Alternative Methods for H1 Simulations in Genome-Wide Association Studies

V. Perduca\textsuperscript{a} C. Sinoquet\textsuperscript{b} R. Mourad\textsuperscript{b, c} G. Nuel\textsuperscript{a}

\textsuperscript{a}MAP5 – UMR CNRS 8145, Université Paris Descartes, Paris, \textsuperscript{b}LINA – UMR CNRS 6241, Université de Nantes, et \textsuperscript{c}Ecole Polytechnique de l’Université de Nantes, Nantes, France

Key Words
Statistical power · Disease model · Rejection · Markov chain Monte Carlo · MCMC · Backward sampling · Receiver operating characteristic · ROC · Area under the curve · AUC

Abstract
Objective: Assessing the statistical power to detect susceptibility variants plays a critical role in genome-wide association (GWA) studies both from the prospective and retrospective point of view. Power is empirically estimated by simulating phenotypes under a disease model H1. For this purpose, the gold standard consists in simulating genotypes given the phenotypes (e.g. Hapgen). We introduce here an alternative approach for simulating phenotypes under H1 that does not require generating new genotypes for each simulation.

Methods: In order to simulate phenotypes with a fixed total number of cases and under a given disease model, we suggest 3 algorithms: (1) a simple rejection algorithm; (2) a numerical Markov chain Monte-Carlo (MCMC) approach, and (3) an exact and efficient backward sampling algorithm. In our study, we validated the 3 algorithms both on a simulated dataset and by comparing them with Hapgen on a more realistic dataset. For an application, we then conducted a simulation study on a 1000 Genomes Project dataset consisting of 629 individuals (314 cases) and 8,048 SNPs from chromosome X. We arbitrarily defined an additive disease model with two susceptibility SNPs and an epistatic effect. Results: The 3 algorithms are consistent, but backward sampling is dramatically faster than the other two. Our approach also gives consistent results with Hapgen. Using our application data, we showed that our limited design requires a biological a priori to limit the investigated region. We also proved that epistatic effects can play a significant role even when simple marker statistics (e.g. trend) are used. We finally showed that the overall performance of a GWA study strongly depends on the prevalence of the disease: the larger the prevalence, the better the power. Conclusions: Our approach is a valid alternative to Hapgen-type methods; it is not only dramatically faster but has 2 main advantages: (1) there is no need for sophisticated genotype models (e.g. haplotype frequencies, or recombination rates), and (2) the choice of the disease model is completely unconstrained (number of SNPs involved, gene-environment interactions, hybrid genetic models, etc.). Our 3 algorithms are available in an R package called ‘waffect’ (‘double-u affect’, for weighted affectations).

Introduction
Genome-wide association (GWA) studies are a widely used approach for localizing susceptibility variants responsible for common complex genetic diseases [1]. Such studies involve investigating a huge number of genetic markers, such as single nucleotide polymorphisms (SNPs; from hundreds of thousands to millions), for cohorts of cases and controls whose sizes range from thousands to...
tens of thousands of individuals. GWA studies have met with many successes, most notably for type 1 and 2 diabetes, inflammatory bowel disease, prostate cancer and breast cancer [2].

In GWA studies, very high false positive rates are expected when there is no correction for multiple testing. Symmetrically, a control for the true negative rate – or power – is necessary. Power estimation is the key to evaluate the efficiency of GWA methods [3]. The correct estimation of both rates must take the existence of high-order dependencies between SNPs, or linkage disequilibrium (LD) into account. The accurate estimation of the family-wise type 1 error risk in the presence of LD consists in sampling the H0 distribution through permutations of phenotypes [4]. Thus, any association between loci and phenotypes is broken. This permutation strategy is implemented as a gold standard in numerous dedicated packages, together with software suites designed for GWA studies [5–9].

Power is an even more complicated function depending on several factors: study design, correlation patterns in the genotypic data, sizes of cohorts, frequency of the susceptibility allele, relative risk conferred by the causal factor, genetic model (additive, dominant, recessive, multiplicative) [10]. As a consequence, the analytical computation of power requires simplified assumptions, including the approximation of the test statistic distribution under H1 through a probability law [11]. Most power calculators based on analytical approaches are used for a two-stage GWA design [see for example 12, 13]. Recently, an analytical approach has been proposed to account for LD, under either H0 or H1 approximation [14]: a fixed-size sliding window locally accounts for the inter-marker correlation. This approach brings an improvement over block-wise strategies by unifying H0 and H1 processes in the same framework [15–17]. However, with regard to both accuracy and computational burden, the optimal choice of window size depends on the structure of the data. Moreover, LD blocks are often ambiguous. Thus, the previously proposed sliding window approach cannot account for high-order dependencies between LD blocks. In particular, this method cannot be used to evaluate the power of any novel approach designed to cope with such high-order dependencies in GWA studies. In this case, the only solution is to use computationally intensive simulations.

Similar to sampling under H0, the simulation of the H1 distribution is an appealing way to keep the LD-structured genotypic data. These simulations consist in generating case and control samples which mimic the LD structure in human genome, i.e. in the creation of, say, k datasets under the H1 assumption (at least one SNP is a susceptibility SNP). Nonetheless, breaking any association between a locus (or several loci) and the phenotype is far easier than to introduce such an association in a dataset.

Two main strategies have been developed to simulate H1. (1) The first one by Chadeau-Hyam et al. [18] consists in first generating a large sample of haplotypes conditional on reference haplotypes such as HapMap haplotypes [19]. The haplotypes are then paired to build diplotypes. The disease status is affected depending on the penetrance model which involves a susceptibility SNP selected at random. (2) The second strategy by Spencer et al. [3] consists in first randomly selecting a disease SNP and generating a fixed number of cases and controls. Then diplotypes are assigned to the disease SNP for cases and controls depending on the penetrance model. Finally, haplotypes are built (two for each diplotype) for all remaining SNPs of the chromosome, conditional on reference haplotypes. Nevertheless, both strategies entail problems of implementation when applied to power estimation in GWA studies. The drawback of the first strategy is that it does not allow controlling the number of cases and controls. To tackle this issue, rejection sampling of case-control samples can be used, but this leads to a waste of time and data. An illustration of the first strategy is Fregene [18]. The second strategy allows controlling these numbers by first fixing them, but it requires generating haplotypes for each simulation. The widely used simulator Hapgen [20] implements the second strategy.

To this aim, we propose a new method which randomly assigns $n_1$ cases and $n_0$ controls conditional on $n = n_0 + n_1$ genotypes and a given disease model. Our idea is to affect the status of individual $i$ according to both the probability $\pi_i$ that $i$ is a case and to the constraint that the total number of cases is fixed. The probability $\pi_i$ is computed from the disease model and is proportional to the relative risk for individual $i$. This method provides several advantages. First, it generalizes permutations, which is the gold standard for generating H0 distributions in order to control the type I error rate in multiple testing. Indeed, permutations represent a particular case of our general approach which is obtained by using the same $\pi_i$ for all individuals. Secondly, our method is faster than previous ones [3, 21] because it does not require a rejection sampling step or the generation of new genotypes for each simulation. Moreover, in the case of LD model-based mapping methods, LD pattern identification needs
to be performed only once. Thirdly, no assumption is made on the genotype distribution, because the latter is the same for each simulation. The last advantage is that power can be directly assessed using real GWA datasets, such as those provided by the 1000 Genomes Project [22], because disease status can be generated according to genotypes without loss due to rejection sampling. This is most valuable when evaluating the performance of new mapping methods.

In the next sections, we first describe the method itself, which we called waffle, then we validate our tool on a simulated toy example and compare its results to Hapgen [20] simulations. We finally illustrate the importance of our approach with a power study conducted on real GWA data.

Materials and Methods

Detailed Aims and Methods

We consider a GWA study with \( n = n_0 + n_1 \) individuals \((n_0 = \text{controls} \text{ and } n_1 = \text{cases})\), and \( p \) SNPs. The \( i \)-th phenotype is \( Y_i \in \{0, 1\} \) (\( Y_i = 1 \) if individual \( i \) is a case, \( Y_i = 0 \) if individual \( i \) is a control). The \( i \)-th genotype is \( X_i \in \{0, 1, 2\} \) (this corresponds to the number of rare alleles for each SNP).

Our disease model, hereafter denoted \( H_1 \), is defined by the probability \( \pi_i = P(Y_i = 1 \mid X_i) \) that individual \( i \) is a case conditionally on her/his genotype. In the particular case when all \( \pi_i \) have the same non-negative value, the observed genotype has no effect on the phenotype. This case corresponds to the null hypothesis \( H_0 \). In the particular case when we consider a single SNP \((p = 1)\), \( \pi_i \) can take only 3 possible values: \( \pi_i = f_0 \) if \( X_i = 0 \), \( \pi_i = f_1 = f_0 RR_1 \) if \( X_i = 1 \), and \( \pi_i = f_2 = f_0 RR_2 \) if \( X_i = 2 \), where \( f_0, f_1 \) and \( f_2 \) are the penetrances of the disease model, and \( RR_1 \) and \( RR_2 \) are the relative risks.

It is obviously easy to simulate under \( H_1 \) using the independent Bernoulli distribution \( B(\pi_i) \) for each individual. Unfortunately, it is quite unlikely that when doing so the case-control design constraint \( C = \{\sum_{i=1}^{n_1} Y_i = n_1\} \) will be fulfilled.

For this reason, the standard strategy (i.e. Hapgen) consists in simulating genotypes conditional on phenotypes using Bayes’ formula:

\[
P(X \mid Y) = \frac{P(Y \mid X)P(X)}{\sum \Delta P(Y \mid X)P(X)} = \pi_i P(X_i) / P(Y_i).
\]

The advantage of this approach is that constraint \( C \) is automatically fulfilled, whereas the major drawback is that a genotype distribution \( P(X) \) must be introduced. In order to be realistic, this imposes sophisticated modeling using haplotype data and recombination rates. As an alternative, we suggest several ways to simulate phenotypes under \( H_1 \) conditional on the genotypes (which therefore remain untouched) while respecting constraint \( C \). To do so, we introduce the notation \( Z_j = \sum_{i=1}^{n_1} Y_i \) (with \( Z_0 = 0 \) by convention) so that the constraint becomes \( C = \{Z_0 = n_1\} \).

Reject

The first approach is straightforward: (1) draw \( Y_j \) \( j = 1, \ldots, n \) using independent Bernoulli distributions; (2) if constraint \( C \) is fulfilled, retain \( Y_j \) \( j = 1, \ldots, n \) as a valid sample; if not, simply reject it and return to (1). The problem with this algorithm is that samples will be rejected at rate \( 1 - P(C) \). The natural question is then: how (un)likely is event \( C \)?

\[ P(C) \] can be computed recursively by introducing the forward quantities: \( F_i(m) = P(Z_n = m) \). Starting with \( F_0(m) \) being equal to 0, except for \( F_0(0) = 1 \), we get the following recursive formula for \( 1 \leq i \leq n \):

\[ F_i(m) = F_{i-1}(m-1)\pi_i + (1-\pi_i). \]

Markov Chain Monte Carlo

In order to overcome this drawback, an interesting idea is to turn to Markov chain Monte Carlo (MCMC) techniques such as the Metropolis-Hastings class of sampling algorithms [23]. Starting from a configuration of phenotypes fulfilling the constraint (e.g. with the observed phenotypes), alternate the following 2 steps: (1) proposal move: select 2 individuals \( i \) and \( j \) such that \( Y_i = 1 \) and \( Y_j = 0 \) and then exchange their phenotypes \((Y_i = 0 \text{ and } Y_j = 1)\); (2) accept this move with rate \( \alpha \) given by:

\[ \alpha = \frac{(1-\pi_j)\pi_i}{\pi_j(1-\pi_i)}. \]

It can be shown that the sequence of phenotype configurations generated by this algorithm is a Markov chain whose stationary distribution is our targeted constrained \( H_1 \) distribution. In practice, the first configurations which are generated must be discarded since the Markov chain is not yet stationary (this stage is called burn-in). In order to obtain a set of independent configurations under \( H_1 \), one can either repeat the burn-in phase of the algorithm as many times as necessary, or use a sequence of pseudo-independent samples by picking phenotype configurations after the burn-in phase once every fixed number of iterations.

The main advantage of the MCMC approach is that it will eventually work regardless of the probability of event \( C \). Its drawback is that the number of iterations for the burn-in phase and pseudo-independence necessary for a good mixing increases at least linearly with the total number \( n \) of individuals. Moreover, the control for stationarity and pseudo-independence is delicate and the practical choice of burn-in and independence thresholds usually requires tedious calibration work.

Backward Sampling

Our proposed alternative is to turn to exact sampling by introducing the backward quantities \( B_i(m) = P(C \mid Z_n = m) \) which can be computed recursively like the forward quantities defined above. Starting from \( B_0(m) \) being equal to 0, except for \( B_0(n_1) = 1 \), we get the following recursive formula for \( 1 \leq i \leq n \):

\[ B_i(m) = \pi_i B_i(m+1) + (1-\pi_i)B_i(m). \]
One should note that we obtain $P(C) = B_0(0)$ in the process.

Then we can sample phenotypes $Y_i$ sequentially with the following probability:

$$P(Y_i = 1|Z_{i-1} = m, C) = \frac{\pi B_i(m + 1)}{B_{i-1}(m)}.$$  \hspace{1em} (3)

Since this expression depends on $Z_{i-1}$, the corresponding probability obviously depends on the number of cases seen by $i - 1$. In mathematical terms, the sequence $(Y_i, Z_i)_{i=1, \ldots, n}$ is a heterogeneous Markov chain.

**H0 Simulations**

H0 simulations are simply performed by permuting the phenotypes $Y_i$. A simple way to do this is to uniformly choose a permutation $\sigma$ of $[1, \ldots, n]$ by performing a $O(n \log n)$ quicksort algorithm on a sample of $n$ independent random variables with a uniform distribution on $[0, 1]$. For example, this can be achieved through the ’sample’ command in R [24]. Alternatively, one may use waffect with the same probability $\pi_i$ for all individuals.

**Simulated Example Dataset**

For validation purposes, we first considered a simulated toy example dataset with $p = 1$ SNP. The $n$ genotypes $X_i$ have the following distribution: 80% have genotype 0, 15% genotype 1, and 5% genotype 2. With $n$ being a multiple of 20, these counts are always integers. For our disease model, we considered the following relative risks: $RR_1 = 1.5$ and $RR_2 = 2.0$ (additive effect of 0.5). We hence obtained the following penetrances: $f_0$, $f_1 = 1.5f_0$, and $f_2 = 2.0f_0$. In order to explore the relevance and limits of each of our 3 approaches, we considered a wide range of values for $n$ and $f_0$.

**Hapgen Simulations**

In contrast to simulations under H1, simulations under H0 are expected to provide few statistically significant results. This contrast can be observed through a binary (or two-class) classifier system. Regarding this aspect, we compared the behavior of waffect with one of the most widely used simulators Hapgen [25] under various conditions. The comparison relied on the so-called receiver operating characteristic (ROC) curve, which is a graphical plot of the true positive classification rate (sensitivity) as a function of the false positive rate (specificity), for a continuum of values of a given discrimination criterion. The performance of each simulator is summarized using the area under the ROC curve, which is a graph of the false positive rate (specificity), for a continuum of values of a given discrimination criterion. The performance of each simulator is summarized using the area under the ROC curve, which is a graph of the false positive rate (specificity), for a continuum of values of a given discrimination criterion. The performance of each simulator is summarized using the area under the ROC curve, which is a graph of the false positive rate (specificity), for a continuum of values of a given discrimination criterion.

**1000 Genomes Project Dataset**

The dataset consists of the genotypes of $n = 629$ individuals ($n_0 = 315$ controls, $n_1 = 314$ cases) from the 1000 Genomes Project [22]. We focused on the first 100,000 SNPs on chromosome X. In the pretreatment stage, we filtered out all SNPs with a MAF less than or equal to 5%, ending up with $n = 8,048$ SNPs. We arbitrarily considered a disease model with 2 interacting disease SNPs. The 2 SNPs respectively have the basepair positions 627,641 and 1,986,325, the MAFs 0.26 and 0.23, and display no LD ($r^2 = 0.02$). We considered an additive effect $\beta > 0$ and an epistatic effect $\eta$ such that:

$$\pi_i = f_0 \times \begin{cases} 
1.0 + \beta X_1^i & \text{if } X_1^i = 0 \\
1.0 + \beta X_2^i & \text{if } X_1^i = 0 \\
1.0 + \eta + \beta(X_1^i + X_2^i) & \text{if } X_1^i X_2^i = 0
\end{cases}$$

where $X_1^i$ and $X_2^i$ correspond to the genotype of individual $i$ at the 2 susceptibility SNPs.

**Statistics of Association**

The association signal of each SNP (single-marker analysis) is computed through PLINK software [9] using the trend statistic. We denote $p_j$ the p value of SNP $j$. In order to avoid the issue of multilisting, we considered the following single real valued statistic:

$$S_j = \max_{p_j \in \mathcal{J}_p} - \log_{10} p_j$$

where $\mathcal{J}_p$ denotes the subset of SNPs such that the distance between their loci and the distance of a disease SNP is less than a radius $\rho$.

For example, $S_{\max}$ corresponds to taking the best $p$ value $p_j$ in negative (decimal) log-scale on the whole dataset. With $\rho = 5$ kb, we consider the best $p$ value of the SNPs within 5 kb of one of the 2 susceptibility SNPs. Including the radius into the statistic is somewhat unusual, since it takes advantage of information which is rarely available in practice: the location of the disease SNPs. However, in our simulation framework, this information helped us discuss the power of the design depending on the priori location of the 1000 Genomes Project.
susceptibility SNPs. From a general point of view, starting with no known location leads to a choice of $\rho = \infty$; after a successful linkage analysis, $\rho = 100$ kb might be more relevant, and if a candidate gene has been identified, $\rho = 5$ kb could be a good choice.

**ROC Curves and AUC**

In order to measure the performance of a given statistic for a given design and $H_1$ model, it is natural to compare the sensitivity and specificity of the testing framework for a given threshold of significance. For example, a global threshold of 5% is commonly accepted as a reasonable choice. It might be interesting to avoid making this arbitrary choice by studying instead the ROC curve which is nothing but a graphical representation of the specificities and sensitivities that can be obtained for all possible values of the threshold of significance [26].

The ROC curve itself can then be further summarized by the AUC which can be qualitatively interpreted as follows: $AUC \leq 0.6$ means 'fail'; $0.6 < AUC \leq 0.70$ means 'poor'; $0.7 < AUC \leq 0.80$ means 'fair'; $0.8 < AUC \leq 0.9$ means 'good'; $0.9 < AUC \leq 1.0$ means 'excellent'. The AUC can easily be estimated from 2 samples of the statistic: one sample under $H_0$, and the other under $H_1$. All ROC curves and AUC computations (including confidence intervals) were performed using the R package pROC [27].

**Results**

**Validation**

**Simulated Toy Example Dataset**

In table 1, the execution times for generating 100 replicates through the rejection, MCMC and backward sampling algorithms on the simulated toy example dataset are shown, together with the corresponding probabilities $P(C)$ of randomly drawing a complete configuration of phenotypes with exactly $n_1$ cases. Different values of $n$ and $f_0$ are considered. $P(C)$ dramatically decreases when $n$ grows and $f_0$ becomes smaller. As a consequence, the running time of the rejection algorithm is already important (more than 10 h) even for $n$ as small as 40 and $f_0$ as large as 0.2. The rejection algorithm was not run for greater values of $n$ as the corresponding probability of a successful finding is negligible. The MCMC and backward sampling algorithms have short running times which grow with $n$ (very gently for the backward sampling algorithm), and, most importantly, which do not depend on $P(C)$. For the MCMC algorithm, we considered a burn-in phase of $10^5 \times n$.

The AUC and the corresponding 95% confidence intervals computed on the simulated toy example dataset replicates are shown in table 2. The values found with the 3 methods are consistent.

| $n$ | $f_0$ | $P(C)$ | Rejection | MCMC | Backward |
|-----|------|--------|-----------|------|----------|
| 20  | 0.2  | $4.5 \times 10^{-3}$ | 0.4 s   | 38 s | 0.05 s   |
| 20  | 0.1  | $1.7 \times 10^{-5}$ | 1.5 min | 38 s | 0.05 s   |
| 20  | 0.07 | $6.7 \times 10^{-7}$ | 38.5 min | 38 s | 0.05 s   |
| 20  | 0.05 | $2.9 \times 10^{-8}$ | 11.2 h  | 38 s | 0.1 s    |
| 40  | 0.2  | $8.2 \times 10^{-5}$ | 17.4 s  | 1.2 min | 0.1 s   |
| 100 | 0.2  | $8.7 \times 10^{-10}$ | NA     | 3.2 min | 0.2 s   |
| 100 | 0.1  | $5.8 \times 10^{-22}$ | NA     | 3.2 min | 0.2 s   |
| 100 | 0.01 | $1.1 \times 10^{-69}$ | NA     | 3.2 min | 0.2 s   |

| $n$ | $f_0$ | AUC [95% confidence interval] | Rejection | MCMC | Backward |
|-----|------|-------------------------------|-----------|------|----------|
| 20  | 0.2  | $0.60 [0.53, 0.67]$           | 0.59 [0.52, 0.66] | 0.61 [0.54, 0.68] |
| 20  | 0.1  | $0.59 [0.52, 0.66]$           | 0.59 [0.52, 0.66] | 0.58 [0.51, 0.65] |
| 20  | 0.07 | $0.62 [0.55, 0.69]$           | 0.56 [0.49, 0.63] | 0.56 [0.49, 0.63] |
| 20  | 0.05 | $0.44 [0.37, 0.51]$           | 0.58 [0.51, 0.65] | 0.53 [0.47, 0.60] |
| 40  | 0.2  | $0.58 [0.50, 0.65]$           | 0.62 [0.54, 0.69] | 0.59 [0.52, 0.67] |
| 100 | 0.2  | NA                            | 0.75 [0.68, 0.81] | 0.72 [0.65, 0.79] |
| 100 | 0.1  | NA                            | 0.64 [0.57, 0.72] | 0.68 [0.61, 0.75] |
| 100 | 0.01 | NA                            | 0.65 [0.58, 0.73] | 0.59 [0.51, 0.67] |
128 GB), the generation and processing of 1,000 replicates required approximately 75 min using the backward sampling algorithm (under H1 or H0) or Hapgen under H0. In contrast, under H1 with Hapgen, approximately 110 h were necessary to generate and process data of the same dimension. More comprehensive comparisons with Hapgen are available in the online supplementary material.

Application

Power Study on the 1000 Genomes Project Dataset

The ROC curves of the statistics $S_\rho$ on the 1000 Genomes Project dataset for different values of the radius $\rho$ are displayed in figure 2. The AUC is very low (95% confidence interval = [0.41, 0.57] when considering all SNPs ($\rho = \infty$). The AUC increases as the region around the 2 susceptibility SNPs being investigated gets smaller. In particular, the performance is good for $\rho \leq 5$ kb. This roughly corresponds to the size of a gene. These results show that given the design and the disease model under consideration, the statistic which consists in simply taking the best $p$ value is successful in detecting a positive signal only in the presence of a biological a priori; thereby narrowing the investigation to the candidate gene level.

We fixed a candidate region corresponding to the radius $\rho = 5$ kb and investigated the dependency of the AUC on the following parameters: epistasis $\eta$, additive effect $\beta$, probability $f_0$ of being a case in the absence of rare alleles, and total number $n$ of individuals. Not unexpectedly, the AUC grows with the additive effect $\beta$ (table 3). More surprisingly, it also grows when the epistatic effect increases (table 4), despite the fact that our statistic $S_\rho$ only uses simple marker statistics.

We then focused on the design of the GWA study itself by studying the effect of the total number $n$ of individuals on the AUC. This was simply done by doubling (tripling, quadrupling) the original dataset before applying our disease model. In the case of our configuration of reference $\eta = 0.3$, $\beta = 0.3$, $f_0 = 0.1$ and $\rho = 5$ kb, we observed that the performance is excellent when the population is twice as large as the original one (AUC = 0.94 [0.90, 0.97] for $n = 1,258$; AUC = 0.80 [0.74, 0.86] for $n = 629$). This gain in power is even more dramatic when all SNPs are taken into account ($\rho = \infty$): the AUC doubles in value when passing from 629 to 1,258 individuals (see table 5).

![Fig. 1. Comparison of the AUC for Hapgen and wffect – 1,000 replicates under H1 and H0, respectively, for MAF in [0.2, 0.3], additive model, RR_1 = 1.6, RR_2 = 2.2. The 95% confidence intervals are indicated.](image1)

![Fig. 2. ROC curves and AUC for different values of the radius $\rho$ of the intervals centered at the 2 disease SNPs.](image2)
Finally, we investigated the effect of \( f_0 \), the probability of being affected without any exposure, in the design of the GWA study (note that for small values, \( f_0 \) is close to the prevalence). We can see in table 6 that small values of \( f_0 \) result in smaller AUCs (i.e. AUC = 0.73 [0.66, 0.80] for \( f_0 = 0.001 \)), while large values of \( f_0 \) result in much better results (e.g. AUC = 0.94 [0.83, 0.93] for \( f_0 = 0.25 \)).

**Discussion**

**Validation**

The 3 algorithms we suggested give similar results but their performances are quite different. The rejection algorithm has a complexity of \( O(N/P(C)) \) to obtain \( N \) samples. This prevents its use in practical cases when \( P(C) \) can easily reach \( 10^{-50} \) or worse. The MCMC approach has the advantage of removing this dependence and is simple to implement. The drawback is that the burn-in parameter requires some calibration work (we suggest to use burn-in = \( 10^3 n \)). Moreover this heavy numerical method is rather slow. In our study, the backward sampling approach proved to be the fastest one, dramatically outperforming the MCMC alternative (i.e. 500 times faster).

Comparison with Hapgen

Our comparison with Hapgen clearly demonstrates that our approach is a valid alternative to this software. It also sheds light on the major differences between the 2 approaches. First, in order to yield a realistic model for simulating genotypes, Hapgen requires additional information such as HapMap frequencies and recombination rates, whereas our approach only needs the original genotypes. Regarding our simulations performed with Hapgen and wffect, running the latest Hapgen version would have entailed no change in the results: the newest Hapgen version is an extension of the former; it is not an improvement (at least regarding single susceptibility SNP simulation). Indeed, in its initial version, Hapgen limits the disease model to only 1 susceptibility SNP. The most recent version can now simulate multiple disease SNPs,
under the assumption that each disease SNP acts independently and that all such SNPs are in Hardy-Weinberg equilibrium, which are 2 strong constraints. Besides, this novel software, unfortunately only available in the 64-bit version, requires the use of an R package to simulate interactions between the disease SNPs. In contrast, simulating any disease model is straightforward with our approach, as long as it results in an individual probability for each individual to be affected. For example, we can consider two or more susceptibility SNPs, and easily add epistatic effects, gene-environment interactions, the contribution of rare variants, etc. Moreover, due to the multiple constraints it must take into account, Happen only respects the specified model on average. As a consequence, the effective MAF and relative risks can vary from one run to another. Besides, selecting for each Happen replicate a different disease SNP entails variations in the number of SNPs available for the association test (see the online suppl. material for more details). Finally, since our approach only generates a fraction of the data (the phenotypes), it is outstandingly faster than Happen. Furthermore, since with our approach the genotype remains untouched, any numerically intensive analysis which is performed on the genotype table (e.g. LD computations) needs to be performed only once, while with Happen it has to be done for each replicate.

Effects
In our application study, we showed how our strategy can be used to investigate various effects on the performance of a GWA study. We first pointed out that with the suggested design, our modest disease model ($\beta = \eta = 0.3$, $f_0 = 0.1$), and the analysis we chose (simple marker using PLINK), the signal is only statistically detectable by limiting the region to be investigated to a 5 kb radius around the susceptibility SNPs. This clearly shows that the present design is practical only if previous work (e.g. linkage analysis) suggests candidate genes.

Then we proved that the AUC increases when either the additive effect ($\beta$) or the epistatic effect ($\eta$) increases. While the result for the additive effect is not surprising, the result for the epistatic effect is worth commenting upon. Indeed, in our application, we deliberately stuck to a classical simple marker analysis (PLINK using the trend test) which is not designed to detect epistatic effects. However, our study showed that the corresponding marginal effects can improve the overall statistical performance of the analysis, even when no specific effort for the detection of epistasis is performed. It would be interesting to do further research to compare simple marker analysis to more complex analysis especially designed to take epistatic effects into account.

Design
Finally, we briefly investigated the influence of the design of the GWA study on its performance. As could be expected, the total number $n$ of individuals plays a critical role. It might be interesting, however, to refine this analysis by also considering different ratios $n_c/n_h$. (This is left for further investigation.) We also showed that for a fixed design (fixed number of cases and controls), the prevalence of the disease (through $f_0$, which is closely related) has an unexpected effect on the performance: higher prevalence gives better results. More investigations might be necessary to understand the reason behind this observation.

Extensions
One should note that our approach can be extended in several directions. First, we can easily consider more than 2 classes, thus complexifying constraint $C$. In this case, the rejection and MCMC algorithms remain exactly the same. For backward sampling, one should first affect one class against the others, and then perform the affectation recursively on the remaining unaffected classes. The resulting complexity is $O(n \times (K – 1))$ for the affectation of $n$ individuals into $K$ classes. As explained above, we can also consider sophisticated genetic models that take additional covariates into account such as environmental exposure or rare variants. Finally, the extension of our approach to other fields is quite straightforward. For example, in the analysis of gene expression, our approach might be used to affect sample status (e.g. cancer or healthy) conditional on the level of expression rather than generating expressions through parametric or non-parametric questionable models as is usually done.

Conclusions
In this paper, we present an alternative to classical simulations under $H_1$ in GWA studies. The idea is to generate the phenotypes conditional on the existing genotypes with respect to the study design (number of cases). For that purpose, we suggest 3 algorithms including the backward sampling which mimics hidden Markov models to provide a fast sampling conditional on the constraint. Our study shows that our algorithms are valid and that their results are consistent with reference software such as Happen. Moreover, our approach has many advantages: it is much faster; it does not need any genotyping mod-
el (the genotypes remain untouched); it allows considering any complex genetic model (including several SNPs, epistasis, covariates, rare variants, etc.). Our approaches are available in an R package called \textit{waffect} ('double-u affect', for weighted affectations), which is available on the CRAN package repository http://cran.r-project.org/.

Acknowledgements

G.N. suggested the original idea and the backward sampling algorithm. G.N. and V.P. implemented the R package. R.M. and C.S. wrote the state-of-the-art on the subject in the Introduction. C.S. took in charge the comparison with Hapgen and the editing of the corresponding supplementary material. V.P. performed all the other simulations and edited the Appendix. The editing of the paper was done by G.N. and V.P. All authors read and approved the final manuscript.

Appendix

In this section, we provide proofs of the mathematical results from the Materials and Methods section, namely of equations 1–3. Recall that $Z_i = \sum_{j=1}^{n_i} Y_j$ is the number of cases among individuals 1, ..., $i$ and that we deal with constraint $C = [Z_n = n_i]$. The forward and backward quantities are the probabilities $F_i(m) = P(Z_i = m)$ and $B_i(m) = P(C | Z_i = m)$. Let $V$ be the set of variables $\{Y_1, ..., Y_n, Z_1, ..., Z_n\}$.

The key property is that the conditional dependencies among all the variables determine the following factorization of the joint probability distribution

$$P(V) = \prod_{i \in I} P(Z_i | Z_{i-1}, Y_i) P(Y_i),$$

where $I = \{1, ..., n\}$ is the set of all the individuals and $Z_0 = 0$ by convention.

\textbf{Theorem 1}

For each individual $i = 1, ..., n$:

$$P(Z_i = m, C) = F_i(m) B_i(m).$$

Moreover

$$P(Y_i = 1, Z_i = m + 1, Z_{i-1} = m, C) = F_{i-1}(m) \pi_i B_i(m + 1),$$

$$P(Y_i = 0, Z_i = m, Z_{i-1} = m, C) = F_{i-1}(m)(1 - \pi_i) B_i(m).$$

\textbf{Proof}

We will only prove equation 5: similar arguments hold for equations 6 and 7. The marginal probability $P(Z_i = m, C)$ is obtained from the joint probability distribution $P(V)$ multiplying it by the indicator function $1_C$ (by definition $1_C = 1$ if and only if $C$ is true) and summing out all the variables in $V \setminus \{Z_i\} = \{Z_1, ..., Z_{i-1}, Z_{i+1}, ..., Z_n, Y_1, ..., Y_n\}$. Because of equation 4, we have

$$P(Z_i = m, C) = \sum_{Y_i} 1_C \prod_{j \in I} Q_j(Z_j, Z_{j-1}, Y_j),$$

where for convenience we denote $Q_j(Z_j, Z_{j-1}, Y_j) = P(Z_j | Z_{j-1}, Y_j) P(Y_j)$. Now consider the sets

$$V_{ij} = \{Z_i, Y_1, ..., Z_n\},$$

$$V_{(i+1)n} = \{Z_{i+1}, Y_{i+1}, ..., Z_n\}.$$

$V_{ij} \cup V_{(i+1)n}$ is a partition of $V$ and $V \setminus \{Z_i\} = V_{ij} - \{Z_i\} \cup V_{(i+1)n}$. Note that equation 8 is equal to

$$\sum_{V_{ij} \cup V_{(i+1)n}} \prod_{j \in I} Q_j \left( \sum_{V_{ij} \cup V_{(i+1)n}} \prod_{j \in I} Q_j \right) =$$

$$\sum_{V_{ij} \cup V_{(i+1)n}} \prod_{j \in I} Q_j \left( \sum_{V_{ij} \cup V_{(i+1)n}} \prod_{j \in I} Q_j \right),$$

where the last equality holds because the only variable shared between the 2 factors is $Z_i$ which is set to be $Z_i = m$ from the beginning. It is now easy to see that the 2 factors in the product above are $F_i(m)$ and $B_i(m)$, respectively.

\textbf{Corollary 2}

Equations 6 and 7 allow computing the forward and backward quantities recursively:

$$F_i(m) = F_{i-1}(m-1) \pi_i + F_{i-1}(m)(1 - \pi_i),$$

$$B_{i-1}(m) = \pi_i B_i(m + 1) + (1 - \pi_i) B_i(m).$$

\textbf{Proof}

We will prove only the recursive formula for $B_i$, a similar argument holds for $F_i$. Note that

$$P(Z_i = m, C) = F_i(m) \pi_i B_i(m) + F_{i-1}(m)(1 - \pi_i) B_i(m).$$

By dividing by $F_i(m)$, we obtain the iterative formula for the backward quantities.

\textbf{Corollary 3}

We can now prove equation 3:

$$P(Y_i = 1 | Z_{i-1} = m, C) = \frac{\pi_i B_i(m + 1)}{B_{i-1}(m)},$$

for each individual $i = 1, ..., n$.

\textbf{Proof}

Apply the definition of conditional probability and divide equation 6 by equation 5.

\textbf{Corollary 4}

$$P(Y_i = 1 | C) = \sum_m F_i(m) \pi_i B_i(m + 1),$$

$$P(Y_i = 0 | C) = \sum_m F_{i-1}(m)(1 - \pi_i) B_i(m).$$
References

1. Morris AP, Cardon LR: Handbook of Statistical Genetics, ed 3, vol 2. Wiley Interscience, 2007, pp 1238–1263.

2. Hindorff LA, Sethupathy P, Junkins HA, Ramos EM, Mehta JP, Collins FS, Manolio TA: Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. Proc Natl Acad Sci USA 2009;106:9362–9367.

3. Spencer CC, Su Z, Donnelly P, Marchini J: Designing genome-wide association studies: sample size, power, imputation, and the choice of genotyping chip. PLoS Genet 2009;5:e1000477.

4. Zhang Q, Ott J: Handbook on Analyzing Human Genetic Data. Heidelberg, Berlin: Springer, 2010, pp 277–287.

5. Aulchenko YS, Ripke S, Isaacs A, van Duijn CM: Genabel: an R library for genome-wide association analysis. Bioinformatics 2007;23:1294–1296.

6. Browning BL: PRESTO: rapid calculation of order statistic distributions and multiple-testing adjusted p-values via permutation for one and two-stage genetic association studies. BMC Bioinformatics 2008;9:309.

7. Gonzalez JR, Armengol L, Sole X, Guino E, Mercader JM, Estivill X, Moreno V: SNPassoc: an R package to perform whole genome association studies. Bioinformatics 2007;23:644–645.

8. Pollard KS, Dudoit S, van der Laan MJ: Multiple testing procedures: the multtest package and applications to genomics; in Gentleman RC, Carey VJ, Huber W, Irizarry R, Dudoit S (eds): Bioinformatics and Computational Biology Solutions Using R and Bioconductor. New York: Springer, 2005. Bioconductor R package.

9. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, Maller J, Sklar P, de Bakker PIW, Daly MJ, Sham PC: PLINK: a toolset for whole-genome association and population-based linkage analyses. Am J Hum Genet 2007;81:559–575.

10. Lettre G, Lange C, Hirschhorn JN: Genetic model testing and statistical power in population-based association studies of quantitative traits. Genet Epidemiol 2007;31:358–362.

11. Klein RJ: Power analysis for genome-wide association studies. BMC Genet 2007;8:58.

12. Menashe I, Rosenberg PS, Chen BE: PGA: power calculator for case-control genetic association analyses. BMC Genet 2008;9:36.

13. Steibel JP, Abecasis GR: QpowR: interactive power calculator for two-stage genetic association studies of quantitative traits. https://www.msu.edu/~steibelj/JP_files/QpowR.pdf, 2008.

14. Han B, Kang HM, Eskin E: Rapid and accurate multiple testing correction and power estimation for millions of correlated markers. PLoS Genet 2009;5:e1000456.

15. Conneely KN, Boehnke M: So many correlated tests, so little time! Rapid adjustment of p-values for multiple correlated tests. Am J Hum Genet 2007;81:1158–1168.

16. Lin D: An efficient Monte Carlo approach to assessing statistical significance in genomic studies. Bioinformatics 2005;21:781–787.

17. Seaman SR, Müller-Myhskov B: Rapid simulation of p values for product methods and multiple-testing adjustments in association studies. Am J Hum Genet 2005;76:399–408.

18. Chadeau-Hyam M, Hoggart CJ, O’Reilly PF, Whittaker JC, De Iorio M, Balding DJ: Frege: simulation of realistic sequence-level data in populations and ascertained samples. BMC Bioinformatics 2008;9:364.

19. The International HapMap Consortium: A second generation human haplotype map of over 3.1 million SNPs. Nature 2007;449:851–861.

20. Su Z, Marchini J, Donnelly P: Hapgen, version 2, 2010.

21. Peng B, Amos CI: Forward-time simulation of realistic samples for genome-wide association studies. BMC Bioinformatics 2010;11:442.

22. The 1000 Genomes Project Consortium: A map of human genome variation from population-scale sequencing. Nature 2010;467:1061–1073.

23. Gilks WR, Richardson S, Spiegelhalter DJ: Markov Chain Monte Carlo in Practice: Introducing Markov chain in Monte Carlo, pp 1–20, Chapman and Hall, 1996.

24. R Development Core Team: R: A Language and Environment for Statistical Computing. Vienna, R Foundation for Statistical Computing, 2011. ISBN 3–900051–07–0.

25. Marchini J, Howie B, Myers S, McVean G, Donnelly P: A new multipoint method for genome-wide association studies via imputation of genotypes. Nat Genet 2007;39:906–913.

26. Metz CE: Basic principles of ROC analysis. Sem Nuc Med 1978;8:283–298.

27. Robin X, Turck N, Hainard A, Tiberti N, Lisacek F, Sanchez JC, Müller M: pROC: an open-source package for R and S+ to analyze and compare ROC curves. BMC Bioinformatics 2011;12:77.