



HAL
open science

FAST COMPUTATION OF GENOME-METAGENOME INTERACTION EFFECTS

Christophe Ambroise, Julien Chiquet, Florent Guinot, Marie Szafranski

► **To cite this version:**

Christophe Ambroise, Julien Chiquet, Florent Guinot, Marie Szafranski. FAST COMPUTATION OF GENOME-METAGENOME INTERACTION EFFECTS. 2018. hal-01906069v1

HAL Id: hal-01906069

<https://hal.science/hal-01906069v1>

Preprint submitted on 26 Oct 2018 (v1), last revised 16 Jun 2020 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FAST COMPUTATION OF GENOME-METAGENOME INTERACTION EFFECTS

A PREPRINT

Christophe Ambroise

UMR 8071 LaMME – UÉVE, CNRS, ENSIIE, USC INRA
23 boulevard de France 91000 Évry, FRANCE
christophe.ambroise@genopole.cnrs.fr

Julien Chiquet

UMR MIA-Paris – AgroParisTech, INRA
16 rue claudes bernard 75005 Paris, FRANCE
julien.chiquet@agroparistech.fr

Florent Guinot^{1,2}

(1) UMR 8071 LaMME – UÉVE, CNRS, ENSIIE, USC INRA
23 boulevard de France 91000 Évry, FRANCE
(2) BIOptimize – 51000 Reims, FRANCE
florent.guinot@genopole.cnrs.fr

Marie Szafranski

UMR 8071 LaMME – UÉVE, CNRS, ENSIIE, USC INRA
23 boulevard de France 91000 Évry, FRANCE
marie.szafranski@math.cnrs.fr

October 26, 2018

ABSTRACT

Motivation: Association studies usually search for association between common genetic variants in different individuals and a given phenotype. In this work we consider two types of biological markers: genetic and metagenomic markers. The genotypic markers allow to characterise the individual by its inherited genetic information whereas the metagenomic markers are related to the environment. Both types of markers are available by millions and represent a unique signature characterizing each individual.

Results: We focus on the detection of interactions between groups of metagenomic and genetic markers to better understand the complex relationship between environment and genome in the expression of a given phenotype. We propose a method that reduces the dimension of the search space by selecting a subset of supervariables in both complementary datasets. These super variables stem from a weighted group structure defined on sets of variables of different scales. A Lasso selection is then applied on each type of supervariables to obtain a subset of potential interactions that will be explored via linear model testing.

Keywords Statistical learning, GWAS, hierarchical clustering, dimension reduction, interactions.

1 Introduction

1.1 Context

Genome Wide Association Study (GWAS). GWAS look for genetic markers linked to a phenotype of interest. Typically, hundreds of thousands of single nucleotide polymorphisms (SNPs) are analyzed with a limited sample size using high-density genotyping arrays. Usually the association between each SNP and the phenotype is tested using single-marker methods. Multivariate approaches may also be considered. They typically select multiple markers using

forward-selection methods or penalized methods. GWAS are a powerful tool for investigating the genetic architecture of complex diseases and have been successful in identifying hundreds of associated variants. However, they have been able to explain only a small proportion of the phenotypic variations expected from classical family studies. Uncovering some of the missing heritability may be achieved by taking into account correlations among variables, interaction with the environment and epistasis [Stanislas et al., 2017, and references therein], but this is rarely feasible in the context of GWAS because of the multiple testing burden and the high computational cost of such analyses [Manolio et al., 2009].

Other avenues to explain the variability in some traits of interest have yet to be explored, for instance an interesting lead would be to consider the contribution of microbial communities on the expression of a phenotype. Indeed, there is growing evidences of the role of gut microbiota in basic biological processes and in the development and progression of major human diseases such as infectious diseases, gastrointestinal cancers, metabolic diseases. . . [Wang et al., 2017]. In plants, the role of rhizosphere¹ microflora on plant growth is well known and has been widely studied [Mukerji et al., 2002, Pinton et al., 2007].

Metagenome Wide Association Study (MWAS). Equivalent type of association analysis has been conducted using metagenome [Wang and Jia, 2016, Segata et al., 2011]. Those metagenome association analyses may often explain larger variation of the phenotype than classical GWAS. These approaches have been successful in finding relevant associations for complex pathologies such as obesity, Crohn’s disease, colorectal cancer. . .

1.2 Combining genome and metagenome analyses

One possible way to relate genetic and metagenomic data consists in considering the metagenome as phenotype and thus performing quantitative trait locus (QTL) mapping. This kind of metagenome QTL analysis demonstrates the role of host genetics in shaping metagenomic diversity between individuals [Wang et al., 2016, Srinivas et al., 2013].

Another possibility for taking into account both types of variables consists in including metagenomic variables as environmental variables in GWAS. In that case interactions may naturally be modelled using a classical generalized linear model with interactions terms [Lin et al., 2013].

The main drawback of the later idea lies in the number of interactions to test, both datasets having a large number of variables. In order to reduce the dimension of the problem, variable selection or variable compression may be of use.

1.3 Taking structures into account in associations studies

Data compression for dimension reduction may be achieved in various ways. A usual distinction is often established between feature selection and feature extraction. Feature selection consists in selecting few relevant variables among the original ones, whereas feature extraction consists in computing new representative variables.

In our problem of association study, feature selection is often preferred to feature extraction for interpretative purposes. In this paper, we advocate for a mixed approach which combines feature extraction, based on underlying structures of genome and metagenome, and feature selection.

The idea of considering group structures has already been suggested both in the context of GWAS [Dehman et al., 2015] and MWAS [Qin et al., 2012]. In the context of prediction from gene expression regression, Park et al. [2007] proposed to hierarchically cluster the genes to obtain a dendrogram that reveals their nested correlation structure. At each level of the hierarchy, supergenes are computed as the average expression of the current clusters. It can be shown that regressing over supergenes improves the precision if the correlation structure is strong enough. In a similar fashion, Guinot et al. [2017] make use of the haplotype structure of the human genome to propose a dimension-reduction approach which can be applied in the context of GWAS.

The proposed method can be summarized as follows: (1) use a hierarchical clustering algorithm to identify a group structure within the data; (2) compress the hierarchical structure by creating supervariables; (3) perform a Lasso procedure on compressed variables with a penalty factor weighted by the length of the gap between two successive levels of a hierarchical clustering; (4) perform multiple hypothesis testing in a linear model with interactions.

1.4 Organization of the paper

The paper is organized as follows. Section 2 introduces the setting related to linear models of interactions and proposes a framework to learn with complementary datasets. Section 3 describes our method which combines compressions of

¹The rhizosphere is the term used to describe the zone of intense activity around the roots of leguminacea (Fabaceae) which contains a considerable diversity of microbial and mycorrhizal species.

data based on hierarchical structures, Lasso selection procedure and model testing for recovering relevant interactions. Finally, our approach is illustrated with numerical simulations in Section 4.

2 Learning with complementary datasets

This section introduces the setting with the notations. It also sketches the approach to define a compact model of interactions between complementary datasets.

2.1 Setting and notations

Let us consider observations stemming from two complementary views, G (for Genomic data) and M (for Metagenomic data), which are gathered into a training set $\mathcal{S} = \{(\mathbf{x}_i^G, \mathbf{x}_i^M, y_i)\}_{i=1}^N$, where $(\mathbf{x}_i^G, \mathbf{x}_i^M, y_i) \in \mathbb{R}^{D_G} \times \mathbb{R}^{D_M} \times \mathbb{R}$.

We assume an underlying biological information on G and M encoded as groups. The group structure over G is defined by N_G groups of variables $\mathcal{G} = \{\mathcal{G}_g\}_{g=1}^{N_G}$. We denote $\mathbf{x}_i^g \in \mathbb{R}^{D_g}$, the sample i restricted to the variables of G from group \mathcal{G}_g . Similarly, the group structure over M is defined by N_M groups of variables $\mathcal{M} = \{\mathcal{M}_m\}_{m=1}^{N_M}$ and $\mathbf{x}_i^m \in \mathbb{R}^{D_m}$ is the sample i restricted to the variables of M from group \mathcal{M}_m .

We also introduce $D_I = D_G \cdot D_M$ and $N_I = N_G \cdot N_M$, the number of variables and the number of groups that may interact.

Finally, we use the following convention: vectors of observations indexed with i , such as \mathbf{x}_i , will usually be row vectors while vectors of coefficients, such as β , will usually be column vectors.

2.2 Interactions in linear models

Interactions between data stemming from views G and M may be captured in the model

$$y_i = \mathbf{x}_i^G \gamma_G + \mathbf{x}_i^M \gamma_M + \mathbf{x}_i^G \Delta_{GM} (\mathbf{x}_i^M)^T + \epsilon_i, \quad (1)$$

where the vectors $\gamma_G \in \mathbb{R}^{D_G}$ and $\gamma_M \in \mathbb{R}^{D_M}$ respectively denote the linear effects related to G and M , the matrix $\Delta_{GM} \in \mathbb{R}^{D_G \times D_M}$ contains the interactions between all pairs of variables of G and M and $\epsilon_i \in \mathbb{R}$ is a residual error.

Models with interactions distinguish between *strong dependency* (SD) and *weak dependency* (WD), the first one being more common (see for instance [Bien et al., 2013] and the discussion therein). Under the hypothesis of *strong dependency*, an interaction is effective if and only if the corresponding single effects are also effective while the hypothesis of *weak dependency* implies that an interaction is effective if one of the main effect is also effective. Formally, for all variables $j \in \mathbf{x}^G$ and for all variables $j' \in \mathbf{x}^M$, if $\gamma_j, \gamma_{j'}$ and $\delta_{jj'}$ are the coefficients related to γ_G, γ_M and Δ_{GM} , then

$$\begin{array}{llll} (SD) & \delta_{jj'} \neq 0 & \Rightarrow & \gamma_j \neq 0 \quad \text{and} \quad \gamma_{j'} \neq 0, \\ (WD) & \delta_{jj'} \neq 0 & \Rightarrow & \gamma_j \neq 0 \quad \text{or} \quad \gamma_{j'} \neq 0. \end{array}$$

In this context, Bien et al. [2013] have proposed a sparse model of interactions which faces computational limitations for large dimensional problems according to Lim and Hastie [2015] and She et al. [2016]. While Lim and Hastie [2015] introduces a method for learning pairwise interactions in a regression model by solving a constrained overlapping Group-Lasso [Jacob et al., 2009] in a manner that satisfies strong dependencies, She et al. [2016] propose a formulation with an overlapping regularization that fits both kind of hypotheses and provide theoretical insights on the resulting estimators.²

Yet, the dimension $D_G + D_M + D_I$ involved in Problem (1) to estimate γ_G, γ_M and Δ_{GM} may be large especially for applications with an important number of variables such as in biology with genomic and metagenomic data. To reduce the dimension, we propose to compress the data according to an underlying structure which may be defined thanks to a prior knowledge or be uncovered with clustering algorithms.

2.3 Compact model

Assuming we are given a compression function for each group of G and M , we can shape Problem (1) into a compact form

$$y_i = \sum_{g \in \mathcal{G}} \tilde{x}_i^g \beta_g + \sum_{m \in \mathcal{M}} \tilde{x}_i^m \beta_m + \sum_{g \in \mathcal{G}} \sum_{m \in \mathcal{M}} \underbrace{(\tilde{x}_i^g \cdot \tilde{x}_i^m)}_{\phi_i^{gm}} \theta_{gm} + \epsilon_i, \quad (2)$$

²To our knowledge, their implementation based on alternating direction method of multipliers is not publicly available.

where $\tilde{x}_i^g \in \mathbb{R}$ is the i^{th} compressed sample of the variables that belong to the group g for the view G and $\beta_g \in \mathbb{R}$ is its corresponding coefficient. The counterparts on the group m for the view M are $\tilde{x}_i^m \in \mathbb{R}$ and $\beta_m \in \mathbb{R}$. Finally, $\theta_{gm} \in \mathbb{R}$ is the interaction between groups g and m .

We can reformulate Problem (2) in a vector form. Let $\tilde{\mathbf{x}}_i \in \mathbb{R}^{N_G}$, $\boldsymbol{\beta}_G \in \mathbb{R}^{N_G}$, $\tilde{\mathbf{x}}_i \in \mathbb{R}^{N_M}$ and $\boldsymbol{\beta}_M \in \mathbb{R}^{N_M}$ be

$$\begin{aligned}\tilde{\mathbf{x}}_i^G &= (\tilde{x}_i^1 \cdots \tilde{x}_i^g \cdots \tilde{x}_i^{N_G}), & \boldsymbol{\beta}_G &= (\beta_1 \cdots \beta_g \cdots \beta_{N_G})^T, \\ \tilde{\mathbf{x}}_i^M &= (\tilde{x}_i^1 \cdots \tilde{x}_i^m \cdots \tilde{x}_i^{N_M}), & \boldsymbol{\beta}_M &= (\beta_1 \cdots \beta_m \cdots \beta_{N_M})^T.\end{aligned}$$

We denote by $\phi_i \in \mathbb{R}^{N_I}$, the vector whose general component is given by ϕ_i^{gm} in Equation (2), that is

$$\phi_i = \left(\phi_i^{11} \cdots \phi_i^{1N_M} \cdots \phi_i^{gm} \cdots \phi_i^{N_G 1} \cdots \phi_i^{N_G N_M} \right),$$

and $\boldsymbol{\theta} \in \mathbb{R}^{N_I}$, the corresponding vector of coefficients, by

$$\boldsymbol{\theta} = (\theta_{11} \cdots \theta_{1N_M} \cdots \theta_{gm} \cdots \theta_{N_G 1} \cdots \theta_{N_G N_M})^T.$$

Finally, Problem (2) reads as a classical linear regression problem

$$y_i = \tilde{\mathbf{x}}_i^G \boldsymbol{\beta}_G + \tilde{\mathbf{x}}_i^M \boldsymbol{\beta}_M + \phi_i \boldsymbol{\theta} + \epsilon_i, \quad (3)$$

of dimension $N_G + N_M + N_I$.

2.4 Recovering relevant interactions

Compared to Problem (1) and provided that N_G and N_M are reasonably smaller than D_G and D_M , the dimension of Problem (3) decreases drastically so that it might be solved thanks to an appropriate optimization algorithm coupled with effective computational facilities. For instance, Donoho and Tsaig [2008] give an overview of ℓ_1 regularized algorithms to solve sparse problems like Lasso, which in our case could take the form:

$$\left\{ \begin{array}{l} \underset{\boldsymbol{\beta}_G, \boldsymbol{\beta}_M, \boldsymbol{\theta}}{\operatorname{argmin}} \quad \sum_{i=1}^n (y_i - \tilde{\mathbf{x}}_i^G \boldsymbol{\beta}_G - \tilde{\mathbf{x}}_i^M \boldsymbol{\beta}_M - \phi_i \boldsymbol{\theta})^2 \\ \quad + \lambda_G \sum_{g=1}^{N_G} |\beta_g| + \lambda_M \sum_{m=1}^{N_M} |\beta_m| + \lambda_I \sum_{g,m=1}^{N_I} |\theta_{gm}|, \end{array} \right.$$

with λ_G , λ_M and λ_I being the positive hyperparameters that respectively control the amount of sparsity related to coefficients $\boldsymbol{\beta}_G$, $\boldsymbol{\beta}_M$ and $\boldsymbol{\theta}$. Still, the dimension may remain large regarding the dimension $N_G + N_M + N_I$ compared to the number of observations N . Also, note that without additional constraints, such a formulation would not induce the dependance hypotheses (SD) and (WD). For that purpose, one could consider the works of Bien et al. [2013], Lim and Hastie [2015] or She et al. [2016] mentioned above. We present in the next Section another way to reduce further the dimension and fulfil the strong dependency hypothesis.

3 Method

In this section, we provide some elements to enhance Problem (3) for biological problems involving metagenomic and genomic data. The proposed approach, entitled SICOMORE for Selection of Interaction effects in COMpressed Multiple Omics REpresentations, is available through a R package at <https://github.com/fguinot/sicomore-pkg>.

3.1 Preprocessing of the data

To tackle problems that involve genomic and metagenomic interactions, some prior transformations are mandatory. Also, a first attempt to reduce the dimension may be achieved at this step.

Transformation for metagenomic data

Metagenome sequencing results in features which take the form of proportions in different samples. This kind of information is referred in the statistical literature as compositional data [Aitchison, 1982] that are known to be subject to negative correlation bias [Pearson, 1896, Aitchison, 1982]. The most common way to circumvent this issue is to transform the D_M features using centered log-ratios and to replace 0 values using maximum-likelihood approaches (see [Gloor et al., 2016] and references therein). A more detailed presentation of these aspects may be found in [Rau, 2017].

A first selection of variables

As seen in Section 2, we assume strong dependencies on interactions, which means that an interaction can be effective only if the two simple effects making up the interaction are involved in the problem. Then, it may be clever to apply a first process of selection to discard the inoperative single effects on G and M respectively. Different approaches may be envisioned to proceed this selection. Among them, screening rules can eliminate variables that will not contribute to the optimal solution of a sparse problem sweeping all the variables upstream to the optimization. When such a screening is appropriate, we may use the work of Lee et al. [2017] focused on Lasso problems, which present a recent overview of these techniques together with a screening rule ensemble. Once the screening is done, the optimization of a Lasso problem gives the final set of variables.

3.2 Structuring the data

Once the data are preprocessed, we can resort to hierarchical clustering using Ward criterion with appropriate distances to uncover the tree structures.

Clustering of metagenomic data

A common approach to analyse metagenomic data is to group sequences into taxonomic units. The features stemming from metagenome sequencing are often modeled as Operational Taxonomic Units (OTUs), each OTU representing a biological species according to some degree of taxonomic similarity. Chen et al. [2013] propose a comparison of methods to identify OTUs that includes hierarchical clustering.

While the structure on microbial species could be defined according to the underlying phylogenetic tree, it also makes sense to use more classical distances to define a hierarchy based on the abundances of OTUs. In our application we use an agglomerative hierarchical clustering with the Ward criterion.

Clustering of genomic markers

On the other hand, when the genomic information is available through SNP, the tree structure on G will be defined using a Ward spatially constrained hierarchical clustering algorithm which integrates the linkage disequilibrium as the measure of dissimilarity [Dehman et al., 2015].

3.3 Using the structure efficiently

Different approaches related to the problem of finding an optimal number of clusters may be envisioned to find the optimal cut in a tree structure obtained with hierarchical clustering (see for instance [Milligan and Cooper, 1985] or [Gordon, 1999]). Whatever the chosen approach, a systematic exploration of different levels of the hierarchy is mandatory to find this optimal cut. We define an alternative strategy to bypass this expensive exploration which consists in:

- (a) Expanding the hierarchy considering all possible groups at a single level ;
- (b) Assigning a weight to each group based on gap distances between two consecutive groups in the hierarchy;
- (c) Compressing each group into a supervariable.

The different steps of these strategy are illustrated in Figure 1, from the original tree structure in Figure 1(a) to a final flatten, weighted and compressed representation in Figure 1(c).

Expanding the hierarchy

(a)

To reduce the dimension involved in Problem (3), the first step consists in flattening the respective tree structures obtained on views G and M so that only a group structure remains. Thus, each group of variables defined at the deepest level may be included in other groups of larger scales, as shown in Figure 1(b).

Assigning weights to the groups

(b)

To keep track of the tree structure, we may integrate an additional measure quantifying the quality of the groups on two successive levels. More specifically, for a tree structure of height H and for $1 \leq h \leq H - 1$, we define s_h as the gap between heights h and $h - 1$. Following the lines of Grimonprez [2016] for the Multi-Layer Group-Lasso, we define this quantity as $\rho_h = 1/\sqrt{s_h}$. The process is shown in Figure 1(a) and 1(b).

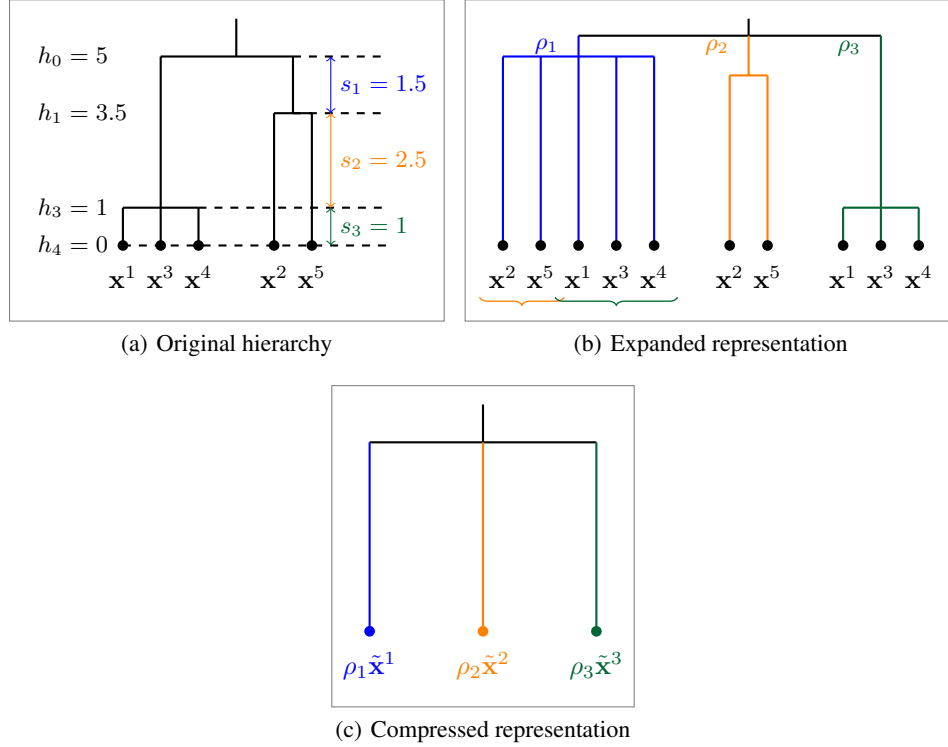


Figure 1: Dimension reduction strategy. (a) Original hierarchical tree with an example for 5 variables. (b) Expanded representation of the tree with all possible weighted groups derived from the original hierarchy. The blue group gathers the variables contained in the orange and green groups. (c) Compressed representation of the tree after construction of the supervariables.

Compressing the data

(c)

To summarize each group of variables, the mean, the median or other quantiles may be used as well as more sophisticated representations based on eigen values decompositions such as the first factor of a PCA.

3.4 Identification of relevant supervariables

With this compressed representation at hand, we can recover relevant interactions with a multiple testing strategy.

Selection of supervariables

The compression is a key ingredient to reduce significantly the dimension involved in Problem (3). Yet, we are going a step further with an additional feature selection process applied to the compressed variables, as suggested at the beginning of this section to preprocess the data, using screening rules and / or applying a Lasso optimization on each view G and M :

$$\operatorname{argmin}_{\beta_G} \sum_{i=1}^n (y_i - \tilde{\mathbf{x}}_i^G \beta_G)^2 + \lambda_G \sum_{g=1}^{N_G} \rho_g |\beta_g|,$$

and

$$\operatorname{argmin}_{\beta_M} \sum_{i=1}^n (y_i - \tilde{\mathbf{x}}_i^M \beta_M)^2 + \lambda_M \sum_{m=1}^{N_M} \rho_m |\beta_m|,$$

with penalty factors being defined by $\rho_g = 1/\sqrt{s_g}$ and $\rho_m = 1/\sqrt{s_m}$ as explained in Section 3.2.

Linear model testing

In a feature selection perspective, the relevant interactions may be recovered separately considering each selected group $g \in \mathcal{G}$ coupled with each selected group $m \in \mathcal{M}$ in a linear model of interaction and by performing an hypothesis test (a standart t -test for instance) on each parameter θ_{gm} :

$$y_i = \tilde{x}_i^g \beta_g + \tilde{x}_i^m \beta_m + (\tilde{x}_i^g \cdot \tilde{x}_i^m) \theta_{gm} + \epsilon_i. \quad (4)$$

This strategy has the advantage of highlighting all the potential interactions between the selected simple effects in an exploratory rather than a predictive analysis perspective. Also it may be regarded as an alternative shortcut to Problem (3) in that it involves N_I problems of dimension 3 instead of a potentially large problem of dimension $N_G + N_M + N_I$. Finally, this scheme of selection preserves strong dependencies by construction.

4 Numerical simulations

We provide here numerical simulations to assess the ability of SICOMORE to recover relevant interactions against three other methods. We also show that our method is computationally competitive compared to the others.

4.1 Data generation

Generation of metagenomic and genomic data matrices

Genomic data. In order to get a matrix \mathbf{X}^G close to real genomic data, we used the software HAPGEN2 [Su et al., 2011]. This software allows to simulate an entire chromosome conditionally on a reference set of population haplotypes (from HapMap3) and an estimate of the fine-scale recombination rate across the region, so that the simulated data share similar patterns with the reference data. We generate the chromosome 1 using the haplotype structure of CEU population (Utah residents with Northern and Western European ancestry from the CEPH collection) as reference set and we selected $D_G = 200$ variables from this matrix to obtain the simulated dataset. An example of the linkage disequilibrium structure among the simulated SNPs is illustrated in Figure 2(a).

Metagenomic data. The data matrix \mathbf{X}^M , with $D_M = 100$ variables, has been generated using a multivariate Poisson-log normal distribution [Aitchison and Ho, 1989] with block structure dependencies. The Poisson-log normal model is a latent gaussian model where latent vectors $Z_i \in \mathbb{R}^{D_M}$ are drawn from a multivariate normal distribution

$$Z_i \sim \mathcal{N}_{D_M}(0, \Sigma),$$

with Σ being a covariance matrix that allows to obtain a correlation structure among the variables. The random variable X_{ij}^M related to the centered phenotypic count data is then drawn from a Poisson distribution conditionally on Z_i

$$X_{ij}^M | Z_{ij} \sim \mathcal{P}(e^{\mu_j + Z_{ij}}).$$

The block structure, pictured in Figure 2(b), has been obtained by drawing a latent multivariate normal vector using a covariance matrix such that the correlation level between the latent variables of a group are between 0.5 and 0.95. Simulating this way, we obtain a matrix of count data with a covariance structure close to what is observed with metagenomic data. As stated in Section 3.1, we calculated the proportions in each random variables and transformed them using centered log-ratios.

Generation of the phenotype

For all simulations, we used a fixed value of $N_M = 6$ groups for the matrix \mathbf{X}^M and for the case of the matrix \mathbf{X}^G , since we cannot exactly control the block structure with HAPGEN2, we used the Gap Statistic [Tibshirani et al., 2001] to identify a number of groups in the hierarchy. For instance in Figure 2(a), the Gap Statistic identified $N_G = 16$ groups. The supervariables were then calculated using averaged groups of variables to obtain the two matrices of supervariables, $\tilde{\mathbf{X}}^G$ and $\tilde{\mathbf{X}}^M$.

To generate the phenotype, we considered a data structure for which the data to regress has been generated using supervariables according a linear model with interactions of the form:

$$y_i = \sum_{g \in \mathcal{S}^G} \tilde{x}_i^g \beta_g + \sum_{m \in \mathcal{S}^M} \tilde{x}_i^m \beta_m + \sum_{g \in \mathcal{S}^G} \sum_{m \in \mathcal{S}^M} \underbrace{(\tilde{x}_i^g \cdot \tilde{x}_i^m)}_{\phi_i^{gm}} \theta_{gm} + \epsilon_i, \quad (5)$$



Figure 2: Examples of hierarchical structures: correlations observed on (a) genomic data \mathbf{X}^G and (b) metagenomic data \mathbf{X}^M .

where \mathcal{S}^G and \mathcal{S}^M are subsets of randomly chosen effects from the matrices $\tilde{\mathbf{X}}^G$ and $\tilde{\mathbf{X}}^M$ respectively, \tilde{x}_i^g is the i^{th} sample of the g effect and β_g its corresponding coefficient, \tilde{x}_i^m is the i^{th} sample of the m effect and β_m its corresponding coefficient. Finally, θ_{gm} is the interaction between variables \tilde{x}_i^g and \tilde{x}_i^m .

We considered $I \in \{1, 3, 5, 7, 10\}$ true interactions between some supervariables to generate the phenotype so that I blocks of the coefficients of θ_{gm} have non zero values. The process was repeated 30 times for each couple of parameters in $N = \{50, 100, 200\} \times sd(\epsilon) = \{0.5, 1, 2\}$.

4.2 Comparison of methods

To evaluate the performance of our method, SICOMORE, to retrieve the true causal interactions, we compared it with three other methods, namely **HCAR** [Park et al., 2007], **MLGL** [Grimonprez, 2016] and **glnetnet** [Lim and Hastie, 2015]. It is worth mentioning that SICOMORE is an approach that borrow from HCAR and MLGL and that is designed to detect interactions. We had then to adapt these approaches to our problematic, as we will describe it in the following sections, they are therefore not evaluated in the context they were meant to be used. Thus, the purpose of this evaluation is to know if SICOMORE is capable of improving the individual performance of these methods by combining them to detect statistical interactions.

Hierarchical Clustering and Averaging Regression

(HCAR)

HCAR is a two-step procedure that combines hierarchical clustering and Lasso. In this work, Park et al. [2007] are averaging the genes within the clusters obtained from an hierarchical clustering to define supervariables, named supergenes, and use them to fit regression models. This methodology can be simply adapted to our problematic by performing two hierarchical clustering on each data matrices \mathbf{X}^G and \mathbf{X}^M and then compute the unweighted compressed representations of those hierarchies as explained in Section 3(c) and illustrated in Figure 1(c). We can then fit a Lasso regression model on both compressed representations with interactions between all possible groups. We consider that HCAR is able to retrieve a true causal interaction if the Lasso procedure selects the interaction term at the correct levels of the two hierarchies.

Multi-Layer Group-Lasso

(MLGL)

Grimonprez [2016] defines MLGL as a two-step procedure that combines a hierarchical clustering with a Group-Lasso regression. It is a weighted version of the overlapping Group-Lasso [Jacob et al., 2009] which performs variable selection on multiple group partitions defined by the hierarchical clustering. A weight is attributed to each possible group identified at all levels of the hierarchy as described in Section 3(b). Such weighting scheme favors groups creating at the origin of large gaps in the hierarchy.

The model is fitted with weights on the groups defined by the expanded representation of the two hierarchies as illustrated in Figure 1(b). This method does not work on the compressed supervariables but on the initial variables. Our evaluation considers that the method is able to retrieve real interactions if it selects the correct interaction terms between two groups of variables at the right level in both hierarchies.

Group-Lasso interaction network

(glinternet)

glinternet [Lim and Hastie, 2015] is a procedure that considers pairwise interactions in a linear model in a manner that satisfies strong dependencies between main and interaction effects: whenever an interaction is estimated to be non-zero, both its associated main effects are also included in the model. This method uses a Group-Lasso model to accommodate with categorical variables and apply constraints on the main effects and interactions to result in interpretable interaction models. The glinternet model fits a hierarchical Group-Lasso model with constraints on the main and interactions effects as specified in Section 2.4 whilst accommodating for the strong dependences hypothesis by adding an appropriate penalty to the loss function (we refer the reader to [Lim and Hastie, 2015] for more details on the form of the penalty). For very large problems (with a number of variables $\geq 10^5$), the Group-Lasso procedure is preceded by a screening step that gives a candidate set of main effects and interactions. They use an adaptive procedure based on the strong rules [Tibshirani et al., 2012] for discarding predictors in lasso-type problems. Since this method can only work at the level of variables, it was necessary to include a group structure into the analysis. Therefore, we decided to fit the glinternet model on the compressed variables and to constraint the model to only fit the interaction terms between the supervariables of the two matrices $\tilde{\mathbf{X}}^G$ and $\tilde{\mathbf{X}}^M$. We explicitly removed all interaction terms between supervariables belonging to the same data matrix.

For a fair comparison with the other methods, we considered two options namely GLtree and GLgap. On one hand, option GLtree works on the unweighted compressed representations of the two hierarchies (Figure 1(c)) thus considering all the possible interactions between the supervariables of the two datasets. On the other hand, option GLgap considers only the interactions between the compressed variables constructed at a specific level in the hierarchies, chosen by the Gap Statistic. Given that D^G and D^M are the number of variables in \mathbf{X}^G and \mathbf{X}^M , the dimension of the compressed matrices $\tilde{\mathbf{X}}^G$ and $\tilde{\mathbf{X}}^M$ are respectively $\tilde{D}^G = D^G + (D^G - 1)$ and $\tilde{D}^M = D^M + (D^M - 1)$. Thus, for GLtree the number of interactions to investigate are $\tilde{D}^G \times \tilde{D}^M$ while for GLgap this number will depend on the level chosen by the Gap statistic but will be either way smaller since we consider only a specific level of the hierarchy in this option. In the numerical simulations, given that $D^G = 200$ and $D^M = 100$, the use of strong rules to discard variables is therefore not necessary as Lim and Hastie [2015] argued that glinternet can handle very large problems without any screening (360M candidate interactions were fitted when evaluating the method on real data examples).

4.3 Evaluation metrics

For each run, we evaluated the quality of the variable selection using Precision and Recall. More precisely, we compared the true interaction matrix θ that we used to generate the phenotype with the estimated interaction matrix $\hat{\theta}$ compute for each models.

For all possible $D_G \times D_M$ interactions, $\theta_{jj'}$ is the interaction term between variable $j \in \mathbf{X}^G$ and variable $j' \in \mathbf{X}^M$, we determined the following confusion matrix:

| | | |
|-----------------------|--------------------------|-----------------------------|
| | $\hat{\theta}_{jj'} = 0$ | $\hat{\theta}_{jj'} \neq 0$ |
| $\theta_{jj'} = 0$ | True Negative | False Positive |
| $\theta_{jj'} \neq 0$ | False Negative | True Positive |

and hence compute Precision = $\frac{TP}{(FP+TP)}$ and Recall = $\frac{TP}{(FN+TP)}$. An example of the interaction matrix $\hat{\theta}$ is given in Figure 3 for $I = 5$ blocks in interaction.

In this context, a True Positive corresponds to a significant p -value on a true causal interaction, a False Positive to a significant p -value on a noise interaction, and a False Negative to a non-significant p -value on a true causal interaction.

For all methods, we correct for multiple testing by controlling the Family Wise Error Rate using the method of Holm-Bonferroni. Even though it is known to be stringent, we chose to rely on Holm-Bonferroni method to adjust for multiple testing because the number hypothesis tests performed in our simulation context is not that high. In a high-dimensional context such as with the analysis of real DNA chip data, we would rather use the Benjamini-Hochberg method for the control of the false discovery rate.

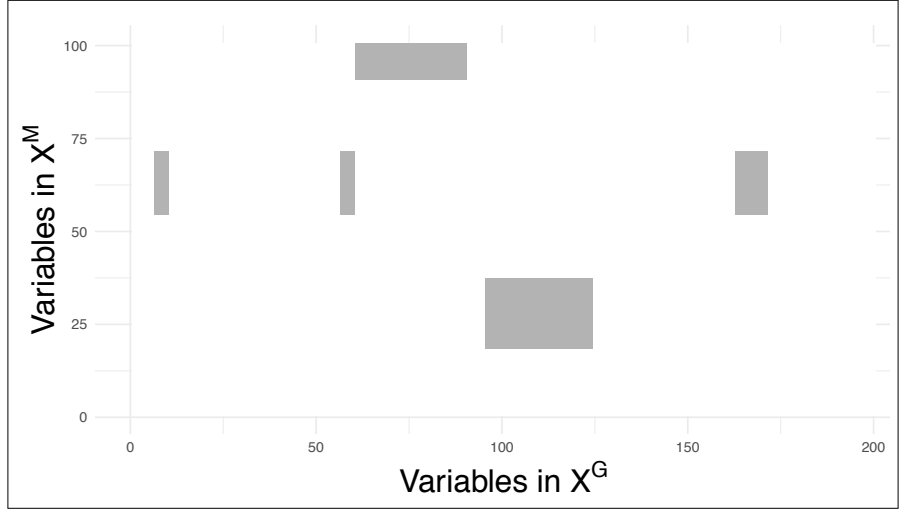


Figure 3: **Illustration of the true interaction matrix θ** with $I = 5$, $\sigma = 0.5$ and $n = 100$. Each non-zero value in this matrix is considered as a true interaction between two variables.

4.4 Performance results

The performances of each method to retrieve the true causal interactions are illustrated in Figure 5(a) for precision and Figure 5(b) for recall. For the sake of clarity we only show the results for $I = 7$ blocks of variables in interaction. The results for $I \in \{1, 3, 5, 10\}$ are provided in Annexes A as a supplementary material. The plots in Figure 4 represent the recovered confusion matrices of interaction θ_{gm} for each compared algorithm for one particular set of simulation parameters ($I = 5$, $\sigma = 0.5$, $n = 100$).

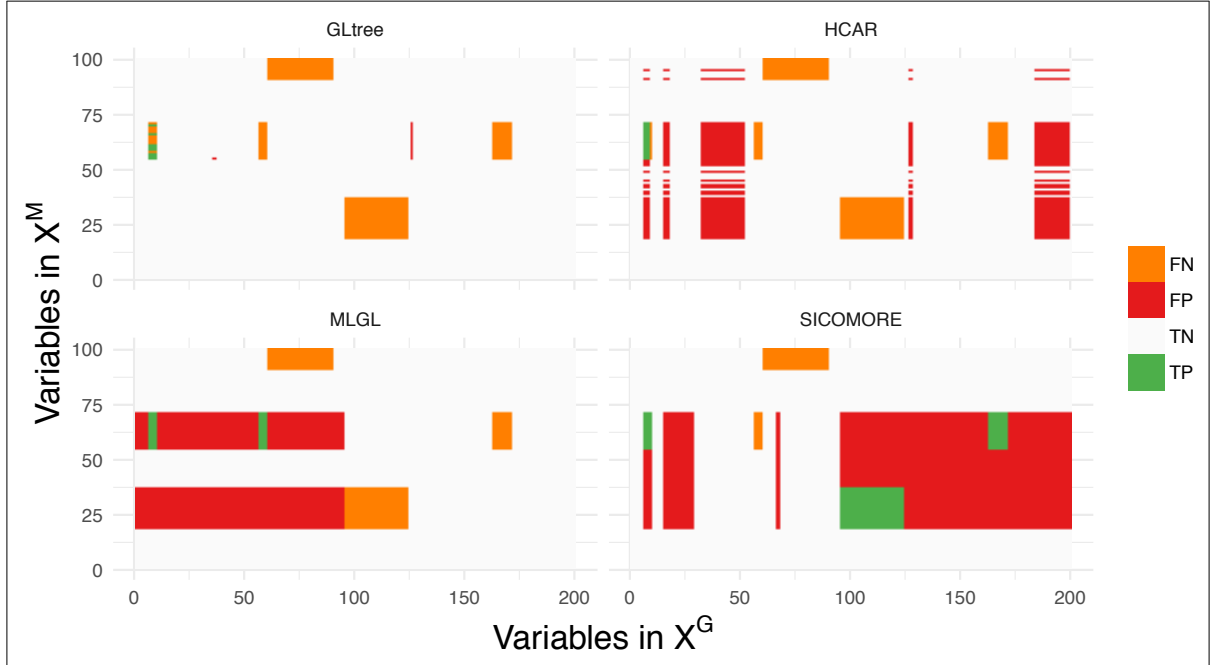


Figure 4: **Confusion matrices of interactions $\hat{\theta}_{jj'}$** for each compared algorithm with the following simulation parameters: $I = 5$, $\sigma = 0.5$, $n = 100$. We can see in this example that MLGL and SICOMORE behaves similarly with very large genomic regions identified while HCAR tends to work with smaller genomic and metagenomic regions.

The results in terms of recall reveal good abilities of MLGL and SICOMORE to retrieve True Positive interactions, with an overall advantage for our method. HCAR achieves a lower performance due to the fact that it favours the selection of

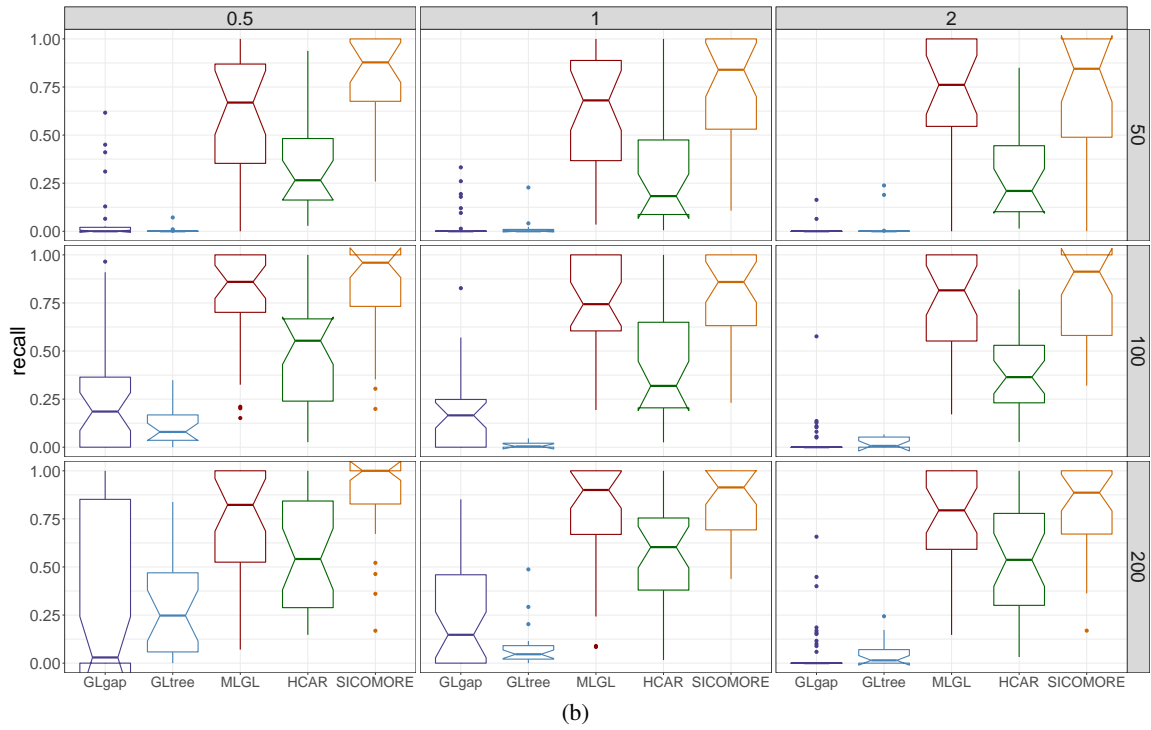
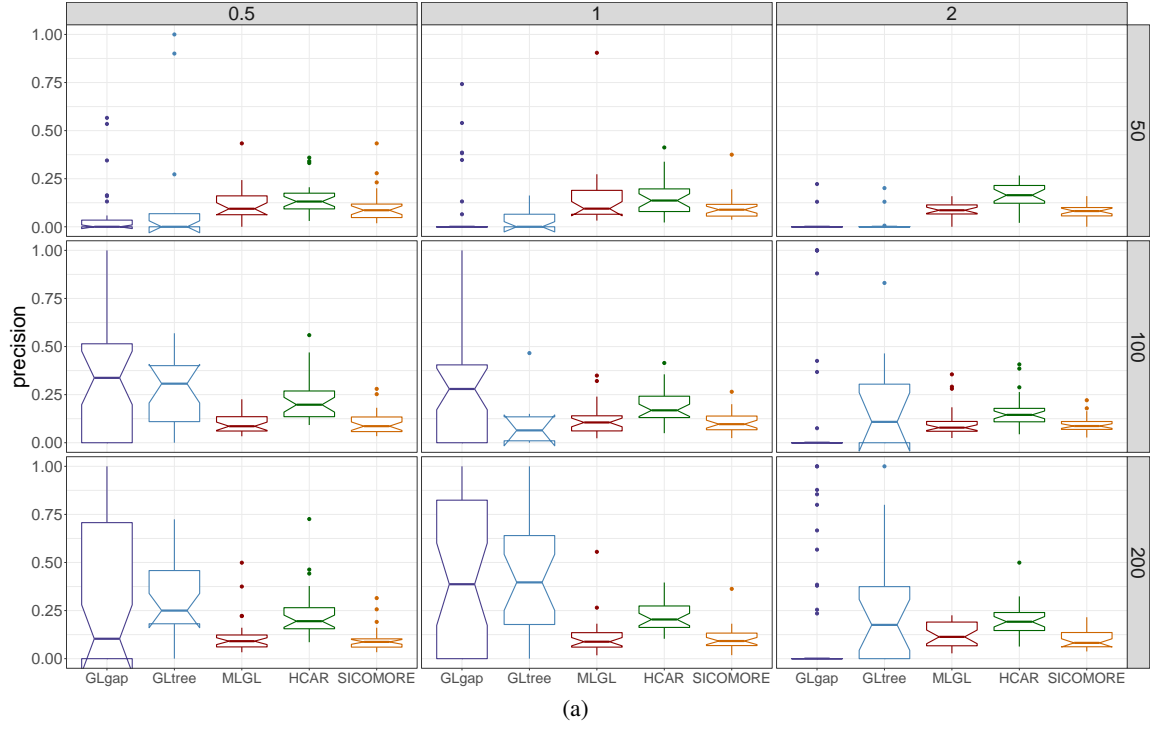


Figure 5: Boxplots of (a) Precision and (b) Recall obtained on the numerical simulations with a Bonferroni-Holm correction for $I = 7$ blocs. The lines show the results for different number of observations (top: $N = 50$, middle: $N = 100$ and bottom: $N = 200$) and the columns the difficulty of the problem (left: $\epsilon = 0.5$, middle: $\epsilon = 1$ and right: $\epsilon = 2$). The boxplots are best seen in colors: from the left to the right, MLGL is in blue, HCAR is in red, SICOMORE is in green and glinternet is in orange.

small groups which are only partly contained in the groups that generate the interactions showing that the weighting scheme of MLGL and SICOMORE is efficient. GLgap is not able to retrieve relevant interactions but the way to define the structure among variables, using the Gap Statistic, is also quite different than for the three other methods. In terms of precision, all methods perform poorly with a significant number of false positive interactions. MLGL and SICOMORE tend to select groups of variables and supervariables too high in the tree structure, inducing false positives which are spatially closed to the true interactions. HCAR, which favours small groups as explained above, is less subject to this issue. The behaviour of GLgap may vary according to the selected cut with the Gap statistic into the tree structure while option GLtree exhibit slightly better precision. Still, the method glinternet is globally not able to retrieve correctly the true interactions whether or not it uses the compressed or original representation.

4.5 Computational time

In order to decrease the calculation time in our algorithm, we chose to restrain the search space in the tree to a certain amount depending on the number of initial features. We can choose to limit the search in the area of the tree where the jumps in the hierarchy are the highest and arbitrarily set the number of groups to evaluate at five times the number of initial features. By doing so, we are reducing the number of variables to be fitted in the Lasso regression without affecting the performance in terms of Recall and Precision.

We compared the computational performance of our method with the three others by varying the number of variables in $\tilde{\mathbf{X}}^G$. We repeated the number of evaluation five times for each size of $\tilde{\mathbf{X}}^G$ and averaged the calculation time.

| N_G | 50 | 100 | 500 | 1000 | 1500 | 2000 | 3000 | 4000 |
|-----------|------|------|------|------|-------|-------|-------|--------|
| SICOMORE | 0.01 | 0.01 | 0.02 | 0.03 | 0.03 | 0.04 | 0.05 | 0.06 |
| HCAR | 0.21 | 0.34 | 0.82 | 0.76 | 0.75 | 0.96 | 0.93 | 1.09 |
| MLGL | 0.06 | 0.09 | 3.35 | 0.86 | 3.12 | 4.52 | 8.02 | 24.20 |
| glinetnet | 0.07 | 0.28 | 0.67 | 3.83 | 11.69 | 26.31 | 88.17 | 210.64 |

Table 1: Results of averaged calculation time (in minutes) over 5 replicates for varying dimensions of $\tilde{\mathbf{X}}^G$.

We can conclude from the results presented in Table 1 that two methods, MLGL and glinternet, are not suitable for large-scale analysis of genomic data since the calculation time increase drastically as soon as the dimension of the problem exceed a few thousand variables. HCAR and SICOMORE behaves similarly. That being said, remember that our implementation of HCAR is tuned with an unweighted compressed representation in the same spirit than SICOMORE, avoiding to choose the optimal cut in the tree. With its original strategy based on a K -folds cross-validation, there is no doubt that the gap between HCAR and SICOMORE would have been much larger. Indeed, the computational cost of an additional exploration to find the optimal cut in HCAR grows with the number of variables and therefore with h_T , the height of the tree. HCAR has to evaluate $h_T \times K$ compressed models while SICOMORE only has to compress $h_T - 1$ groups to evaluate the final model.

5 Conclusion

One possible way to understand the expression of certain diseases is to consider gene-environment interactions. Sensitivity to environmental risk factors for a disease may be inherited, leading to cases where individuals exposed to the same environment but with different genotypes can be affected differently, resulting in different disease phenotypes. In the context of medical genetics and epidemiology, the study of gene-environment interactions is of great importance. Indeed, if we estimate only the separate contributions of genes and environment to a disease, and ignore their interactions, we will incorrectly estimate the fraction of phenotypic variance attributable to genes, environment, and their joint effect.

Although the detection of interaction effects in a high-dimensional remains a difficult problem, on one hand due to the multiple testing burden and on the other hand to the small effect sizes in term of significance, our approach has demonstrated the ability to recover interaction effects with a high statistical power. In our simulations, whether we varied the sample sizes, noise or number of true interactions, SICOMORE always exhibited the strongest recall compared to MLGL, HCAR or glinternet. This can be explained mainly by the fact that we advantageously use the strengths of different methods to combine them in a powerful single algorithm. Also, it is worth noting that SICOMORE is significantly more efficient in terms of computational time compared to the three others.

Despite these interesting results, some aspects may be addressed in future works to further improve SICOMORE in terms of model consistency. First of all, the variable selection step to select the supervariables in both complementary datasets may suffer from instability when setting the amount of selection. To circumvent these aspect, one could resort

to resampling techniques [Bach, 2008, Meinshausen and Bühlmann, 2010]. Then, although the lasso procedure is relevant for a dimension-reduction purpose, it may induce some biases in the multiple testing procedure used afterwards since this variable selection step is performed before adjusting the p -values. One way around this problem could be to use post-hoc inference for multiple comparisons [Goeman et al., 2011]. These kind of extensions should have a positive impact on precision results.

References

- John Aitchison and C. H. Ho. The multivariate poisson-log normal distribution. *Biometrika*, 76(4):643–653, 1989.
- Jonh Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society, Series B (Methodological)*, 44(2):139–177, 1982.
- Francis R. Bach. Bolasso: model consistent lasso estimation through the bootstrap. In *Proceedings of the 25th Annual International Conference on Machine Learning*, pages 33–40, 2008.
- Jacob Bien, Jonathan Taylor, and Robert Tibshirani. A Lasso for hierarchical interactions. *Annals of statistics*, 41(3): 1111, 2013.
- Wei Chen, Clarence K. Zhang, Yongmei Cheng, Shaowu Zhang, and Hongyu Zhao. A comparison of methods for clustering 16S rRNA sequences into OTUs. *PLoS ONE*, 8(8):e70837, 2013.
- Alia Dehman, Christophe Ambroise, and Pierre Neuvial. Performance of a blockwise approach in variable selection using linkage disequilibrium information. *BMC bioinformatics*, 16(1):148, 2015.
- David L. Donoho and Yaakov Tsaig. Fast solution of ℓ_1 -norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory*, 54(11):4789–4812, 2008.
- Gregory B. Gloor, Jean M. Macklaim, Michael Vu, and Andrew D. Fernandes. Compositional uncertainty should not be ignored in high-throughput sequencing data analysis. *Austrian Journal of Statistics*, 45:73–87, 2016.
- Jelle J Goeman, Aldo Solari, et al. Multiple testing for exploratory research. *Statistical Science*, 26(4):584–597, 2011.
- Aharon D. Gordon. *Classification*. Monographs on statistics and applied probability. Chapman & Hall, 1999.
- Quentin Grimonprez. *Sélection de groupes de variables corrélées en grande dimension*. PhD thesis, Université de Lille, 2016.
- Florent Guinot, Marie Szafranski, Christophe Ambroise, and Franck Samson. Learning the optimal scale for gwas through hierarchical snp aggregation. *arXiv preprint arXiv:1710.01085*, 2017.
- Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group Lasso with overlap and graph Lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 433–440, 2009.
- Seunghak Lee, Nico Görnitz, Eric P. Xing, David Heckerman, and Christoph Lippert. Ensembles of Lasso screening rules. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017.
- Michael Lim and Trevor Hastie. Learning interactions via hierarchical group-Lasso regularization. *Journal of Computational and Graphical Statistics*, 24(3):627–654, 2015.
- Xinyi Lin, Seunggeun Lee, David C. Christiani, and Xihong Lin. Test for interactions between a genetic marker set and environment in generalized linear models. *Biostatistics*, 14(4):667–681, 2013.
- Teri A. Manolio, Francis S. Collins, Nancy J. Cox, David B. Goldstein, Lucia A. Hindorff, David J. Hunter, Mark I. McCarthy, Erin M. Ramos, Lon R. Cardon, Aravinda Chakravarti, Judy H. Cho, Alan E. Guttmacher, Augustine Kong, Leonid Kruglyak, Elaine Mardis, Charles N. Rotimi, Montgomery Slatkin, David Valle, Alice S. Whittemore, Michael Boehnke, Andrew G. Clark, Evan E. Eichler, Greg Gibson, Jonathan L. Haines, Trudy F. C. Mackay, Steven A. McCarroll, and Peter M. Visscher. Finding the missing heritability of complex diseases. *Nature*, 461(7265):747–753, October 2009.
- Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society, Series B (Methodological)*, 72(4):417–473, 2010.
- Glenn W. Milligan and Martha C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.

- Krishna Gopal Mukerji, C Manoharachary, and BP Chamola. *Techniques in mycorrhizal studies*. Springer Science & Business Media, 2002.
- Mee Young Park, Trevor Hastie, and Robert Tibshirani. Averaged gene expressions for regression. *Biostatistics*, 8(2): 212–227, 2007.
- Karl Pearson. Mathematical contributions to the theory of evolution. on a form of spurious correlation which may arise when indices are used in the measurement of organs. *Proceedings of the Royal Society of London*, 60:489–498, 1896.
- Roberto Pinton, Zeno Varanini, and Paolo Nannipieri. *The rhizosphere: biochemistry and organic substances at the soil-plant interface*. CRC press, 2007.
- Junjie Qin, Yingrui Li, Zhiming Cai, Shenghui Li, Jianfeng Zhu, Fan Zhang, Suisha Liang, Wenwei Zhang, Yuanlin Guan, Dongqian Shen, et al. A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature*, 490(7418):55–60, 2012.
- Andrea Rau. *Statistical methods and software for the analysis of transcriptomic data*. Habilitation à diriger des recherches, Université d’Evry Val d’Essonne, 2017.
- Nicola Segata, Jacques Izard, Levi Waldron, Dirk Gevers, Larisa Miropolsky, Wendy S Garrett, and Curtis Huttenhower. Metagenomic biomarker discovery and explanation. *Genome biology*, 12(6):R60, 2011.
- Yiyuan She, Zhifeng Wang, and He Jiang. Group regularized estimation under structural hierarchy. *Journal of the American Statistical Association*, 113(521):445–454, 2016.
- Girish Srinivas, Steffen Möller, Jun Wang, Sven Künzel, Detlef Zillikens, John F Baines, and Saleh M Ibrahim. Genome-wide mapping of gene–microbiota interactions in susceptibility to autoimmune skin blistering. *Nature communications*, 4, 2013.
- Virginie Stanislas, Cyril Dalmasso, and Christophe Ambroise. Eigen-epistasis for detecting gene-gene interactions. *BMC Bioinformatics*, 18(1):54:1–54:14, 2017.
- Z. Su, J. Marchini, and P. Donnelly. Hapgen2: simulation of multiple disease snps. *Bioinformatics*, 27(16):2304, 2011.
- Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 63, 2001.
- Robert Tibshirani, Jacob Bien, Jerome Friedman, Trevor Hastie, Noah Simon, Jonathan Taylor, and Ryan J Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):245–266, 2012.
- Baohong Wang, Mingfei Yao, Longxian Lv, Zongxin Ling, and Lanjuan Li. The human microbiota in health and disease. *Engineering*, 3(1):71 – 82, 2017. ISSN 2095-8099. doi: <https://doi.org/10.1016/J.ENG.2017.01.008>. URL <http://www.sciencedirect.com/science/article/pii/S2095809917301492>.
- Jun Wang and Huijue Jia. Metagenome-wide association studies: fine-mining the microbiome. *Nature Reviews Microbiology*, 14(8):508–522, 2016.
- Jun Wang, Louise B. Thingholm, Jurgita Skiecevičienė, Philipp Rausch, Martin Kummén, Johannes R Hov, Frauke Degenhardt, Femke-Anouska Heinsen, Malte C Rühlemann, Silke Szymczak, et al. Genome-wide association analysis identifies variation in vitamin d receptor and other host factors influencing the gut microbiota. *Nature Genetics*, 2016.

A Supplementary materials

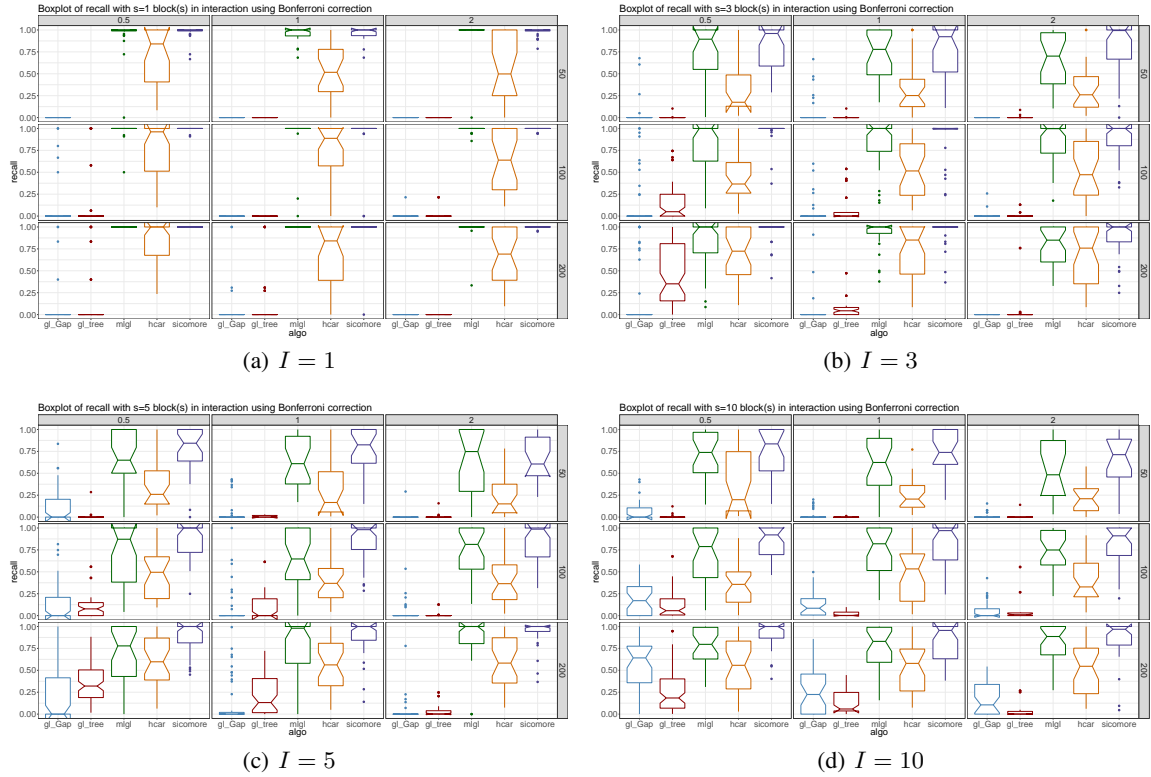


Figure 6: Boxplots of Recall obtained on the numerical simulations with a Bonferroni-Holm correction for $I = \{1, 3, 5, 10\}$ blocs. The lines show the results for different number of observations (top: $N = 50$, middle: $N = 100$ and bottom: $N = 200$) and the columns the difficulty of the problem (left: $\epsilon = 0.5$, middle: $\epsilon = 1$ and right: $\epsilon = 2$). The boxplots are best seen in colors: from the left to the right, MLGL is in blue, HCAR is in red, SICOMORE is in green and glinternet is in orange.

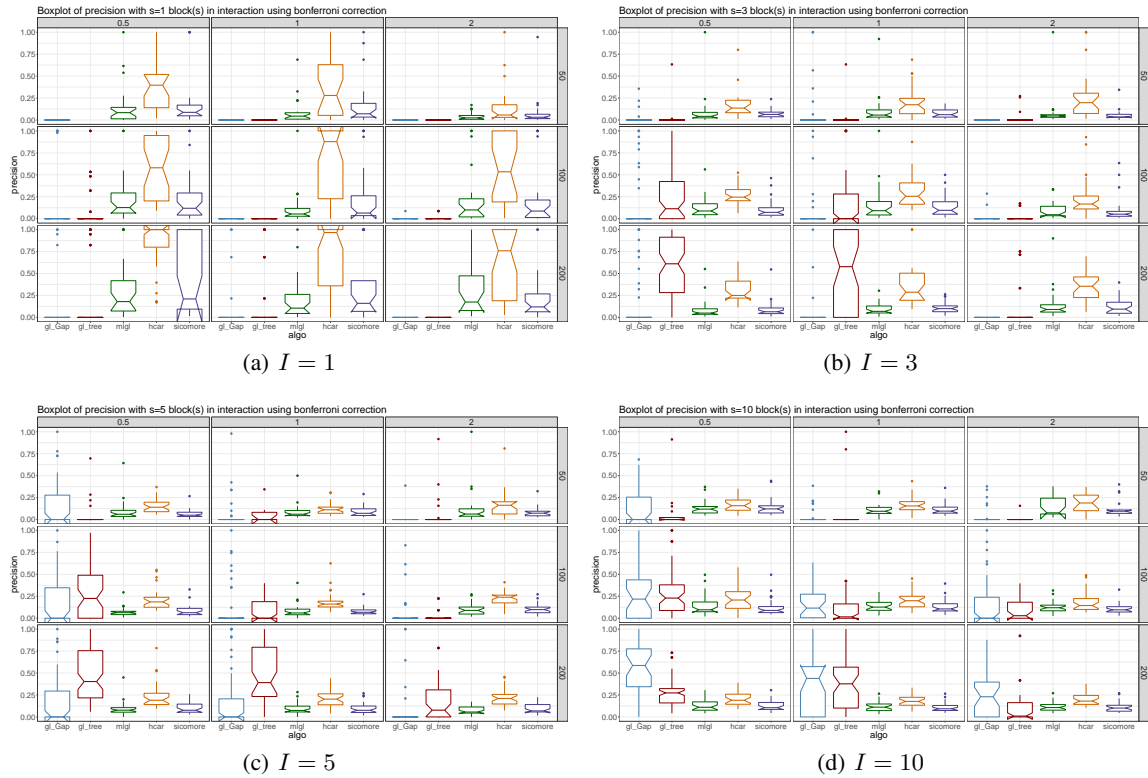


Figure 7: Boxplots of Precisions obtained on the numerical simulations with a Bonferroni-Holm correction for $I = \{1, 3, 5, 10\}$ blocs. The lines show the results for different number of observations (top: $N = 50$, middle: $N = 100$ and bottom: $N = 200$) and the columns the difficulty of the problem (left: $\epsilon = 0.5$, middle: $\epsilon = 1$ and right: $\epsilon = 2$). The boxplots are best seen in colors: from the left to the right, MLGL is in blue, HCAR is in red, SICOMORE is in green and glinternet is in orange.