Published in final edited form as:
Ann Hum Genet. 2010 January ; 74(1): 88–96. doi:10.1111/j.1469-1809.2009.00548.x.

Variable selection method for quantitative trait analysis based on parallel genetic algorithm

Siuli Mukhopadhyay[1], **Varghese George**[2], and **Hongyan Xu**[2]

[1]Department of Mathematics, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India

[2]Department of Biostatistics, Medical College of Georgia, 1469 Laney Walker Blvd. Augusta, Georgia 30912, U.S.A.

Summary

Selection of important genetic and environmental factors is of strong interest in quantitative trait analyses. In this study, we use parallel genetic algorithm (PGA) to identify genetic and environmental factors in genetic association studies of complex human diseases. Our method can take account of both multiple markers across the genome and environmental factors, and also can be used to do fine mappings based on the results of haplotype analysis to select the markers that are associated with the quantitative traits. Using both simulated and real examples, we show that PGA is able to choose the variables correctly and is also an easy-to-use variable selection tool.

Keywords

QTL; parallel genetic algorithm; complex traits

INTRODUCTION

Most of the traits in complex human diseases are quantitative in nature. Generally, quantitative traits are influenced by joint effects of multiple genetic and environmental factors. The genetic basis of most of these complex traits is unknown and there is strong interest in genetic analysis of the quantitative traits. It is the purpose of these analyses to identify the relevant genetic and environmental factors, which are important for understanding the etiology and treatment of these diseases.

For identifying the genetic factors, it is of interest to know both the number and location of the underlying genes, or quantitative trait loci (QTLs). A common approach to identifying the QTLs involves testing for a set of genetic markers serially by testing a single genetic marker each time, using approaches such as t-test or ANOVA. An alternative approach is interval mapping based on the regression of flanking markers (Haley & Knott, 1992; Lander & Botstein, 1989). Both of these approaches are built on a single-QTL model, which can bias the marker selection and estimates of its genetic effects when the trait is actually controlled by multiple loci. For complex quantitative traits controlled by QTLs, it is necessary to take an approach based on a multiple-loci model. One effective approach is to consider this as a variable (or model) selection problem in the linear multiple-regression framework. Several variable selection methods have been developed for genetic analysis of quantitative traits, mostly using Bayesian approaches, in animal breeding settings (Ball,

Corresponding author: Hongyan Xu Ph.D., Department of Biostatistics, Medical College of Georgia, 1469 Laney Walker Blvd., Augusta, GA 30912, Tel: (706)721-3785, Fax: (706)721-6294, hxu@mcg.edu.

2001; Broman & Speed, 2002; Piepho & Gauch, 2001; Yi et al., 2003). However, the performance of these methods could be dependant on the choice of prior distributions.

In this study, we propose an alternative variable selection method for quantitative trait analysis based on parallel genetic algorithm (PGA). For most complex quantitative traits in human, it is generally assumed that there are a few loci with large effects and many with small effects. Therefore, it is expected that only the markers linked to the QTLs with large effect will be selected in the final models. Our method can take account of both multiple markers across the genome and environmental factors.

A genetic algorithm (GA) is an optimisation procedure used to solve combinatorial optimisation problems. In this paper, we frame the problem of selecting genetic and environmental factors as a combinatorial optimisation problem, and apply a parallel version of the GA to obtain the optimal set of factors. We illustrate the performance of PGA in selecting the best set of genetic and environmental factors for quantitative traits using both simulated and real data. We also compare PGA with the well-known Bayesian variable selection procedure, more popularly known as stochastic search variable selection (SSVS) (George and McCulloch, 1993).

MATERIAL AND METHODS

Modelling and estimation

Given a disease phenotype for a continuously distributed trait, *y*, and a set of potential genetic and environmental factors, we are interested in finding the subset of factors which actually affect *y*. We assume that the relationship between y_i and the set of genetic and environmental factors can be modeled as follows,

$$y_i=\beta_0+\sum_{j=1}^{p_g}\beta_j x_{ij}+\sum_{k=1}^{p_e}\delta_k z_{ik}+\epsilon_i,\ i=1,\ldots,n, \tag{1}$$

where x_{ij} denotes the *j*th ($j = 1,\ldots,p_g$) genetic factor for the *i*th individual and z_{ik} is the *k*th ($k = 1,\ldots,p_e$) environmental factor for the *i*th individual and $\epsilon_i \sim N(0; 1)$ are the identically and independently distributed errors. Thus, p_g and p_e are the number of genetic factors and environmental factors, respectively. Here, β_0 is the intercept term, β_j's are coefficients corresponding to the *j*th genetic factor, and δ_k's are coefficients corresponding to the *k*th environmental factor. If we assume that A and a represents the major and the minor allele respectively with A dominant to a with complete penetrance, then

$$x_{ij}=\begin{cases}1 & \text{if the genotype of the } i\text{th individual is either AA or Aa}\\ 0 & \text{if the genotype is aa}\end{cases} \tag{2}$$

Thus, x_{ij} is a binary variable taking two values, 1 and 0, while z_{ik} is continuous. Here we assume complete penetrance for the simplicity of the model without losing generality. When there is incomplete penetrance, the frequency of x_{ij} taking value 0 becomes the product of a penetrance parameter and frequency of genotype aa. The predicted response is

$$\widehat{y_i}=\widehat{\beta_0}+\sum_{j=1}^{p_g}\widehat{\beta_j} x_{ij}+\sum_{k=1}^{p_e}\widehat{\delta_k} z_{ik},\ i=1,\ldots,n, \tag{3}$$

where $\hat{\beta}_0$, $\hat{\beta}_j$, and $\hat{\delta}_j$ are the least square estimates of β_0, β_j and δ_j, respectively. Writing in matrix notations,

$$\widehat{\mathbf{y}}=\widehat{\beta_0}+\mathbf{X}\widehat{\theta}, \tag{4}$$

where $\hat{\mathbf{y}}_{n\times 1} = (\hat{y}_1,\ldots,\hat{y}_n)^T$, $\hat{\boldsymbol{\theta}}_{p\times 1} = (\beta\hat{}_1,\ldots,\beta\hat{}_{p_g},\hat{\delta}_1,\ldots,\hat{\delta}_{p_e})^T$, $p = p_g + p_e$, denotes the total number of unknown parameters, and $\mathbf{X}$ is the $n \times p$ design matrix,

$$\mathbf{X}_{n\times p}=\begin{pmatrix} x_{11} & \cdots & x_{1p_g} & z_{11} & \cdots & z_{1p_e} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} & \cdots & x_{np_g} & z_{n1} & \cdots & z_{np_e} \end{pmatrix}$$

In equation (4), $\hat{\boldsymbol{\theta}}$ is the least squares estimate of $\boldsymbol{\theta}$. Note in equation (1) we do not consider any interaction effects between gene-gene. If the necessity arises we can easily incorporate a gene-gene interaction term in the same model.

Optimisation problem

In equation (1), we have a set of $p(= p_g + p_e)$ total factors consisting of main effects of genes and environmental factors. By excluding or including each of the p factors in the model we can construct a total of 2^p alternate models. In variable selection the task is to select one model which best represents the response y from among these 2^p models. For example, if $p = 20$ then we have to choose the best model from a set of $2^{20} = 1048576$ possible models. Let us denote C to be the set of all p factors and Ω to be the set of all possible subsets (2^p) of C. Thus, the variable selection problem of choosing the "best" representative subset, ω, from the total 2^p subsets of C can be viewed as a "combinatorial optimisation" problem.

Genetic algorithm

A genetic algorithm (GA) is a global optimisation procedure well-suited for solving combinatorial optimisation problems (Goldberg, 1989). It uses the Darwinian principle of the "survival of the fittest" as its optimisation strategy. GA has been shown to be a robust and effective search method requiring very little information about the problem to explore a large search space. Chatterjee et al. (1996) shows how GA can be applied to many statistical problems. Application of GA and a parallel version of GA (parallel genetic algorithm) to variable selection problems in statistics has been discussed in detail by Liu & Iba (2001) and Zhu & Chipman (2006). Carlborg et al. (2000) applied genetic algorithms to QTL mapping. The algorithm starts with a randomly selected initial population, $\omega_1,\ldots,\omega_m$, consisting of binary strings (vectors of zeros and ones). Each binary string ω_i is treated as the genetic code of an individual in the initial population and each of the positions in the string are treated as a single gene. Thus the name "genetic algorithm". The chosen initial population goes through three operations, namely, selection, reproduction and mutation to produce a new generation.

GA in a variable selection framework

In the context of the variable selection problem, each ω_i represents a different subset of variables and $(\omega_1,\ldots,\omega_m)$ together specifies the set of m different models. Suppose we are interested in studying the relationship between response y and two genetic factors (x_1 and x_2) and two environmental factors (z_1 and z_2), then $p = 2 + 2 = 4$ and the total number of possible models is 2^4. Let $\omega_1 = \{x_1, z_1\}$, then we can code ω_1 as {1, 0, 1, 0}, which is a

binary string of length 4. Here 1 represents that the particular factor (main effect of gene or environment) has been included while 0 represents that it has been excluded.

- *Selection*: Each of the *m* models is evaluated by a fitness function (F). Common choices of fitness functions are AIC, BIC, and cross-validated score of each model. If the goal is to minimise (maximise) F, then the *m* models are arranged in an increasing (decreasing) order of their F values and the top *m*/2 models are chosen to survive till the next generation. Models are then selected from this survival pool to undergo the operations of reproduction and mutation.
- *Reproduction*: Two models (parent models) are selected at random to produce a new model (child) using the principle of recombination (cross-over). A cross-over position *j* is chosen at random between 1 to *p* so the new model has the first *j* terms from one parent model and the rest of the terms (i.e., *j* + 1 to *p*) from the other parent model.
- *Mutation*: The new model (child) is given a small probability (mutation rate) to alter one of its binary code (1 to 0 or vice versa) at any one random position from 1 to *p*.

The processes of reproduction and mutation are continued until a total of *m*/2 new models are produced and the next generation is also of size *m*. Thus, half of the next generation is chosen by the selection process and the other half by the reproduction and mutation processes. The algorithm is repeated until it converges, i.e., each of the *m* selected models in a single generation are identical to each other.

As the number of possible factors, *p*, increases so does the size of the search space (2^p), thereby increasing the computational cost of running a GA. In order to apply the genetic search to large-scale problems, parallel genetic algorithms (PGAs) are widely used. A natural way to parallelise a GA is to run several GAs simultaneously to find the optimal solution. It has been shown by Zhu & Chipman (2006) that PGA is more successful in obtaining the optimal solution for a variable selection problem. Using several examples, they explain why a PGA is more successful in variable selection than a GA. A PGA makes use of multiple computing resources at the same time and divides the larger problem into several smaller ones. Consequently, this paper aims to use PGA in choosing the most important factors affecting a quantitative disease phenotype.

Parallel genetic algorithm (PGA)

Suppose *B* GAs each of length *N* are being run simultaneously (in parallel). Each of the *B* GAs start with a fixed number of models, *m*, and are run *N* number of times undergoing the three operations, namely selection, reproduction and mutation. We use the generalised cross-validation criterion (GCVC) as the fitness function F in the selection process,

$$F(\omega) = -GCVC(\omega) = -\sum_{i=1}^{n}\left(\frac{y_i - \widehat{y}_i(\omega)}{1-(r+1)/n}\right)^2 \quad (5)$$

where *r* is the number of variables (main effects of genetic and environmental factors) contained in the subset ω, and $\hat{y}_i$ is the predicted value of y_i from the model containing *r* variables (plus the intercept term). Thus, each of the *m* models in the *B* parallel GAs are evaluated using the generalized cross-validation. Since the goal is to minimise the score, we arrange the models in an increasing order and choose the top *m*/2 models to go into the survival pool. Note that other popular fitness functions like, Akaike Criterion (AIC), PRESS criterion and Mallow's C_p (Kutner et al., 2001) may also be used. Golub et al. (1979) shows

that GCVC a rotation invariant version of the the PRESS criterion, performs better than the PRESS criterion for ill-conditioned design matrices when used for model selection. We have not shown any computations using PGA based on any other criterion than GCVC, however we have checked that the time taken for PGA to converge using any other criterion is more or less the same.

After running the B GAs simultaneously N number of times, each variable is assigned an importance measure (Zhu & Chipman, 2006). Using these measures the variables are then ranked and the top few variables (say q) are selected to form the "best" subset of variables which affect y.

Let $P(b, t)$ denote the t-th ($t = 1,\ldots,N$) generation of the bth GA ($b = 1,\ldots,B$). Define the quantity $r(j, b)$ as

$$r(j,b)=\frac{1}{m}\sum_{i=1}^{m}\omega_i(j),\ \omega_1,\ldots,\omega_m\ \in P(b,N),\ j=1,\ldots,p, \tag{6}$$

where $\omega_i(j)$ is the binary variable (0/1) representing the inclusion (takes value 1) or exclusion (takes value 0) of the jth factor in the ith model. Thus, $r(j, b)$ is the proportion of the models in the N-th generation of the bth GA that contains variable j. Combining the information regarding the jth variable across the B GAs we compute

$$\bar{r}_j=\frac{1}{B}\sum_{b=1}^{B}r(j,b), \tag{7}$$

where $\bar{r}_j$ ($j = 1,\ldots,p$) is the importance measure of the jth variable averaged across the B GAs. If a variable is important i.e., it actually affects y then $r(j, b)$ will be high for most values of b, giving rise to a high importance measure $\bar{r}_j$. However, a spurious variable (a variable not affecting y) will have a high $r(j, b)$ value for only some values of b and low values of $r(j, b)$ for others, thus resulting in a low $\bar{r}_j$.

We next plot the $\bar{r}_j$ ($j = 1,\ldots,p$) values after ordering them from largest to smallest. Such a plot is called an order plot. In an order plot, we look for the largest gap existing between groups of variables and select those which lie above the largest gap. Let the number of variables lying above the largest gap in the order plot be denoted by q. This set of q variables then forms the best subset of variables affecting y. Plotting the unordered $\bar{r}_j$ values gives the bubble plot. Since important variables have a high $\bar{r}_j$ value thus they float on top of the plot like bubbles.

Specifying the parameters of a PGA

Here, we discuss certain guidelines for choosing the various parameters in a PGA.

- *Population size and Mutation rate*: De Jong and Spears (1990), Grefenstette (1986), Goldberg (1989), and Zhu & Chipman (2006) discuss various choices of m and mutation rates. Zhu & Chipman (2006) chooses the size of the initial population, m, to be equal to the number of factors, when p is even. For cases where p is odd, m is chosen to be $(p + 1)$. To avoid fractions in the survival pool (refer to Selection on page 10), the population size is always chosen to be an even number. The mutation rate is chosen to be the reciprocal of the population size m. For our simulations and real data we use Zhu & Chipmans's choice of m and mutation rate (results given in Table 2).

Dejong and Spears in their 1990 paper recommend the standard settings of $m = 50$ and mutation rate = 1/1000. However, in Grefenstette (1986) the recommended settings are $m = 30$ and mutation rate = 1/100. In Table 3 and Table 4 we use the three settings as suggested by Grefenstette (1986), Dejong & Spears (1990) and Zhu & Chipman (2006) to see the effect of the tuning parameters on PGA's efficiency.

- *Stopping criterion*: In each iteration of the algorithm we have a total of m models. Each of these models are coded as binary strings of length p. Let f_{uj} denote the value at the jth ($j = 1,\ldots,p$) position of the binary string for the uth ($u = 1,\ldots,m$) model, and $\bar{f}_j = \frac{1}{m}\sum_{u=1}^{m} f_{ij}$ denote the value at the jth position averaged over the collection of m models. Thus f_{uj} takes values 1 or 0 according to the presence or absence, respectively, of the jth factor in the uth model. If all the m models are identical to each other in a single generation, then $\bar{f}_j$ takes values 1 or 0 according to the presence or absence, respectively, of the jth factor in all the m models. Define the average entropy of a collection of m models as Zhu & Chipman (2006) (p. 495).

$$\text{entropy} = -\frac{1}{p}\sum_{j=1}^{p}[\bar{f}_j \log_2(\bar{f}_j) + (1-\bar{f}_j)\log_2(1-\bar{f}_j)]. \tag{8}$$

  Note, if $\bar{f}_j$ is 1 or 0 for all j then the entropy value is 0. Hence we can say that the GA has converged (all m models are identical in a single generation) when the entropy is close to zero (i.e., below a specified value δ, where δ is chosen to be small).

- *Number of iterations*: to choose N, a single GA is run for a few times (say 5 or 10 times) using different initial populations of size m and the average number of generations needed to achieve convergence (entropy $< \delta$) is recorded. Half of the average number of generations needed for the single GA to converge is then used as a value for N for running the PGA (Zhu & Chipman, 2006) (p. 495). Since the task of a PGA is to break down the computational load of a large GA into several smaller ones, the value of N chosen is usually small (in the vicinity of 10). For example, running 50 (= B) GAs parallelly with $N = 10$ generations each, is equivalent to a single GA of $50 \times 10 = 500$ generations.

- *Number of parallel GAs*: Zhu & Chipman (2006) (p. 495) shows how the choice of B (number of parallel GAs) is linked to the length of the largest gap in the bubble plot by the following relation

$$d \geq Z_\alpha \sqrt{\frac{1}{4B}}, \tag{9}$$

  where, Z_α is the α-th quantile of the standard normal. Here, d is the difference between the minimum $\bar{r}_j$ value in the group of the top q variables of a bubble plot and the maximum $\bar{r}_j$ value among the remaining $(p - q)$ variables which sink to the bottom of the bubble plot. Thus, in choosing B we should be confident that the gap between the two sets of variables separated by d is not due to chance. Since each of the parallel GAs work independently of each other, we can start the PGA with a small B and increase B until condition (9) is satisfied. Moreover, for a particular value of B if we find that condition (9) has been satisfied, then increasing the value of B does not cause any change in the accuracy of the algorithm. Results reported

later in Table 3 and Table 4 show that there is no appreciable change in the efficiency of PGA if we increase B from 50 to 150, since for $B = 50$ condition (9) has already been satisfied. However, if we start with a small B and find that condition (9) is not satisfied then we should increase B till (9) holds.

More details about the choice of the tuning parameters and the stopping criterion for a GA can be found in Zhu & Chipman (2006) (p. 495).

Simulated examples

We consider four different scenarios here, and conduct repeated simulations in each to evaluate the performance of a PGA. The performance of PGA is then compared with the Bayesian variable selection tool also known as SSVS (George and McCulloch, 1993). In each of the four scenarios we repeat the simulation 300 times and count the proportion of times each method chooses the true model.

The key idea of SSVS is to use a latent binary vector to index different possible subsets of variables (models). Priors are then imposed on regression parameters as well as on the set of possible models. In this framework, the promising subsets of predictors can be identified as those with higher posterior probabilities. The search of the models with high posterior probabilities is done using Gibbs sampling. Those subsets of variables with higher posterior probability can be identified by their more frequent appearance in the Gibbs sample. A brief description of SSVS is presented in the next paragraph (more details given in George and McCulloch (1993)).

Consider the regression situation,

$$\mathbf{y} \sim N(\mathbf{X}\theta, \sigma^2 I),$$

where $\mathbf{y} = (y1,\ldots,y_n)^T$, $\mathbf{X}$ is the $n \times p$ design matrix, $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_p)^T$ and σ^2 are unknown parameters.

Introducing p binary latent variables ν_i, ($i = 1,\ldots,p$), the regression coefficients are assumed to have the following multivariate normal prior distribution

$$\theta|\boldsymbol{\nu} \sim N_p(0, D_{\boldsymbol{\nu}} R D_{\boldsymbol{\nu}})$$

where $\boldsymbol{\nu} = (\nu_1,\ldots,\nu_p)$ and

$$f(\nu_i) = \lambda_i^{\nu_i}(1 - \lambda_i)^{(1-\nu_i)},$$

and R is the prior correlation matrix and $D_{\boldsymbol{\nu}} = diag(a_1\tau_1,\ldots,a_p\tau_p)$ with a_i taking value 1 if $\nu_i = 0$, and $a_i = c_i$ otherwise.

Following George and McCulloch (1993), a small non-negative value of τ_i should be chosen when $\nu_i = 0$ since then $\theta_i \sim N(0, \tau_i^2)$. While a large value ($> 1$) should be chosen for c_i when $\nu_i = 1$, since $\theta_i \sim N(0, c_i^2\tau_i^2)$. Some of the choices the authors consider in their 1993 paper are $(\sigma_{\theta i}/\tau_i, c_i) = (1, 5)$; (1, 10); (10, 100) and (10, 500), where $\sigma^2_{\theta i}$ is the variance of the estimate of θ_i and $\lambda_i = 0{:}5$, for all $i = 1,\ldots,p$. Some recommended choices of R are the identity matrix or the design correlation matrix, $(\mathbf{X}'\mathbf{X})^{-1}$. Finally, σ^2 has the prior distribution,

$$\sigma^2|\nu \sim \mathrm{IG}(a/2, ab/2).$$

The main goal is to obtain the marginal posterior density, $f(\mathbf{v}|\boldsymbol{y}) \propto f(\boldsymbol{y}|\mathbf{v})\, f(\mathbf{v})$. The algorithm uses the Gibbs sampler to sample from the posterior density of **v**. After the convergence of the Gibbs sequence the values of $f(\mathbf{v}|\mathbf{y})$ provide a ranking which is used to select the most promising subset of the factors.

Scenario 1 (High minor allele frequencies)—We create a data set with 30 genetic factors ($p = p_g = 30$) of length $n = 100$. Here $p_e = 0$, since there are no environmental factors. The true underlying model used to generate the continuous disease phenotype y is,

$$y_i = 2\, x_{i5} + 2.5\, x_{i10} + 2\, x_{i15} + 2.5\, x_{i20} + 2\, x_{i25} + 2.5\, x_{i30} + \epsilon_i, \tag{10}$$

where $\epsilon_i \sim N(0, 1)$, $i = 1,\ldots,100$. The genetic factors affecting y, i.e., x_5, x_{10}, x_{15}, x_{20}, x_{25} and x_{30} are obtained as independent vectors from Bernoulli distributions with parameters 0.70, 0.73, 0.76, 0.79, 0.82, 0.85, respectively. The other predictor variables which have no effect on y (i.e., not present in equation 10) are generated independently from the following Bernoulli distributions: $x_1,\ldots,x_4 \sim$ Ber (0.5), $x_6,\ldots,x_9 \sim$ Ber (0.6), $x_{11},\ldots,x_{14} \sim$ Ber (0.7), $x_{16},\ldots,x_{19} \sim$ Ber (0.8), $x_{21},\ldots,x_{24} \sim$ Ber (0.55), $x_{26},\ldots,x_{29} \sim$ Ber (0:65). Note that the genetic factors related to the disease phenotype have relatively high minor allele frequencies (> 0.10). Using similar allele frequencies we also generated y using the true model,

$$y_i = x_{i5} + x_{i10} + x_{i15} + x_{i20} + x_{i25} + x_{i30} + \epsilon_i, \tag{11}$$

where $\epsilon_i \sim N(0,1)$, $i = 1,\ldots,100$.

Scenario 2 (Low minor allele frequencies)—In the second scenario we consider informative genes with low minor allele frequencies. Once again we consider a data set with 30 genetic factors ($p = p_g = 30$) of length $n = 100$. The underlying model is,

$$y_i = 3\, x_{i5} + 3.5\, x_{i10} + 4\, x_{i15} + 4.5\, x_{i20} + 5\, x_{i25} + 5.5\, x_{i30} + \epsilon_i, \tag{12}$$

where $\epsilon_i \sim N(0,1)$, $i = 1,\ldots,100$. The informative genes, x_5, x_{10}, x_{15}, x_{20}, x_{25}, x_{30}, are obtained independently from Bernoulli(0.95) distributions. Thus, each of the informative genes, x_5, x_{10}, x_{15}, x_{20}, x_{25}, x_{30}, have minor allele frequencies equal to 0.05. The non-informative genetic factors are generated independently from the following Bernoulli distributions: $x_1,\ldots,x_4 \sim$ Ber (0.8), $x_6,\ldots,x_9 \sim$ Ber (0.85), $x_{11},\ldots,x_{14} \sim$ Ber (0.9), $x_{16},\ldots,x_{19} \sim$ Ber (0.95), $x_{21},\ldots,x_{24} \sim$ Ber (0.93), $x_{26},\ldots,x_{29} \sim$ Ber (0.95). We also used a second model to generate y as follows;

$$y_i = 2.5\, x_{i5} + 2.5\, x_{i10} + 2.5\, x_{i15} + 2.5\, x_{i20} + 2.5\, x_{i25} + 2.5\, x_{i30} + \epsilon_i, \tag{13}$$

while keeping the allele frequencies unchanged.

Scenario 3 (Pairwise correlation between genetic factors)—The response y is generated using the model,

$$y_i = 2\,x_{i5} + 2\,x_{i10} + 2\,x_{i15} + 2\,x_{i20} + 2\,x_{i25} + 2\,x_{i30} + \epsilon_i, \quad (14)$$

where $\epsilon_i \sim N(0,1)$; $i = 1,\ldots,100$. Genetic factor x_5 is taken to be pairwise-correlated (correlation of 0.20) with the other factors (i.e., x_{10}, x_{15}, x_{20}, x_{25}, x_{30}). The genetic factors were generated from a multivariate binary distribution where $corr(x_5, x_j) = 0.20$; $j = 10, 15, 20, 25, 30$, using the BINDATA package from R software (Version 0.9–14; http://cran.r-project.org/web/packages/bindata/index.html). The informative genes, x_5, x_{10}, x_{15} are obtained independently from Ber(0.85), while x_{20}, x_{25}, x_{30} are obtained independently from Ber(0.90). The non-informative genetic factors are generated independently from the following Bernoulli distributions: $x_1,\ldots,x_4 \sim$ Ber (0.8), $x_6,\ldots,x_9 \sim$ Ber (0.85), $x_{11},\ldots,x_{14} \sim$ Ber (0.90), $x_{16},\ldots,x_{19} \sim$ Ber (0.85), $x_{21},\ldots,x_{24} \sim$ Ber (0.90), $x_{26},\ldots,x_{29} \sim$ Ber (0.80).

Scenario 4 (Genetic and environmental factors)—In this scenario, we consider the effect of both genetic and environmental factors on the disease phenotype *y*. A data set is created with 20 genetic factors ($p_g = 20$) and 10 environmental factors ($p_e = 10$) of length $n = 100$. The outcome variable *y* representing the continuous disease phenotype is generated using the model,

$$y_i = 2\,x_{i5} + 2\,x_{i10} + 2\,x_{i15} + 2\,x_{i20} + 2\,z_{i1} + 2\,z_{i4} + 2\,z_{i6} + 2\,z_{i8} + \epsilon_i, \quad (15)$$

where $\epsilon_i \sim N(0,1)$. The associated genetic factors (i.e., x_5, x_{10}, x_{15}, x_{20}) are obtained as independent vectors from Bernoulli distributions with parameters 0.80, 0.83, 0.85, 0.90, respectively. Environmental factors z_1, z_4, z_6, z_8 which are associated with the phenotype *y*, are obtained as independent standard normal vectors. The genetic factors which are not affecting *y*, are each generated independently from the following Bernoulli distributions: $x_1,\ldots,x_4 \sim$ Ber (0.8), x6,…,$x_9 \sim$ Ber (0.83), $x_{11},\ldots,x_{14} \sim$ Ber (0.85), $x_{16},\ldots,x_{19} \sim$ Ber (0.9). Un-associated environmental factors, z_2, z_3, z_5, z_7, z_9, z_{10}, are each generated independently from $N(0, 1)$ distributions.

Example based on real data

The data for our real example is obtained from the Genetic Analysis Workshop 2006. This is a multi-centred study to identify genetic factors that predispose for rheumatoid arthritis by the North American Rheumatoid Arthritis Consortium. We use one of the quantitative traits, anti-cyclic citrullinated peptide (anti-CCP), which is more specific for the disease and is a better predictor of erosive outcome (Huizinga et al., 2005) as *y*. The data consist of $n = 685$ observations on a normally distributed response variable *y* (Anticcp) and $p = p_g = 20$ SNP markers on Chromosome 6. Thus there are 2^{20} possible models. Here we only consider models of the form

$$y_i = \beta_0 + \sum_{j=1}^{20} \beta_j x_{ij} + \epsilon_i, \; i = 1, \ldots, 685. \quad (16)$$

Simulated example based on recessive associated genes

We consider a situation where the associated genes affect the phenotype *y* recessively. Model (1) is then re-written as

$$y_i=\beta_0+\sum_{j=1}^{p_g}\beta_j x_{ij}+\sum_{k=1}^{p_e}\delta_k z_{ik}+\epsilon_i,\ i=1,\ldots,n, \tag{17}$$

where x_{ij}, the jth ($j = 1,\ldots,p_g$) genetic factor for the ith individual is coded as

$$x_{ij}=\begin{cases} 1 & \text{if the genotype of the } i\text{th individual is aa} \\ 0 & \text{if the genotype is either AA or Aa} \end{cases} \tag{18}$$

In our simulation we assume the true underlying model for the phenotype y to be,

$$y_i=2.5\ x_{i5}+2.5\ x_{i10}+2.5\ x_{i15}+2.5\ x_{i20}+2.5\ x_{i25}+2.5\ x_{i30}+\epsilon_i, \tag{19}$$

where $\epsilon_i \sim N(0,1)$, $i = 1,\ldots,100$. The associated genes are distributed independently as; $x_5 \sim Ber(0.05)$, $x_{10} \sim Ber(0.95)$, $x_{15} \sim Ber(0.03)$, $x_{20} \sim Ber(0.95)$, $x_{25} \sim Ber(0.95)$, $x_{30} \sim Ber(0.04)$. The non-informative genes are from the following independent Bernoulli distributions: $x_1,\ldots x_4 \sim$ Ber (0:8), $x_6,\ldots,x_9 \sim$ Ber (0.85), $x_{11},\ldots,x_{14} \sim$ Ber (0.9), $x_{16},\ldots,x_{19} \sim$ Ber (0.95), $x_{21},\ldots,x_{24} \sim$ Ber (0.93), $x_{26},\ldots x_{29} \sim$ Ber (0.95).

RESULTS

Simulation results

We performed 300 simulations in each of the four scenarios and determined the proportion of times the two methods, PGA and SSVS, are able to identify the correct model. PGA will be said to have found the right model if all the correct predictor variables are ranked ahead of the rest, using the $\bar{r}_j$ values. Similarly SSVS will be said to have chosen the right model if all the correct variables are ranked ahead of the rest, using the posterior probabilities. We list the parameters of a PGA for each of the four simulations in Table 1.

To run SSVS we choose two different prior distributions for the hyper parameters; (σ_β/τ, c) = (1, 10), (1, 5) and $R = I$. The Gibbs sampler underwent 10000 iterations with 1000 burn-ins in each case. The value of λ was chosen to be 0:5 in each case.

Table 2 summarizes the simulation results for each of the four different situations. It shows that PGA and SSVS perform well for scenario 1 (model 10), scenario 2 (model 12), scenario 3 and 4. For true model 11 (scenario 1) the efficiency of both PGA and SSVS is reduced. This reduction is due to using a low value of the coefficients for each genetic factor in model (11). For true model 13 (scenario 2), we see that PGA's performance is much worse when compared to SSVS. PGA in this case chooses the correct model in only 73% of the cases while SSVS chooses the correct model 90% of the time. Overall, we see from Table 2 that the performance of PGA is highly comparable to SSVS except for model 13. However, to use SSVS one needs to know the best choice of priors for the hyperparameters, and a misspecification of the hyperparameters may lead to an incorrect choice of variables. Also PGA is easy to use computationally. We used the R software (Version 2.7.1) to run PGA and Matlab (Version 7) to run SSVS. The average runtime for the PGA was one hour and for SSVS was 6 hours. The computer used for computations was a Intel (R) Core(TM)2 Duo CPU E6750 @ 2.67 GHz, 1.97 GB of RAM. The computer programs are available on request from the first author.

To study the effects of the different tuning parameters (*m*, mutation rate, *B* and *N*) on PGA, we vary the parameter values and list the results in Table 3 and Table 4. We study only simulations 1 (model 11) and 2 (model 13) for different tuning parameters. From Table 3 and Table 4 we see that there is no appreciable change in the results corresponding to a variation in the parameter values.

Results for real data

For the PGA we use $B = 50$ parallel universes, each involving $N = 8$ generations with a population size of $m = 20$.

Figure 1 shows the bubble plot for $B = 50$ parallel paths. From the figure, we see that the variables X_6 and X_{14} are floating high at the top of the bubble plot well separated from the rest of the variables. PGA selects the two variables, X_6 and X_{14}, above line G. The results of the PGA is also supported by the order plot in Figure 2. In the order plot we see that the variables X_6 and X_{14} are separated from the rest by the largest gap in the plot. This gap is also significant according to formula 3, i.e., for $B = 50$ the gap between the two sets of variables ((X_6, X_{14}) and the rest of the variables) is not by random fluctuation. The $\bar{r}_j$ values of variables X_6 and X_{14} are ranked ahead of the rest of the variables. So we conclude using the PGA method that the SNPs rs11908 and rs909472 denoted by X_6 and X_{14} respectively, affect the response variable Anticcp. Interestingly, both of the SNPs, rs11908 and rs909472, are located within the *HLA-DRB1* gene on 6p21, which has been known to be implicated in contributing to the risk for rheumatoid arthritis by many studies (Newton et al., 2004;Chiu et al., 2007).

We next consider the following model for the response Anticcp

$$y_i = \beta_0 + \sum_{j=1}^{20} \beta_j x_{ij} + \gamma_{6(14)} x_{i6} x_{i(14)} + \epsilon_i, \tag{20}$$

for $i = 1,\ldots,685$. From the order and bubble plots we already know that the SNPs rs11908 and rs909472 denoted by X_6 and X_{14} respectively, affect the response y. Thus, instead of testing the interaction effects between all the factors we restrict ourselves to the interaction between the two factors already chosen by PGA. We run PGA for model (14) with $p = 21$ for $B = 100$ parallel paths. Again, PGA selects the variables X_6 and X_{14}. Thus the interaction effect between X_6 and X_{14} is not included in the best subset of variables chosen by PGA.

Simulation results for example based on recessive associated genes

We ran PGA for model (16) and performed 300 simulations. Parameter values used were $B = 50$, $m = 30$, mutation rate = 1/30 and $N = 10$. The proportion of times PGA chose the correct model was 0:71. Thus, we see that for cases where the associated gene is recessive the efficiency of PGA is not very high.

DISCUSSION

In this paper, we showed that the PGA-based model selection method could be used to identify the possible QTLs as well as environmental factors. In our simulations, we assumed that the number of possible genetic markers p are not more than 30. In real applications, we usually have a handful of interesting candidate regions, which could come from previous research or after the genome-wide scanning. Our methods could be useful for following-up the candidate regions and fine mapping with haplotype-tagging SNPs. In particular, the haplotype-tagging SNPs are chosen so that one specific SNP could be used to capture the

Linkage Disequilibrium (LD) structure in one haplotype block so that the LD between these SNPs could be rather low. Our simulations show that PGA performs very well when the correlations between genetic markers are low. The other applications of this method is to scan the genome for possible associations with the quantitative trait using a moving window approach. However, PGA-based approaches are rather general. Zhu & Chipman (2006) in their paper applied PGA to select variables when the number of factors was 60. Liu & Iba (2001) used PGA in microarray gene expression analysis a for much larger *p*. They used two data sets with 7192 and 2000 genes, respectively. Thus, the PGA can be applied to cases with large number of genetic and environmental factors and interaction among these factors. It is also possible to extend the current method to allow more complex genetic models and mult-allelic markers by allowing x_{ij} take more than just two values. Such extension would be useful for microsatellite markers and haplotypes comprised of several SNP markers. The current PGA method does have the limitation that it cannot be directly applied to genome-wide association studies containing millions of markers, and rather is more suitable for following promising markers as in stage II of a two-stage design (Wang et al., 2006, Zuo et al., 2006, Skol et al., 2007).

PGA seems to work well when tested using both real and simulated data sets. PGA has also been shown to be highly comparable to the widely used Bayesian variable selection method SSVS. Interestingly, Oh (2007) applied SSVS for linkage with Haesman Elston regression on the same data set from the Genetic AnalysisWorkshop 15. However, Oh used IgM as phenotype, not anti-CCP. The author found strong evidence for main effects on chromosome 6, which could be due to the correlation of IgM and anti-CCP. It should be noted that the correct choice of the variables using any Bayesian variable selection method is largely dependent on the choice of the hyperparameters. Recently, there have been several studies on the improvement of the Bayesian variable selection methods (Chipman et al., 2001; Swartz et al., 2006, 2008). Other variable selection methods like stepwise procedures such as forward and backward elimination using the Akaike information criterion (AIC) or the Bayesian information criterion (BIC) also perform poorly when compared to PGA (see Zhu & Chipman (2006), pp. 497–499).

Acknowledgments

This work is supported by R21NS057506 grant from the US National Institute of Health; We also want to acknowledge NIH grants GM031575 and AR44422 for supporting the Rheumatoid Arthritis data sets.

References

Ball RD. Bayesian methods for quantitative trait loci mapping based on model selection: approximate analysis using the bayesian information criterion. Genetics. 2001; 159:1351–1364. [PubMed: 11729175]

Broman KW, Speed TP. A model selection approach for the identification of quantitative trait loci in experimental crosses. J R Stat Soc Ser B Stat Methodol. 2002; 64:641–656.

Carlborg O, Andersson L, Kinghorn B. The use of a genetic algorithm for simultaneous mapping of multiple interacting quantitative trait loci. Genetics. 2000; 155:2003–2010. [PubMed: 10924492]

Chatterjee S, Laudoto M, Lynch LA. Genetic algorithms and their statistical applications: an introduction. Computational Statistics and Data Analysis. 1996; 22:633–651.

Chipman, H.; George, EI.; McCulloch, RE. The Practical Implementation of Bayesian Model Selection. In: Lahiri, P., editor. IMS Lecture Notes - Monograph Series. Vol. 38. Beachwood, OH: IMS; 2001. p. 63134

Chiu Y, Chiou J, Chen Y, Kao H, Hsu F. Incorporating quantitative variables into linkage analysis using affected sib pairs. BMC Proceedings. 2007; 1 Suppl 1(1):S98. [PubMed: 18466602]

De Jong, KA.; Spears, WM. An analysis of the interacting roles of population size and crossover in genetic algorithms, Proc. First Workshop Parallel Problem Solving from Nature. Berlin: Springer-Verlag; 1990. p. 38-47.

George EI, McCulloch RE. Variable selection via gibbs sampling. J Amer Statist Assoc. 1993; 88:881–889.

Goldberg, DE. Genetic Algorithm in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley; 1989.

Golub GH, Heath M, Wahba G. Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics. 1979; 21:215–223.

Grefenstette J. Optimization of control parameters for genetic algorithms. IEEE Transactions on Systems, Man and Cybernetics. 1986; 16:122–128.

Haley CS, Knott SA. A simple regression method for mapping quantitative trait loci in line crosses using flanking markers. Heredity. 1992; 69:315–324. [PubMed: 16718932]

Huizinga TW, Amos CI, van der Helm-van Mil AH, Chen W, van Gaalen FA, Jawaheer D, Schreuder G, Wener M, Breedveld FC, Ahmad N, Lum RF, de Vries RR, Gregersen PK, Toes RE, Criswell LA. Refining the complex rheumatoid arthritis phenotype based on specificity of the HLA-DRB1 shared epitope for antibodies to citrullinated proteins. Arthritis Rheum. 2005; 52:3433–3438. [PubMed: 16255021]

Kutner, MH.; Nachtsheim, CJ.; Neter, J.; Li, W. Applied Linear Statistical Methods. 5th Edition. Boston: McGraw-Hill Irwin; 2001.

Lander ES, Botstein D. Mapping mendelian factors underlying quantitative traits using RFLP linkage maps. Genetics. 1989; 121:185–199. [PubMed: 2563713]

Liu J, Iba H. Selecting informative genes with parallel genetic algorithms in tissue classification. Genome Informatics. 2001; 12:14–23. [PubMed: 11791220]

Newton JL, Harney SM, Wordsworth BP, Brown MA. A review of the MHC genetics of rheumatoid arthritis. Genes Immun. 2004; 5:151–157. [PubMed: 14749714]

Oh C. A Bayesian genome-wide linkage analysis of quantitative traits for rheumatoid arthritis via perfect sampling. BMC Proceedings. 2007; 1 Suppl 1:S110. [PubMed: 18466451]

Piepho HP, Gauch HG Jr. Marker pair selection for mapping quantitative trait loci. Genetics. 2001; 157:433–444. [PubMed: 11139523]

Skol AD, Scott LJ, Abecasis GR, Boehnke M. Optimal designs for two-stage genome-wide association studies. Genet Epidemiol. 2007; 31:776–788. [PubMed: 17549752]

Swartz MD, Kimmel M, Mueller P, Amos K. Stochastic Search Gene Suggestion: A Bayesian Hierarchical Model for Gene Mapping. Biometrics. 2006; 62:495–503. [PubMed: 16918914]

Swartz MD, Yu RK, Shete S. Finding factors influencing risk: Comparing Bayesian stochastic search and standard variable selection methods applied to logistic regression models of cases and controls. Statist. Med. 2008; 27:6158–6174.

Wang H, Thomas DC, Pe'er I, Stram DO. Optimal two-stage genotyping designs for genome-wide association scans. Genet Epidemiol. 2006; 30:356–368. [PubMed: 16607626]

Yi N, George V, Allison DB. Stochastic search variable selection for identifying multiple quantitative trait loci. Genetics. 2003; 164:1129–1138. [PubMed: 12871920]

Zhu M, Chipman HA. Darwinian evolution in parallel universes: A parallel genetic algorithm for variable selection. Technometrics. 2006; 48:491–502.

Zuo Y, Zou G, Zhao H. Two-stage designs in case-control association analysis. Genetics. 2006; 173:1747–1760. [PubMed: 16624925]

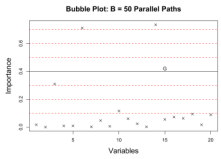

Figure 1.
Bubble plot from the PGA for $B = 50$ parallel paths. The PGA selects variables, X_6 and X_{14}, above solid line G. Variables X_6 and X_{14} are floating high at the top of the bubble plot well separated from the rest of the variables

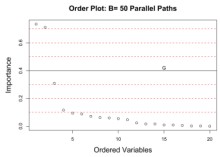

Figure 2.
Order plot from the PGA for $B = 50$ parallel paths. The PGA selects variables, X_6 and X_{14}, above line G. Variables X_6 and X_{14} are ranked ahead of the rest of the variables by their $\bar{r}_j$ values.

Table 1

Parameters of the PGA for situations 1:4

Parameters	Scenario 1	Scenario 2	Scenario 3	Scenario 4
m	30	30	30	30
mutation rate	$\frac{1}{30}$	$\frac{1}{30}$	$\frac{1}{30}$	$\frac{1}{30}$
N	10	10	10	10
B	50	50	50	50

Table 2

Comparison of the results of PGA and SSVS for simulations 1:4. PGA has been conducted using the parameters given in Table 1. Two sets of hyperparameters (1, 10) and (1, 5) have been used for SSVS. Both models 10 and 11 have been considered for simulation 1, similarly both models 12 and 13 have been used for simulation 2. For simulation 3 and 4 we use models 14 and 15, respectively.

Method	Scenario 1		Scenario 2		Scenario 3	Scenario 4
	Model (10)	Model (11)	Model (12)	Model (13)		
PGA	1.00	0.60	0.94	0.73	0.94	0.92
SSVS (1,10)	1.00	0.68	1.00	0.90	0.98	1.00
SSVS (1,5)	1.00	0.72	0.97	0.90	0.98	0.99

Table 3

Effect of tuning parameters (*B*, *m*, *N*, mutation rate) on the performance of PGA for scenario 1 (model 11)

***m* = 30**			
N	mutation rate	*B* = 50	*B* = 150
10	1/30	0.61	0.57
	1/100	0.55	0.60
15	1/30	0.61	0.69
	1/100	0.65	0.66
m = 50			
N	mutation rate	*B* = 50	*B* = 150
10	1/1000	0.64	0.66
15	1/1000	0.68	0.72

Table 4

Effect of tuning parameters (*B*, *m*, *N*, mutation rate) on the performance of PGA for scenario 2 (model 13)

***m* = 30**			
N	mutation rate	*B* = 50	*B* = 150
10	1/30	0.73	0.74
	1/100	0.70	0.73
15	1/30	0.71	0.71
	1/100	0.69	0.71
m = 50			
N	mutation rate	*B* = 50	*B* = 150
10	1/1000	0.71	0.73
15	1/1000	0.65	0.75