP* outperforms *QTCAT* in all simulation scenarios, in a remarkably consistent way (proven by corresponding performance distributions). Nevertheless, predictive performance by itself cannot suffice when comparing feature selection methods, since the number of true positive and false positive features identified by each method are metrics of great importance, especially for knowledge discovery. For example, a feature selection method can achieve high performance by selecting many irrelevant features, i.e. features that are randomly associated with the phenotype, thus resulting in improvement of final model's predictive performance by selecting many such features. In our work, we evaluated *gOMP* and *QTCAT* in terms of True Positive Rate and False Discovery Rate, to demonstrate the importance of this trade-off. Again, *gOMP* selected systematically more truly associated SNPs than *QTCAT*, while keeping *FDR* to more than acceptable levels. The *TPR* and *FDR* values improve further when taking into account the corresponding equivalent features of the reference signature, pointing out the importance of multiple solutions.

In terms of time complexity, *gOMP* proves to be far more efficient than *QTCAT*, owing much of its superiority to the residual-based selection strategy. Arguably, *QTCAT* allocates high computational load to the initial hierarchical clustering for all SNPs, while *gOMP* identifies correlated SNPs only for the selected ones, forgoing unnecessary operations regarding not associated SNPs. Although one can argue that the initial clustering should be carried out only once for a given genomic dataset, this would lead to a methodologically incorrect and biased data analysis,

i.e. performing cross-validation with a filtering (produced hierarchical tree) based on all the available samples. Finally, we have to note that *QTCAT* uses an internal $10-fold$ cross validation in order to tune the λ regularization parameter for LASSO selection procedure, thus burdened by additional operations with lower sample size.

Finally, in this thesis we argue that *gOMP* with multiple solutions accounts for population structure, thus correcting automatically in presence of such phenomena. In order to provide evidence that this correction is accomplished indeed, we provide an empirical evaluation presented in section 4.1.2. In any case, this experimental study cannot substitute a mathematical formulation which proves that this procedure accomplishes correction of population structure, conclusively. For the time being, we rely on this empirical evaluation along with the superiority of *gOMP* compared to *QTCAT*, a method that accounts for population structure.

5.2 Evaluation on Real Datasets

In the majority of the analyses on real data, predictive performance of identified genomic signatures was exceptionally high. In 5 datasets (Parkinson's, Psoriasis, Ankylosing Spondylitis, Pharmacogenomic response to statins and Barretts Oesophagus) the association problem was rather easy, since only one causative SNP, along with its equivalent SNPs, was able to achieve a nearly perfect prediction of the disease status. *gOMP* proved to be a powerful tool even when multiple SNPs were associated to the phenotype (multivariate problem), thus being able to capture additive effects, e.g. for Scizophrenia and Multiple Sclerosis. Specifically for Multiple Sclerosis, despite of high performance, SNP identification was inconsistent across the 3 repeats (non-overlapping signatures), something that requires further investigation.

5.3 Future Work

gOMP is able to accept a variety of outcome variables' types, so further, extensive tests on such types, e.g. multi-class, time-to-event, etc, should be considered to verify that this flexibility is accompanied by equal performance. On top of that, this method is able to detect solely linear relationships between features and outcome, rendering the adaptation of a non-linear feature selection method an important improvement, especially towards the identification of epistatic phenomena.

Following the work of Tsamardinos et al. (2018a) with respect to algorithmic implementation, partitioning of samples and implementing suitable statistical tests for such a task will contribute greatly to the direction of big data handling.

Nishizaki and Boyle (2017) argued about the importance of non-coding SNPs and the adaptation of suitable analyses pipelines in genetic discovery and translational research. In our work, we identified many non-coding SNPs, mainly as introns, that were highly predictive of the phenotype, reinforcing this paper's arguments.

Chapter 6

Supplementary

SNP IDs						
rs10018315	rs10215427	rs10491763	rs1051336	rs10517083	rs1060688	rs10807889
rs10953555	rs11009599	rs11097691	rs11190126	rs11190134	rs11260562	rs1138536
rs11709195	rs11765537	rs11967280	rs12646827	rs12673441	rs13117744	rs13133567
rs13226064	rs133164	rs133204	rs133592	rs1385961	rs1468433	rs1658691
rs16842766	rs16861974	rs16870123	rs16870851	rs16870857	rs16870858	rs16870863
rs16870875	rs16870895	rs16870905	rs16870966	rs16889981	rs16892581	rs16920014
rs16979826	rs17012387	rs17017386	rs17057254	rs17061987	rs17075395	rs17082268
rs17083068	rs17089258	rs17097611	rs17097633	rs17102066	rs17120254	rs17134725
rs17171025	rs17272355	rs17381247	rs17440213	rs17495612	rs17657688	rs17749281
rs1939769	rs2140211	rs2159432	rs2227617	rs2290253	rs2348362	rs2395932
rs2526747	rs2564005	rs2919427	rs295146	rs2999547	rs3024493	rs3024505
rs3135393	rs402914	rs41336649	rs41452148	rs41471147	rs41494046	rs424288
rs4317445	rs4365962	rs4386497	rs4535723	rs4730281	rs4923993	rs6034816
rs6056957	rs615672	rs6670226	rs6677211	rs6786678	rs6828390	rs6851158
rs6940296	rs6949033	rs6974243	rs6987509	rs7061	rs7078219	rs7081330
rs7155496	rs729525	rs7404688	rs7507552	rs7532909	rs7626113	rs7710409
rs7722786	rs7782999	rs7841460	rs7894394	rs7905537	rs8035453	rs881176
rs886774	rs9268852	rs9283487	rs9284246	rs9311374	rs9352947	rs9449595
rs9547956	rs9838138	rs9862002	rs9912428			

Table 6.1: Associated SNPs for Ulcerative Colitis

SNP IDs						
rs1001088	rs10052085	rs10067215	rs10072441	rs10132866	rs10166677	rs10241910
rs10249429	rs10264891	rs10432082	rs10433915	rs10873568	rs11044149	rs11061815
rs11192811	rs11502439	rs11591957	rs11833256	rs11838383	rs11910961	rs11981623
rs11992325	rs12019098	rs12116471	rs12156609	rs12372975	rs12401332	rs12461842
rs12498442	rs12516495	rs12579602	rs12605120	rs12620439	rs12643506	rs12766409
rs12775038	rs13017214	rs13053037	rs13153243	rs13230063	rs13294973	rs13437824
rs1354287	rs1596912	rs1672256	rs16886662	rs16935498	rs16940368	rs1704794
rs17113769	rs17168416	rs1720699	rs17595460	rs1793967	rs1807366	rs1832086
rs1905220	rs1906664	rs2110347	rs2148613	rs215109	rs2212048	rs2259315
rs2260188	rs2338241	rs2368805	rs2370866	rs2448507	rs2456599	rs2488217
rs2537367	rs2539177	rs2549598	rs2568287	rs2580567	rs2598648	rs2659792
rs2668863	rs2669384	rs2684146	rs2699830	rs2704135	rs2719476	rs2729227
rs2734415	rs2740001	rs2743476	rs2746702	rs2754683	rs28384110	rs28404871
rs28418883	rs28441383	rs28445367	rs28446843	rs2848213	rs28495424	rs28551470
rs28575733	rs28674911	rs2876845	rs28881575	rs28971843	rs3019831	rs3130676
rs34239705	rs34912894	rs35158069	rs35224109	rs36012344	rs368374	rs3798356
rs3815277	rs3821231	rs385032	rs385248	rs3894703	rs3897248	rs3926475
rs3984089	rs4006953	rs4017124	rs4023483	rs408012	rs4082490	rs4086410
rs4115951	rs4130395	rs4132083	rs4253288	rs4259063	rs4426661	rs456323
rs4636313	rs4756249	rs4876154	rs4978281	rs4983352	rs4983501	rs4988718
rs5758601	rs6053078	rs6061391	rs6414619	rs6531483	rs6709346	rs673211
rs6746716	rs678808	rs6912226	rs6939702	rs6944182	rs6953310	rs7031665
rs7080322	rs7183701	rs7339834	rs7350758	rs7368154	rs7429583	rs7430494
rs7560407	rs7591046	rs7594898	rs7610590	rs7791652	rs7950983	rs7962701
rs7975201	rs8189691	rs9435653	rs9446038	rs9580861	rs9624081	rs9665272
rs9675147	rs9682544	rs9703854	rs9709395	rs9714780	rs9797770	rs9941959
rs9995751						

Table 6.2: Associated SNPs for Parkinson's

SNP IDs						
rs11976716	rs17148284	rs2259315	rs2667265	rs2774141	rs2897356	rs3129860
rs3129941	rs3815675	rs910049	rs9271366	rs9580861		

Table 6.3: Associated SNPs for Multiple Sclerosis

SNP IDs						
rs10147488	rs10228176	rs10240798	rs10255009	rs10470871	rs10471846	rs10767181
rs10805379	rs11061815	rs11254491	rs11645356	rs11776503	rs11838207	rs11838383
rs11899396	rs11903746	rs11957319	rs12259548	rs12297855	rs12337345	rs12449903
rs12500655	rs12594787	rs12731588	rs12782401	rs12919630	rs12930400	rs13100804
rs13230063	rs13274025	rs13294973	rs152833	rs1610833	rs167442	rs17017303
rs17168416	rs1832086	rs1842121	rs1874595	rs2000096	rs2135088	rs2148613
rs2212048	rs2217766	rs2232060	rs2259315	rs2304890	rs2309232	rs2314169
rs2330200	rs2370866	rs2387513	rs2532400	rs2598648	rs2628811	rs2754683
rs2775733	rs2840054	rs28404871	rs28418883	rs2843247	rs2848213	rs28491656
rs28495424	rs28535255	rs28537431	rs2855977	rs28647952	rs2865600	rs28690750
rs2869237	rs2873745	rs28797870	rs28873202	rs28881575	rs288895	rs28893518
rs28972322	rs2967102	rs34475990	rs34902207	rs34934808	rs35302909	rs35303150
rs35885418	rs3815277	rs3821231	rs3865067	rs3865746	rs3874616	rs3894999
rs3971507	rs3977397	rs3999299	rs4081622	rs4098611	rs4112941	rs4236561
rs4291961	rs4437523	rs446603	rs4523129	rs4593786	rs4726690	rs4866292
rs4988718	rs513556	rs6115614	rs673132	rs6853606	rs6868914	rs694997
rs7156777	rs7227430	rs7260374	rs7303888	rs7429583	rs7433242	rs7560407
rs7950983	rs7975201	rs891491	rs9290539	rs9435653	rs957878	rs9647131
rs9682544	rs9706464	rs9714780	rs9797770			

Table 6.4: Associated SNPs for Psoriasis

SNP IDs						
pgxUn0002	rs10132866	rs10145523	rs10228176	rs10229299	rs10241910	rs10414839
rs10433277	rs10433315	rs10434278	rs10471846	rs1059873	rs10770692	rs10805379
rs11061815	rs11072343	rs11248188	rs11528840	rs11534657	rs11731634	rs11790913
rs11849093	rs11903746	rs12069907	rs12091111	rs12297855	rs12374251	rs12461842
rs12517245	rs12594787	rs12631243	rs12775038	rs12819206	rs12919630	rs13017214
rs13100804	rs13294973	rs1553451	rs1672256	rs167442	rs16879725	rs17017303
rs1704794	rs1842121	rs1964498	rs2005023	rs2135088	rs215109	rs2202040
rs220566	rs2212048	rs2259315	rs2260188	rs2265305	rs2314169	rs2370866
rs2456599	rs2457519	rs2486586	rs2530222	rs2539177	rs2568287	rs2580472
rs2580567	rs2632184	rs2667870	rs2729227	rs2753530	rs2774477	rs2810989
rs28364249	rs28441383	rs2848213	rs28495424	rs28551470	rs28587062	rs28674911
rs2869237	rs28770169	rs28797894	rs28893518	rs28972402	rs2925344	rs2960979
rs2964850	rs3019743	rs3116439	rs34239705	rs34370305	rs34759622	rs34902207
rs34912894	rs34934808	rs34957779	rs35010833	rs35427885	rs357706	rs35885418
rs35885593	rs3865741	rs3894997	rs3894999	rs3977397	rs4015127	rs4062120
rs4067651	rs4091033	rs4100614	rs4147617	rs4148460	rs4149320	rs4291961
rs4505257	rs4593786	rs4866292	rs4988718	rs5751678	rs6001168	rs6061391
rs6472976	rs6853606	rs6944182	rs6966163	rs7031665	rs7071703	rs7212038
rs7429583	rs7430494	rs7433242	rs7461038	rs7540001	rs7594898	rs7607534
rs7698124	rs7736236	rs7975201	rs8051706	rs957878	rs9580861	rs9645104
rs9647131	rs9679574	rs9706464	rs9714780	rs9790519	rs9865715	

Table 6.5: Associated SNPs for Ankylosing Spondylitis

SNP IDs						
rs10134877	rs10200882	rs11851128	rs11899533	rs150914	rs1536688	rs16867128
rs16987153	rs17006636	rs17022585	rs17043520	rs17120254	rs1859790	rs2271041
rs35879674	rs4688732	rs5027299	rs6851158	rs7645943	rs7737972	rs9327934

Table 6.6: Associated SNPs for Schizophrenia

SNP IDs		
rs16867128	rs5002300	rs6851158

Table 6.7: Associated SNPs for Pharmacogenomic Response to Statins

SNP IDs						
rs10067215	rs10076207	rs10240798	rs10241910	rs10433277	rs10470871	rs10781048
rs10789525	rs10936714	rs11044149	rs11061815	rs11128495	rs11134253	rs1142089
rs11490504	rs11614472	rs1176728	rs11903746	rs11903965	rs11910961	rs12104394
rs12240398	rs12263659	rs12346806	rs12500655	rs12517245	rs12593069	rs12625483
rs12674455	rs12761343	rs12774838	rs12782401	rs13100804	rs13230063	rs13294973
rs13306054	rs13388848	rs13397036	rs1484838	rs152833	rs16828211	rs16857472
rs16895084	rs16907584	rs16932534	rs1693963	rs16984404	rs1704794	rs17131473
rs17131513	rs1741053	rs1766096	rs1832086	rs1964498	rs1972986	rs1996503
rs2239671	rs2317807	rs2346809	rs2362618	rs2370866	rs2472595	rs2539177
rs2556002	rs2575741	rs2598648	rs2614422	rs2659792	rs2669979	rs2686971
rs2687152	rs2704134	rs2720777	rs2769116	rs2774477	rs2820132	rs28399468
rs2840054	rs28495424	rs28575733	rs28645720	rs28873202	rs2946650	rs297583
rs3218521	rs34522123	rs34591674	rs34681058	rs34829079	rs34893665	rs34912894
rs35010833	rs35224109	rs35843992	rs35885593	rs3771226	rs3821231	rs3874053
rs3912913	rs4017124	rs4043971	rs4086410	rs4089195	rs4097469	rs4147847
rs4236561	rs4253866	rs4310904	rs4360514	rs4542925	rs4646339	rs4646788
rs4726690	rs4983501	rs5570	rs593120	rs6122060	rs620841	rs6416668
rs6472977	rs6723949	rs7032197	rs7136278	rs7201947	rs7250513	rs7291110
rs7356526	rs7434011	rs7551405	rs7560407	rs7577813	rs7776829	rs7940703
rs7958420	rs8187823	rs8192821	rs8192862	rs9435653	rs9472097	rs9483597
rs9604096	rs9730887	rs9843032	rs9929860	rs9959523		

Table 6.8: Associated SNPs for Barretts Oesophagus

SNP IDs						
rs10134246	rs10134344	rs10135705	rs10141936	rs1015024	rs10179370	rs10184186
rs10223138	rs1037690	rs10434638	rs10434639	rs10472138	rs10472140	rs10472652
rs10472653	rs10472660	rs10508543	rs1060740	rs10760337	rs10805815	rs10865554
rs10940743	rs10940744	rs11159423	rs1145586	rs1154929	rs11625030	rs11704577
rs11745355	rs11847376	rs11847499	rs11850024	rs11852146	rs1186981	rs11964028
rs11965182	rs11965221	rs1202314	rs12283	rs12339663	rs12431425	rs12432206
rs12635804	rs12637983	rs12657778	rs12881175	rs12894007	rs13127257	rs13130999
rs13162517	rs13163024	rs13163337	rs13167044	rs13167620	rs13180575	rs13220759
rs1329038	rs13292783	rs13403932	rs13404368	rs13404447	rs13408023	rs1341697
rs1384173	rs1476077	rs1554328	rs1554454	rs1555363	rs1569333	rs16826180
rs16863913	rs16878701	rs16907813	rs16945290	rs16945332	rs16945351	rs16959865
rs16962801	rs17039107	rs17047634	rs17054165	rs17054866	rs17054922	rs17135297
rs17141517	rs17141543	rs17274597	rs17585174	rs17754673	rs1845829	rs1894962
rs1927364	rs1927367	rs2015035	rs2064003	rs2064005	rs2067044	rs2078142
rs217742	rs2178887	rs2182621	rs2205224	rs2220962	rs2221205	rs2270810
rs2270811	rs2272143	rs2401694	rs2430586	rs2703342	rs2887940	rs3024981
rs3024985	rs3129317	rs3777054	rs3794110	rs3803603	rs3822720	rs3829268
rs4234891	rs4241333	rs4280626	rs4296806	rs4397136	rs4405799	rs4418106
rs4432692	rs4455559	rs4470767	rs4470768	rs4509026	rs4515097	rs4558343
rs4576150	rs4594479	rs4621562	rs4637160	rs4645333	rs4676917	rs4677252
rs4677261	rs4677262	rs4677263	rs4677265	rs4677266	rs4691168	rs4691170
rs4748405	rs4748406	rs4748408	rs4748411	rs4748412	rs4750298	rs4767479
rs4814325	rs4814755	rs4814756	rs4814759	rs4823088	rs4854402	rs4854404
rs4854405	rs4854406	rs4866028	rs4866114	rs4866340	rs4899750	rs4959685
rs4959688	rs5000630	rs5007588	rs5997619	rs6035066	rs6045443	rs6045454
rs6045474	rs6081192	rs6081223	rs6136416	rs620590	rs6420918	rs6514907
rs6548267	rs6548268	rs6574529	rs6574532	rs6727076	rs6751638	rs6764873
rs6775343	rs6808382	rs6844945	rs6877027	rs6894312	rs6898404	rs6986186
rs6996562	rs7068684	rs7075270	rs7142797	rs7149937	rs7152418	rs7153010
rs7336772	rs7580145	rs7612030	rs7613484	rs7619189	rs7701168	rs7723606
rs7728809	rs7901640	rs7912011	rs7937602	rs8006733	rs8010423	rs8017083
rs8017997	rs8018470	rs8021467	rs8022661	rs8060282	rs8076439	rs890895
rs911115	rs9292282	rs9308098	rs9323679	rs941681	rs9459275	rs9518737
rs9582585	rs966163	rs9826357	rs9923273	rs993920	rs9942307	

Table 6.9: Associated SNPs for Reumatoid Arthritis

Bibliography

Aliferis, C. F., Statnikov, A. R., Tsamardinos, I., Mani, S., and Koutsoukos, X. D. (2010). Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part I : Algorithms and Empirical Evaluation. *Journal of Machine Learning Research*, 11:171–234.

Borboudakis, G. (2018). Extending Greedy Feature Selection Algorithms for Multiple Solutions.

Borboudakis, G., Stergiannakos, T., Frysali, M., Klontzas, E., Tsamardinos, I., and Froudakis, G. E. (2017). Chemically intuited, large-scale screening of MOFs by machine learning techniques. *npj Computational Materials*, 3(1):1–6.

Borboudakis, G. and Tsamardinos, I. (2017). Forward-backward selection with early dropping. *arXiv preprint arXiv:1705.10770*.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory - COLT '92*, pages 144–152.

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

Grimm, D. G., Roqueiro, D., Salomé, P. A., Kleeberger, S., Greshake, B., Zhu, W., Liu, C., Lippert, C., Stegle, O., Schölkopf, B., Weigel, D., and Borgwardt, K. M. (2017). easyGWAS: A Cloud-Based Platform for Comparing the Results of Genome-Wide Association Studies. *The Plant Cell*, 29(1):5–19.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Horton, M. W., Hancock, A. M., Huang, Y. S., Toomajian, C., Atwell, S., Auton, A., Muliyati, N. W., Platt, A., Sperone, F. G., Vilhjálmsson, B. J., Nordborg, M., Borevitz, J. O., and Bergelson, J. Genome-wide patterns of genetic variation in worldwide Arabidopsis thaliana accessions from the RegMap panel.

Klasen, J. R., Barbez, E., Meier, L., Meinshausen, N., Bu, P., and Koornneef, M. (2016). A multi-marker association method for genome-wide association studies without the need for population structure correction. (May):1–8.

Lagani, V., Athineou, G., Farcomeni, A., Tsagris, M., and Tsamardinos, I. (2017). Feature Selection with the R Package MXM: Discovering Statistically-Equivalent Feature Subsets. *Journal of Statistical Software*, 80(7).

Nishizaki, S. S. and Boyle, A. P. (2017). Mining the Unknown: Assigning Function to Noncoding Single Nucleotide Polymorphisms. *Trends in Genetics*, 33(1):34–45.

Orfanoudaki, G., Markaki, M., Chatzi, K., Tsamardinos, I., and Economou, A. (2017). MatureP: Prediction of secreted proteins with exclusive information from their mature regions. *Scientific Reports*, 7(1):1–12.

Pantazis, Y., Lagani, V., and Tsamardinos, I. (2017). Enumerating multiple equivalent lasso solutions. *arXiv preprint arXiv:1710.04995*.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *1993 Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pages 40–44. IEEE.

Quang H. Vuong, Society, T. E. (1989). Likelihood Ratio Tests for Model Selection and Non-Nested Hypotheses Author (s): Quang H . Vuong Published by : The Econometric Society Stable URL : http://www.jstor.org/stable/1912557 http://links.jstor.org/page/info/about/policies/terms.jsp . JSTOR ' . *Society*, 57(2):307–333.

Segura, V., Vilhjálmsson, B. J., Platt, A., Korte, A., Seren, Ü., Long, Q., and Nordborg, M. (2012). An efficient multi-locus mixed-model approach for genome-wide association studies in structured populations. *Nature Genetics*, 44(7):825–830.

Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, Prediction, and Search*. MIT press.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Tsagris, M., Papadovasilakis, Z., Lakiotaki, K., and Tsamardinos, I. Efficient feature selection on gene expression data : Which algorithm to use ? 3:1–18.

Tsamardinos, I., Aliferis, C. F., and Statnikov, A. (2003). Time and sample efficient discovery of Markov Blankets and direct causal relations. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 673–678. ACM.

Tsamardinos, I., Borboudakis, G., Katsogridakis, P., Pratikakis, P., and Christophides, V. (2018a). A greedy feature selection algorithm for Big Data of high dimensionality. *Machine Learning*.

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006). The Max-Min Hill-Climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78.

Tsamardinos, I., Greasidou, E., and Borboudakis, G. (2018b). Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation. *Machine Learning*, 107(12):1895–1922.

Tsamardinos, I., Lagani, V., and Pappas, D. (2012). Discovering multiple, equivalent biomarker signatures. In *Proceedings of the 7th conference of the Hellenic Society for Computational Biology & Bioinformatics, Heraklion, Crete, Greece.*

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

Waldmann, P., Mészáros, G., Gredler, B., Fuerst, C., and Sölkner, J. (2013). Evaluation of the lasso and the elastic net in genome-wide association studies. *Frontiers in Genetics*, 4(DEC):1–11.

Zhang, T. (2008). Adaptive Forward-Backward Greedy Algorithm for Sparse Learning with Linear Models. *Nips*, (1):1–8.