universität wien

MAGISTERARBEIT

Titel der Magisterarbeit

QTL mapping using mBIC and Genetic Algorithms

Verfasserin

Helga Björk Arnardóttir

angestrebter akademischer Grad

Magistra der Sozial- und Wirtschaftswissenschaften
(Mag. rer. soc. oec.)

Wien, 2013

Studienkennzahl lt. Studienblatt:	A 066 951
Studienrichtung lt. Studienblatt:	Magisterstudium Statistik
Betreuer:	Dipl.-Ing. Dr. Florian Frommlet, Privatdoz

Abstract

In this thesis, the problem of identifying quantitative trait loci (QTLs) is considered. Multiple regression analysis can be applied to locate QTLs in experimental populations. In that context, the use of a modified version of the Bayesian Information Criterion (mBIC) as a selection criterion has been well established in the past. Previously, stepwise selection procedures were used to find the model with the minimal selection criterion and consequently to find putative QTLs. However, while finding a solution relatively quickly, stepwise selection often fails to locate the globally minimal solution. In this thesis, a Genetic Algorithm (GA) is proposed to minimize mBIC. Furthermore, the Bayesian origin of the selection criterion is used to compute marker posterior probabilities using all models visited by the GA. The GA depends on several parameters and, in an extensive simulation study based on two different QTL scenarios, different GA parameter settings are examined. In both scenarios, the performance of the GA improves with larger *population size* and smaller *tournament group size* settings. In the final simulation study population size is examined in more detail in order to determine how small population size can be chosen without loosing power to detect correct QTLs. For most other parameters, we suggest using those settings that minimize runtime of the GA.

To Lilja Björk

Acknowledgements

This thesis was funded by the Vienna Science and Technology Fund (WWTF) through project MA09-007a.

First of all, I would like to express my graditute to my advisor, Florian Frommlet, for the opportunity to work on this project. He was a source of motivation and abundantly available for support.
I am very greatful to Malgorzata Bogdan and her warm welcome and hospitality at Politechnika Wroclawska. Her help at the early stages of the programming and her inspiring explanations of the origin of mBIC are greatly appreciated.
I would also like to thank Ivana Ljubic. She introduced me to the Genetic Algorithm and provided me with useful hints on how to make beautiful flow charts in LateX.

I would like to express my deepest love and gratitude to my family and friends. My parents, who have always given unconditional love and support and taught me what matters in life. My parents-in-law, for countless babysitting hours and reassurance at the last meters. My beutiful daughter Lilja, who was born in the middle of the writing, and has given me a reason to wake up with a smile every day. But most of all I would like to thank Frank, none of this would have been possible without you.

List of Abbreviations

AIC	Akaike Information Criterion
ANOVA	Analysis of Variance
BIC	Bayesian Information Criterion
CIM	Composite Interval Mapping
FPN	Total Number of False Positives and Negatives
FWER	Family Wise Error Rate
GA	Genetic Algorithm
IM	Interval Mapping
mBIC	Modified Bayesian Information Criterion
MIM	Multiple Interval Mapping
PGA	Parallel Genetic Algorithm
QTL	Quantitative Trait Locus
SNP	Single Nucleotide Polymorphism

Contents

Chapter 1

Introduction

Quantitative traits are phenotypes or characteristic traits that vary in magnitude. Typically they are influenced both by genetic as well as by environmental factors. Examples of such traits found in living organisms are the height of a human being, size of a tomatoe or how much milk a cow produces. The regions of the genome that contain genes that affect a quantitative trait are called quantitative trait loci (QTLs). In the early days of genetics, biologists often concentrated on traits which result from the mutation of a single gene. However, many interesting traits turn out to be of a more complex nature, where many different genes are influential.
The search for the number and position of QTLs, often called QTL mapping, has produced various statistical methods, such as interval mapping, composite interval mapping and multiple QTL mapping. In this thesis, we use multiple regression combined with model selection to detect possible QTLs. Variable selection is performed with a modified version of the Bayesian Information Criterion (mBIC). Since the number of models increases exponentially with the size of the database, a search over all possible models is not suitable. An obvious first approach is provided by stepwise selection procedures which however often terminate in local minima and fail to find the globally optimal solution. Therefore, we investigate the performance of Genetic Algorithms (GAs), a search heuristic mimicking natural evolution, to identify possible QTLs affecting the trait of interest.

This thesis is organized as follows: In Chapter 2, a brief summary of the genetic background of QTL mapping is provided, followed by a review of the classic statistical approaches used in QTL mapping. The origin of the mBIC is subsequently described. The Genetic Algorithm is introduced in Chapter 3. First, the traditional operators of the GA are outlined. Then, our version of the GA and the estimation of the marker posterior probabilities are described in detail. In Chapter 4 various GA parameters are examined for two different scenarios in an extensive simulation study. A final simulation study is undertaken to explore how small population size can be without loosing too much of the GA's performance power. The thesis is concluded with a summary of the results and a brief discussion of possible topics for future research.

Chapter 2

QTL Mapping

2.1 Experimental Crosses

Quantitative traits are usually influenced by both genetic and environmental factors. To identify the genetic factors influencing a quantitative trait one ideally investigates populations for which the environmental effects have been reduced to a minimum. In case of plants or animals, classical inbreeding experiments can be performed to determine genetic differences. Inbreeding experiments involve reproduction of two organisms which are genetically related to each other. The results are strains that have homozygous traits. When two strains which are raised in the same environment show consistent differences in the quantitative trait we can conclude that the variance is due to genetic influences. The simplest breeding to identify these genetic differences is backcross (see Figure 2.1).

Meiosis

During reproduction of two parents, their genetic material undergoes a biological process called meiosis. First, the chromosomes of each parent are duplicated during DNA replication. Then, the maternal and paternal homologous chromosomes pair to each other and perform crossover, that is, they exchange their genetic material and therefore make the offspring's chromosomes genetically distinct from either parent. This process contributes to genetic diversity in populations. If we observe at two different loci that the genetic material comes from different parents, then we can assume that an odd number of crossovers must have occured between them. We call this event recombination between two loci.

2.1.1 Genetic markers

Differences between the genetic material of individuals are observed at the level of DNA. DNA (deoxyribonucleic acid) consists of two long strands of nucleotides that form a spiral called the double helix. Each nucleotide consists of a sugar molecule, a phosphate molecule and one of the four chemical bases: adenine (A), guanine (G), cytosine (C) or thymine (T). The bases pair up with each other, A with T and G with C. These DNA strands are a part of an organism's chromosome which is stored in the cell nucleus of eukaryotes (including plants, animals and fungi). Diploid organisms, including humans, have a pair of homologous chromosomes, one from each parent, meaning that they have two pieces of every gene. A specific location on a chromosome is called a genetic locus and if a locus shows genetic variation between individuals, then different forms of that locus are called alleles. If at a particular locus on a chromosome, two different alleles exist, then we call them A and a (not to be confused with the adenine base). Therefore, a diploid organism, can have three different genotypes at a biallelic locus, namely, AA, Aa or aa. Loci with more alleles do exist but for simplicity we consider only biallelic markers. The first and the third genotype are called homozygous and the second heterozygous.
A genetic marker is a stretch of DNA with known physical location on a chromosome whose genotype shows variation between individuals. Each chromosome consists of 100.000 to 10.000.000.000 basepairs, but in QTL mapping a much smaller number of genetic markers distributed along the chromosomes is considered. Today, the most common types of genetic markers are single nucleotide polymorphism (SNP), which are a one basepair variation. Microsattelites, which are repeating sequences of 1-6 basepairs, previously played a more important role.

2.1.2 Backcross

In backcross design, two parental inbred lines which differ in the trait of interest are crossed to obtain a F_1 generation. Let's assume that the genotypes at a particular locus of the parental strains are AA and aa respectively. The individuals of the F_1 generation receive a chromosome from each parent, therefore having genotype Aa. Then the individuals of the F_1 strain are crossed with one of the parental strains. The backcross strain has finally two possible genotypes, namely AA and Aa (if the F_1 generation was crossed with the parent with genotype AA) or Aa and aa (if the F_1 generation was crossed with the parent with genotype aa)(see Figure 2.1).

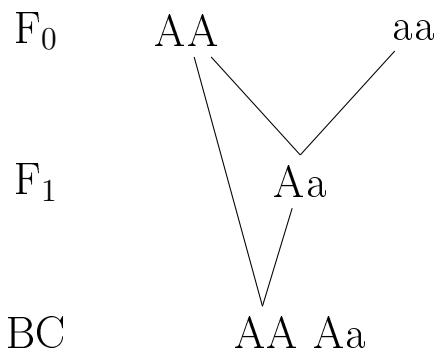

Figure 2.1: Backcross design

Due to crossover of chromosomes during meiosis, the chromosome received from the F_1 parent is a mixture of the two F_0 generation chromosomes (see Figure 2.2).

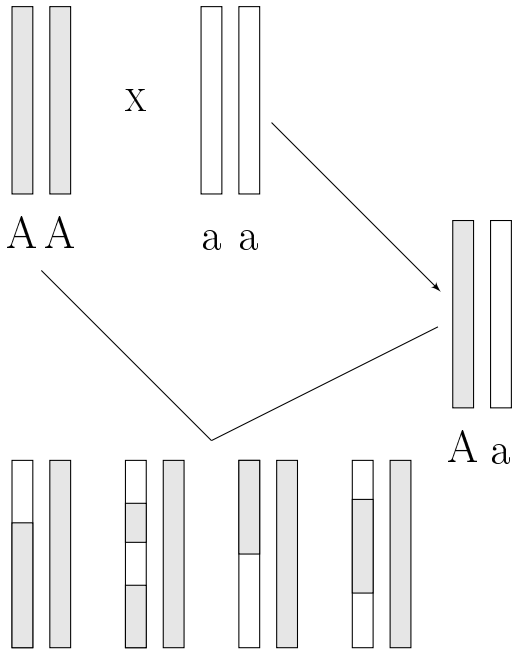

Figure 2.2: Chromosomes during backcross breeding

After a particular number of offsprings has been produced, the genotype of every genetic marker is determined and a genetic map for each individual is constructed. In backcross design, the only genetic variation among individuals of the BC generation is due to whether the F_1 parent passed on an A or an a allele. Typically, genotype AA is coded as 0 and genotype Aa as 1 in statistical modelling.

Each individual's genetic map shows the genotypes at all marker loci in linear order. Two neighbouring markers will have different genotypes if an odd number of crossovers has occured between them. Markers close together will have higher correlation than those far apart.

2.1.3 Intercross

More variation of genotypes is acquired when doing intercross (see Figure 2.3). In intercross F_1 generation parents are crossed with each other yielding the possible genotypes AA, Aa and aa.

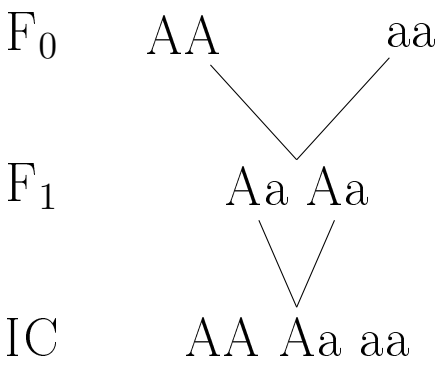

Figure 2.3: Intercross design

Again, because of meiosis, the offspring chromosomes become a mixture of the two F_0 generation chromosomes (see Figure 2.4) but now both chromosomes can receive both alleles from the F_0 generation.

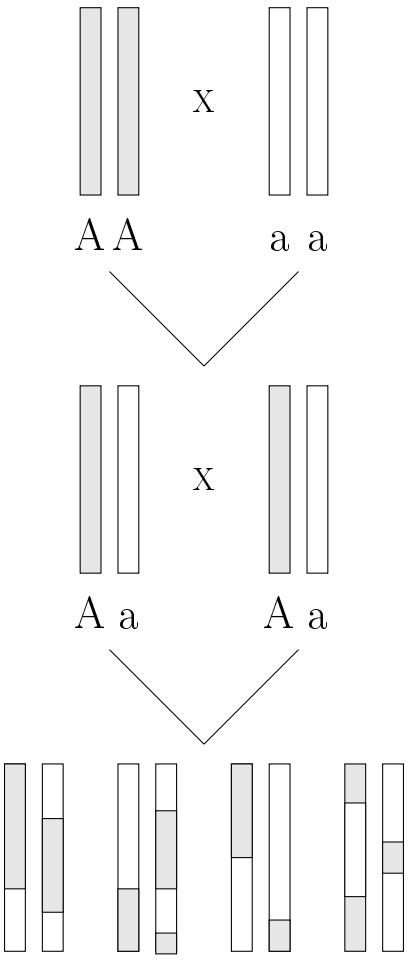

Figure 2.4: Chromosomes during intercross breeding

One advantage of doing intercross is that it allows to detect genes that have dominance effects. When a gene is dominant, the presence of one allele shows the same effect to the quantitative trait as when there are two alleles present. In backcross, a marker on this dominant gene might be missed since we would need all three genotypes to detect difference in the trait of interest. (see Figures 2.5 and 2.6)

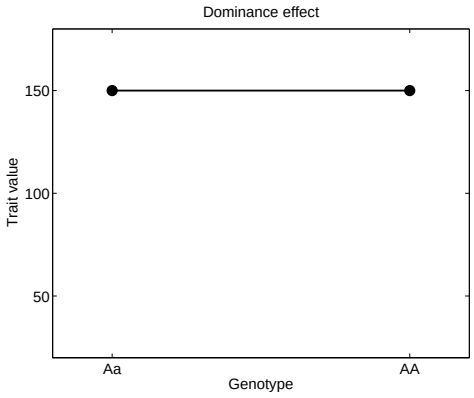

Figure 2.5: Dominance effect for a Backcross generation

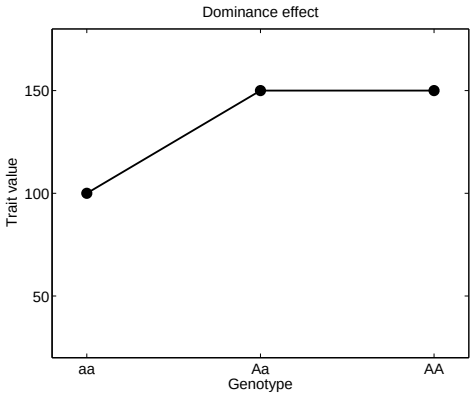

Figure 2.6: Dominance effect for an Intercross generation

When a gene has an additive effect, markers will be detected in both backcross and intercross design. A homozygous gene locus will have twice as much effect on the trait value as a heterozygous locus (see Figures 2.7 and 2.8).

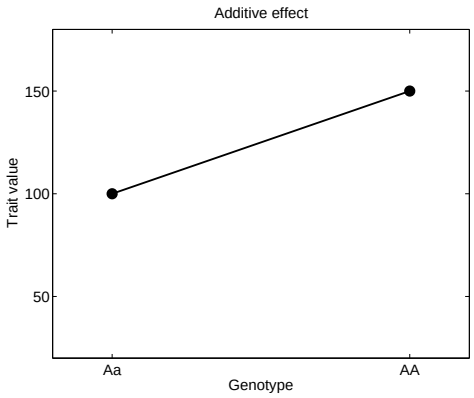

Figure 2.7: Additive effect for a Backcross generation

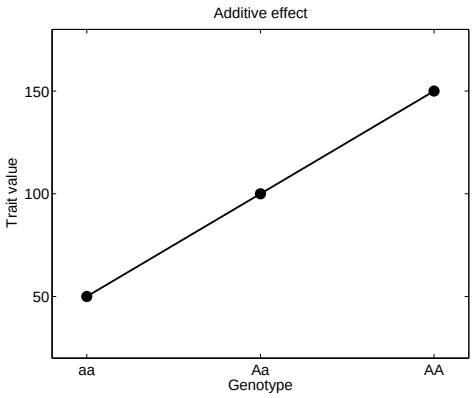

Figure 2.8: Additive effect for a Intercross generation

While intercross design is more powerful and more flexible for detecting different kinds of QTLs, its statistical analysis is also slightly more challenging (see Baierl, Bogdan, Frommlet and Futschik (2006)). In principle the methods discussed

in this thesis could be quite easily extended to intercross design but for the sake of clarity we focus on backcross design with additive effects.

2.1.4 Genetic distance

The physical distance between two loci on a chromosome is measured by the number of nucleotides between them. We will however measure the distance in Morgans (M), a unit which corresponds to crossover frequency. When two loci have 1 cM distance, the expected number of crossovers between them is 0.01.

Haldane mapping function

To estimate the probability of recombination between two loci, the Haldane mapping function is most commonly used. It assumes that crossovers occur independently of each other. Consequently, the number of crossovers on a single chromosome can be modeled as a Poisson process. If d is the distance between two loci, measured in Morgans, and X is the number of crossovers between the loci, then we can assume that X follows a Poisson process

$$P(X = \kappa) = e^{-d}\frac{d^\kappa}{\kappa!}, \quad \text{for } \kappa = 0, 1, 2, ...$$

The probability of recombination, that is, the probability of an odd number of crossovers, between the two loci is

$$\begin{aligned} r &= \sum_{\kappa=0}^{\infty} P(X = 2\kappa + 1) = \sum_{k=0}^{\infty} e^{-d}\frac{d^{2\kappa+1}}{(2\kappa+1)!} = e^{-d}\sum_{k=0}^{\infty}\frac{d^{2\kappa+1}}{(2\kappa+1)!} \\ &= e^{-d}\sinh(d) = e^{-d}\frac{e^d - e^{-d}}{2} = \frac{1 - e^{-2d}}{2}. \end{aligned}$$

2.2 Statistical Methods

Identifying QTLs has been attempted with various different statistical approaches. Several reviews and books on many of those methods have been published in recent years (see Doerge, Zeng and Weir 1997, Wu, Ma and Casella 2007, Broman and Sen 2009 and Siegmund and Yakir 2010).

2.2.1 Classical Methods

Analysis of Variance

The simplest way of finding QTLs is to perform a t-test or analysis of variance (ANOVA) (Soller *et al.* 1976). For offsprings of backcross breeding, the data is split into two groups at each marker, according to their genotype. The means of each group are then compared using a t-test. For intercross design, the more general form of the ANOVA is used to compare the group means, namely the F-test. According to Broman and Speed (2002) ANOVA, which considers only one locus at a time, has two major weaknesses. First, when detecting a QTL one cannot be sure about the location because no difference will be seen between a QTL that is located directly at a marker with a weak effect and a QTL located somewhere between markers with a strong effect. Second, individuals with missing genotypes at a particular marker cannot be used when searching for a QTL at that marker.

Interval Mapping

In interval mapping (see Lander and Botstein 1989), one calculates LOD scores for each position on a dense grid on the chromosome and thus estimates the location of the putative QTL. These estimates depend only on the genotypes of the flanking markers. To determine the LOD score at each location one assumes that the trait of interest follows a normal distribution with mean μ_{AA} and μ_{Aa} and a variance σ^2. These three parameters can be estimated using the EM algorithm (Dempster, Laird and Rubin 1977) at each position on the chromosome. The LOD score is then calculated comparing the hypothesis that there is a QTL at the given location with the hypothesis that there is no QTL. The LOD score can be considered as a function of chromosome position where its maximum indicates possible QTL locations.
The biggest benefits of doing interval mapping instead of ANOVA are the following: Missing data are allowed and areas between markers are considered as possible positions for a QTL. This increases the power when the markers are sparse and allows to estimate the location of a QTL more precisely. Also QTL effects are thus estimated better. When the markers are densely spaced and missing data are

almost non-existent, interval mapping and ANOVA perform quite similarly. Both methods consider only one QTL at a time, so they are not ideal in the case of complex QTL traits.

Composite Interval Mapping and Multiple Interval Mapping

Composite interval mapping (CIM) (Jansen 1993, Jansen and Stam 1994, Zeng 1993, 1994) is a combination of interval mapping and multiple regression methods. A subset of markers is included as regressors while interval mapping is performed to improve the power of QTL detection and the estimation of QTL effects. The biggest problem of this method is that it is not clear which markers should be used as regressors since choosing too many markers will decrease the power of QTL detection.
Multiple interval mapping, which was proposed by Kao, Zeng and Teasdale (1999), is similar to CIM except that the additional markers are not required and that it is closer to the standard model selection approach.

2.2.2 Model Selection

If the marker map is rather dense and the genotype data has few missing values the use of multiple regression is appropriate (Broman 2001). For QTLs with additive effects and data from backcross breeding, we denote the marker genotypes as $x_{ij} = 0$ for genotype AA and $x_{ij} = 1$ for genotype Aa, and y_i as the trait value of interest for individual $i = 1, ..., n$ and marker $j = 1, ..., m$. Here n is the total number of individuals and m the total number of markers. It is assumed that the relationship between the trait value and QTL genotype is given by the multiple regression model

$$y_i = \beta_0 + \sum_{j=1}^{m} \beta_j x_{ij} + \epsilon_i, \tag{2.1}$$

where $\epsilon_i \sim \mathrm{N}(0, \sigma^2)$ is the environmental noise and β_0 is the intercept. This formulation allows some regressors β_j to be zero when the marker is not affecting the trait value. Identification of those markers which are QTLs (or are close to one) poses a model selection problem. There are two main concerns when selecting the appropriate model for the data: First, how to decide which of two models is more suitable and second, establishing an adequate search strategy for finding the best model from the model space. Next section describes the model selection criterion which is used.

Selection criterion

To compare different models, some criterion is needed to evaluate the quality of each model. Let M_ι denote linear model ι which includes k_ι number of regressors. Furthermore, let $\theta_\iota = (\beta_0, \beta_1, ..., \beta_{k_\iota}, \sigma)$ be the vector of model parameters and

$$L(Y|M_\iota, \theta_\iota) = \frac{1}{(\sqrt{2\pi}\sigma)^n} \exp\left(-\frac{\mathrm{RSS}_{M_\iota,\theta_\iota}}{2\sigma^2}\right) \tag{2.2}$$

be the likelihood of the data Y given the model M_ι and parameters θ_ι. Here

$$\mathrm{RSS}_{M_\iota,\theta_\iota} = \sum_{\iota=1}^{n} \left(Y_\iota - \beta_0 - \sum_{o=1}^{k_\iota} \beta_o X_{\iota o}\right)^2 \tag{2.3}$$

is the residual sum of squares.
In order to find the best model we could maximize the likelihood function $L(Y|M_\iota, \hat{\theta}_\iota)$ where $\hat{\theta}_\iota$ is the maximum likelihood estimator of θ_ι. However, when the number of regressors in a model grows the maximum of $L(Y|M_\iota, \hat{\theta}_\iota)$ never decreases, even if regressors that are completely irrelevant are included. It is therefore a customary approach to include some penalty for the model complexity. The most popular model selection criteria are Akaike's information criterion (AIC) (see Akaike 1974), which maximizes

$$\log L(Y|M_\iota, \hat{\theta}_\iota) - k_\iota \tag{2.4}$$

and the Bayesian information criterion (BIC) (see Schwarz 1978), which maximizes

$$\log L(Y|M_\iota, \hat{\theta}_\iota) - \frac{1}{2} k_\iota \log n. \tag{2.5}$$

The second terms of equations (2.4) and (2.5) are penalties, where model complexity is measured by the number of regressors k_ι.

When σ^2 is unknown, it's maximum likelihood estimator under the model M_ι is $\hat{\sigma^2} = \frac{\mathrm{RSS}_\iota}{n}$, where $\mathrm{RSS}_\iota = Y^T Y - n\hat{\beta}^T\hat{\beta}$ and $\hat{\beta} = \left(X_{M_\iota}^T X_{M_\iota}\right)^{-1} X_{M_\iota}^T Y$. Then we get the following

$$\log L(Y|M_\iota, \hat{\theta}_\iota) = \log\left\{ \frac{1}{\left(\sqrt{2\pi\frac{\mathrm{RSS}_\iota}{n}}\right)^n} \exp\left(-\frac{\mathrm{RSS}_\iota}{2\frac{\mathrm{RSS}_\iota}{n}}\right)\right\} \tag{2.6}$$

$$= -\frac{n}{2}\left(\log 2\pi + \log\frac{\mathrm{RSS}_\iota}{n} + 1\right) \tag{2.7}$$

and then (2.5) will be

$$\log L(Y|M_\iota, \hat{\theta}_\iota) - \frac{1}{2} k_\iota \log n = -\frac{1}{2}\left(n \log 2\pi + n \log \frac{\mathrm{RSS}_\iota}{n} + n + k_\iota \log n \right). \quad (2.8)$$

Finding the model that maximizes (2.8) is equivalent to finding the model M_ι that minimizes

$$\mathrm{BIC} = n \log \frac{\mathrm{RSS}_\iota}{n} + k_\iota \log n. \quad (2.9)$$

Piepho and Gauch (2001) examined different model selection criteria through simulation and found that BIC performed better than AIC in QTL mapping. However, according to Broman and Speed (2002), the number of QTLs are often overestimated when the original BIC is used as a selection criterion.

To explain the origin of BIC briefly, we let $f(\theta_\iota|M_\iota)$ be the density of the prior distribution for θ_ι and $\pi(M_\iota)$ be the prior probability of M_ι. We denote $L(Y|M_\iota)$ as the likelihood of the data given M_ι, that is

$$L(Y|M_\iota) = \int L(Y|M_\iota, \theta_\iota) f(\theta_\iota|M_\iota) d\theta_\iota.$$

The posterior probability of M_ι given the data is

$$P(M_\iota|Y) = \frac{\pi(M_\iota) L(Y|M_\iota)}{\sum_w \pi(M_w) L(Y|M_w)}. \quad (2.10)$$

The denominator of 2.10 is the sum of $\pi(M_w)L(Y|M_w)$ over all possible models M_w. Since the Bayesian rule selects the model when $P(M_\iota|Y)$ is maximized, we equivalently select the model M_ι when $\pi(M_\iota)L(Y|M_\iota)$ is maximized.
When using BIC we neglect the posterior probability $\pi(M_\iota)$ and approximate $\log L(Y|M_\iota)$ by (2.5) (see Schwarz 1978). However, if we neglect $\pi(M_\iota)$, then all models have the same prior probability, but the model dimension gets some non-uniform prior information. For example, if we have data with 110 possible regressors available, then the number of models that include 55 regressors is $\binom{110}{55} \approx 9.85 \cdot 10^{31}$ so that the prior probability of having 55 regressors in the model is $> 10^{21}$ times higher than having 7 regressors in the model.
In QTL mapping we most often have many regressors available but only a small portion of them affect the quantitative trait. We therefore need a model selection criterion that assigns higher prior probability to the events when there are few regressors in the model.

To extend the standard model selection criterion for QTL mapping Bogdan, Gosh and Doerge (2004) suggested a modified version of the BIC. This modified criterion includes a penalty for the model dimension that increases when the number of available markers increases. They suggest using the following prior probability

$$\pi(M_\iota) = p^{k_\iota}(1-p)^{m-k_\iota} \tag{2.11}$$

where m is the total number of possible regressors and p is the probability that a randomly chosen regressor affects the trait value. This gives

$$\log\pi(M_\iota) = m\log(1-p) - k_\iota\log\left(\frac{1-p}{p}\right).$$

Bogdan, *et al.* therefore recommend choosing a model that maximizes

$$\log L(Y|M_\iota,\hat{\theta}_\iota) - \frac{1}{2}k_\iota\log n - k_\iota\log\left(\frac{1-p}{p}\right). \tag{2.12}$$

When using (2.11) as a prior, the probability that the number of regressors k_ι is equal to k is $P(k_\iota = k) = \binom{m}{k}p^k(1-p)^{m-k}$, that is, it is binomially distributed with coefficients m and p and expected value $\mathrm{EN} = mp$. Then 2.12 can be rewritten as

$$\log L(Y|M_\iota,\hat{\theta}_\iota) - \frac{1}{2}k_\iota\log n - k_\iota\log\left(\frac{m}{EN} - 1\right). \tag{2.13}$$

Using (2.7) again, we obtain

$$\begin{aligned}&\log L(Y|M_\iota,\hat{\theta}_\iota) - \frac{1}{2}k_\iota\log n - k_\iota\log\left(\frac{m}{EN} - 1\right)\\ &= -\frac{1}{2}\left(n\log 2\pi + n\log\frac{\mathrm{RSS}_\iota}{n} + 1 + k_\iota\log n + 2k_\iota\log\left(\frac{m}{\mathrm{EN}} - 1\right)\right).\end{aligned}$$

Bogdan, *et al.* therefore propose selecting the model that minimizes the following quantitiy

$$\mathrm{mBIC} = n\ \log\ \mathrm{RSS} + k\ \log\ n + 2k\ \log(\eta - 1) \tag{2.14}$$

where n is the sample size, k is the number of regressors in the model and η is the penalty parameter defined as $\eta = m/\mathrm{EN}$. Here, m is the total number of markers in the model and EN is the expected number of QTL effects. When we have no prior information on the number of QTL, the standard choice EN= 4 is suggested by Bogdan, Gosh and Zak-Szatkowska (2008) based on frequentist arguments concerning the control of Type I error.

Selection of models from the model space

If we have a densely spaced marker map the number of possible models is enormous. For example, if there are 100 markers, not particularly dense, on the genome, then the number of possible models to investigate is $2^{100} \approx 10^{30}$, making it computationally impossible to visit all these models. Therefore, some efficient procedure for searching through the model space is required.
The simplest method of examining the search space is stepwise selection. If the number of markers is larger than the sample size, starting with forward selection is suitable. One starts with the model that only includes the intercept and no markers. Then, for each marker, the criterion of choice is calculated for all possible models that have one added marker. If the criterion is improved, the model that improves the criterion most is selected. This is repeated, adding one marker at a time and calculating the criterion, until the criterion cannot be improved any further. One may then proceed with backward elimination, which works similarly to forward selection except that the criterion is calculated for each model that has one marker deleted. Likewise, this is repeated until the criterion cannot be improved.
Although stepwise selection procedures are popular and can be useful, they very often fail to find the optimal model for the data even in relatively simple situations. This motivates the use of more sophisticated model search strategies, like the Genetic Algorithm introduced in the next section.

Chapter 3

Genetic Algorithms

3.1 Review

In QTL multiple regression, it is often hard to find the best model according to some criterion. We frequently face the situation that the value of the model selection criterion is similar for many different models, giving rise to the question whether it is appropriate to classify one model as the best. In this situation, Genetic Algorithms (GAs) can be used to solve the complicated optimization problem involved. Furthermore GAs do not only find one optimal model, but a whole population of good models. Due to the Bayesian origin of the model selection criterion mBIC, the GA can be used to compute model posterior probabilities and also posterior probabilities for individual markers.

3.1.1 Introduction

The GA (see Goldberg 1989, Obitko 1998 and Whitley 1994), which was inspired by the mechanisms of biological evolution, belongs to evolutionary computing, a subfield of artificial intellegence. It is a heuristic optimization algorithm that applies selection, recombination and mutation operators to a population in order to generate new individuals, similar to the mechanisms in natural evolution. Although the GA is computationally time consuming, it can be used for difficult problems, it is easy to implement and is less likely than other methods to get stuck in local extrema. A simple structure of the GA is shown in Figure 3.1.

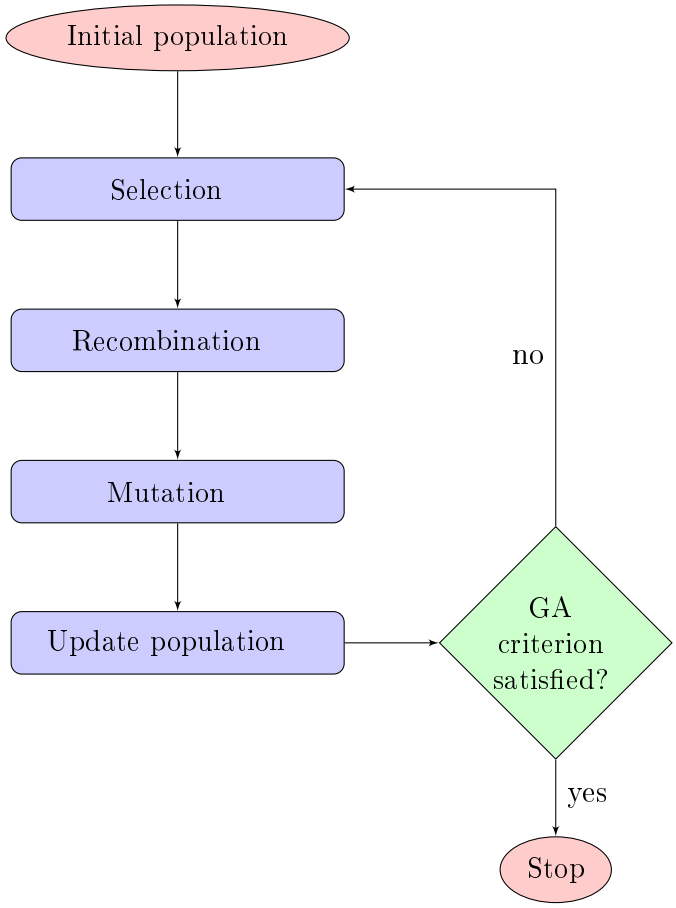

Figure 3.1: A simple outline of the Genetic Algorithm

3.1.2 The traditional Genetic Algorithm

Initiation

Genetic Algorithms work with a population of so called GA chromosomes. These are usually represented as binary strings, with one GA chromosome for each GA individual. Traditionally, the GA chromosomes consist of 0's and 1's but other encodings are also possible. We keep this representation for simplicity and as it fits our model selection problem. In Section 3.2 we will outline in detail how linear models for QTL mapping can be encoded as GA chromosomes. The initial GA population is most often generated randomly. Each GA chromosome is evaluated and assigned a fitness value from some fitness function of choice, which we might either want to maximize or minimize.

Selection

A certain amount of parents are selected, usually two, according to their fitness. That is, the better their fitness value, the higher the probability is that they are selected. This operation is intended to simulate Darwin's theory of natural selection. Various selection methods exist. In roulette-wheel selection each GA chromosome in the GA population is drawn with a probability proportional to its fitness. An alternative strategy is tournament selection, where a random sample of GA chromosomes is drawn, and the fittest among them is selected as a parent.

Recombination

Recombination is carried out with probability p_r, which is usually set between $65\% - 95\%$. The most common method of recombination is to randomly select a position on the GA chromosomes for recombination, partition the GA chromosomes at that position and then swap the fragments between the two GA parents (see Figure 3.2). After recombination has been performed, the GA chromosome with better fitness is chosen as a so called GA child. If no recombination is done, the GA parent with the better fitness is chosen.

Mutation

Mutation is performed on the new GA child with probability p_m. In case of muation, one bit of the string, which is selected randomly, is changed from 0 to 1, or from 1 to 0. The mutation probability is usually set very low, somewhere between 0.5% and 5%.

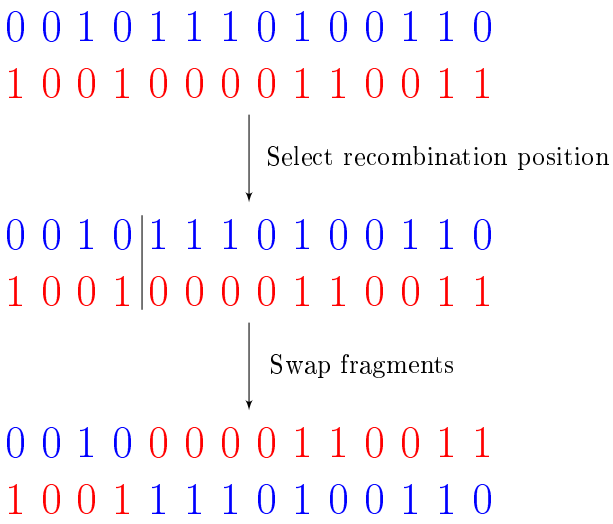

Figure 3.2: Simple version of recombination

Updating the population

After performing selection, recombination and mutation, one checks if the GA child improves the fitness of the GA population. If it does, the GA child is added to the GA population, if not, the GA child is discarded. These steps are iterated until some stopping rule applies or some iteration limit has been reached.

Previous research

Genetic Algorithms for QTL mapping have recently been studied by various authors. Nakamichi, Ukai and Kishino (2001) simulated intercross data and used a Genetic Algorithm to identify QTLs. They base their detection of QTLs on the final best solution of the Genetic Algorithm using Akaike's information criterion to evaluate the fitness of each solution.
Carlborg, Andersson and Kinghorn (2000) simulated intercross data and used a real data set to examine the performance of a Genetic Algorithm, which included local improvement strategy. They used the residual sum of squared errors from a weighted least squares approach as fitness function and reported the number of times the best solution was found by their algorithm.
Namkung, Nam and Park (2006) used Genetic Algorithms to identify expression quantitative trait loci (eQTLs) in real data which consisted of 1692 SNPs.

They also included local improvement in their Genetic Algorithm and their fitness function was $-\log$(p-value) where the p-value was found by performing analysis of variance.
Zhu and Chipman (2006) suggested to run a number of traditional GAs in parallel without allowing each GA to fully converge. The information gained by the parallel GA runs was then combined and the most important variables, according to the majority vote across all GA runs, returned. The resulting algorithm, called Parallel Genetic Algorithm (PGA), has been found to deliver relatively good results.
Mukhopadhyay, George and Xu (2010) also investigated Genetic Algorithms by simulating four different scenarios of backcross data using PGA suggested by Zhu and Chipman. They based their results on one final solution using a generalized cross validation criterion as fitness function.

One novelty of our approach is the fitness criteria and how we make use of the results of the GA. When using Genetic Algorithms, we do not only find the optimal solution but also a vast population of good models ranked according to the modified Bayesisan Information Criterion (mBIC). This allows us to estimate the posterior probability of the models in the population and in particular to give posterior probabilities for each marker similar to LOD scores. These marker posterior probabilities provide intuitive plots, allowing one to determine potential QTLs.

3.2 Application to QTL Mapping

The GA can be applied to any optimization problem which can be formulated using binary strings. For the purpose of QTL mapping, we can denote the markers entering a specific model using a binary string, that is, each digit of the binary string represents a marker, 1 signifying that the marker was included in the model and 0 that the marker was not included in the model. For example, if we have a genetic map with 10 markers and we consider a model including markers 2, 6 and 9, then the corresponding string would be 0100010010. For more convenience and computational efficiency, each individual can be represented by the set of markers which have entered the model. That is, $S = \{2, 6, 9\}$ in our example.

Fitness function

The fitness function should quantify the qualitiy of a selected model. We will make use of the model selection criterion mBIC discussed in Section 2.2.2. The properties of mBIC as a model selection criterion in QTL mapping has been studied comprehensively by Bogdan *et al.* (2004, 2008) and Baierl *et al.* (2006). Based

on this previous research, it is reasonable to use mBIC as a fitness function, where a string is rated better the smaller its mBIC value.

Initiation

In our application of GA, an initial model M_I was obtained using stepwise selection with mBIC as a selection criterion for finding significant markers. We started with forward selection which begins with a model that includes no markers. The markers which minimize mBIC are then added consecutively until the selection criterion cannot be decreased. We then try to add up to five extra markers and if in this way a new minimum is found we continue with forward selection. Otherwise no additional markers are included, and backward elimination is performed. Here unimportant markers are eliminated repeatedly until no improvement of the criterion can be made. The described forward and backward steps are then iterated. This search procedure is described in detail in Baierl, *et al.* (2006).
To produce an initial population for the algorithm, let m_c denote the number of markers on chromosome c and k_c the number of markers on chromosome c selected in the candidate model M_I. Here $c = 1, ..., l$, where l is defined as the total number of chromosomes. The relative frequency of selected markers on each chromosome is denoted as

$$p_c = \frac{k_c}{m_c}.$$

If there were no markers on chromosome c included in M_I, that is, if $k_c = 0$, then we set

$$p_c = \frac{1}{4 \cdot m_c}. \tag{3.1}$$

Although there are no detected markers on chromosome c in M_I the chance of including a marker on that chromosome should not be zero. However, the probability should be smaller than when there are markers present in the model. The expected value for the number of markers on chromosome c is $m_c \cdot p_c$ which we set to $\frac{1}{4}$ for no detected markers in M_I, motivating the choice of 3.1.
Let u be the size of the inital population. We generate u different models M_υ, where $\upsilon = 1, ..., u$. For the first half of the population, each marker was included in the model M_υ with probability p_c. Each marker is then a Bernoulli random variable with probability p_c.
For the second half of the population we performed the same initiation steps as before but with probability $2 \cdot p_c$ that each marker is in model M_υ.
Finally, before a string was added to the initial population local improvement was performed, which will be introduced in the next section.

Local Improvement

In our specific implementation of the GA for QTL mapping, the application of local improvement search strategies plays an important role in various stages of the GA. The purpose of using local improvement is to improve the convergence of the GA and to increase the quality of GA children. Furthermore, we are interested in finding many good models and the local improvement search strategy serves that purpose.
In experimental crosses, neighbouring markers are usually highly correlated (see Section 2.1). Accordingly, neighbouring markers behave quite similarly as regressors, which motivates the following search strategy. Consider the string $S = \{s_1, s_2, ..., s_k\}$, where k is the number of markers in the model, $s_\tau \in \{1, ..., m\}$, m is the number of possible markers and $\tau = 1, ..., k$. Having calculated the mBIC value of the string S, we try for each marker s_τ that is present in the model to improve mBIC by replacing it with a neighbouring marker while keeping all other markers fixed. We start by calculating mBIC for $S_{\tau,-1} = \{s_1, ..., s_\tau - 1, ..., s_k\}$. If this substitution improves mBIC we continue shifting the marker, that is, we calculate mBIC for $S_{\tau,-2} = (s_1, ..., s_\tau - 2, ..., s_k)$. We continue in this fashion until the mBIC value cannot be improved any further. We then apply the same method in the opposite direction, starting with $S_{\tau,+1} = (s_1, ..., s_\tau + 1, ..., s_m)$ and continue until mBIC is minimized. This procedure is applied to all markers in S. Occasionally, two markers will be merged in which case we continue with a model that has one less marker. For relatively small k, the local improvement is computationally feasible. However, for large models, it might be necessary to perform local improvement for only a subset of the markers. Figure 3.3 shows an example of local improvement for one marker.

Selection

The selection of parents which produce a new individual was performed with tournament selection. A random sample of individuals of size t is selected from the population. Each individual of the population has the same chance to become a member of the tournament group. Within the tournament group the individual with largest fitness is chosen. This procedure is then repeated until a second parent has been found which is different from the first one. The size t of the tournament group is a parameter which has to be decided upon.

Recombination

Recombination was performed with probability p_r. However, the classical recombination strategy was abandoned and substituted by the following procedure: We started with the parents S_1 and S_2 found with the tournament selection procedure

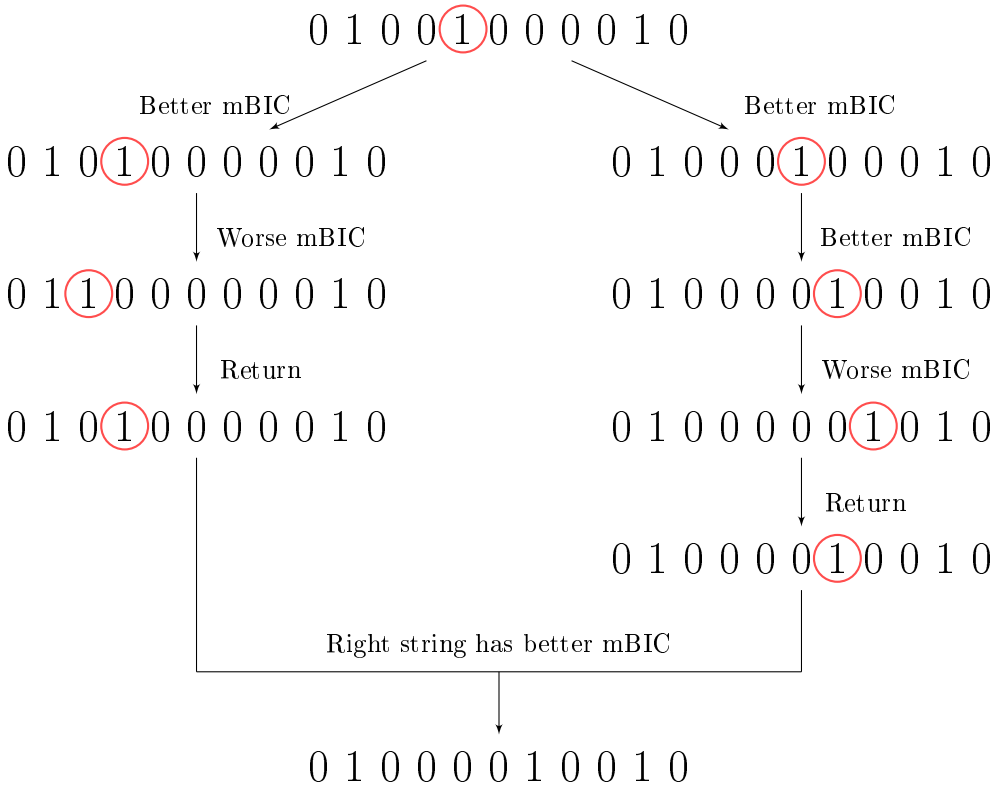

Figure 3.3: Local improvement for marker at position 5

explained above. As before, these parents can be considered as sets of positive integers according to which markers are included in the models. For example, the binary string [00010000010] is represented by the set $S = \{4, 10\}$, meaning that markers 4 and 10 are present in the model. We define the following

$$\text{Intersection: } S_I = S_1 \cap S_2$$

$$\text{Union: } S_U = S_1 \cup S_2$$

$$\text{Difference: } S_D = (S_1 \setminus S_2) \cup (S_2 \setminus S_1)$$

Our recombination step consists of a revised version of forward and backward selection, which always included the markers of the intersection S_I of the two parents S_1 and S_2 which were selected before. For the forward selection version, we started

with the intersection of markers S_I and then tried to improve the fitness function of the string by adding iteratively markers from S_D. The resulting best model of the modified forward selection is called S_F. In the modified backward version we started with the union of markers S_U and then tried to improve the fitness function by deleting iteratively markers from S_D. The resulting model of the modified backward elimination is called S_B. Finally, we calculate the fitness of the strings S_F and S_B and choose the one with the better fitness as the final string.
The previously described method is performed with probability p_r. If no recombination is performed, the parent (S_1 or S_2) with higher fitness value is chosen.

Example

Assume that the following parents were chosen with tournament selection:

$$S_1 = 1, 5, 16, 47, 52$$

$$S_2 = 1, 8, 14, 39, 52, 72$$

The union, intersection and difference are then $S_U = 1, 5, 8, 14, 16, 39, 47, 52, 72$, $S_I = 1, 52$ and $S_D = 5, 8, 14, 16, 39, 47, 72$ respectively. The modified version of forward selection starts with S_I, and only adds markers from S_D. The fitness function is denoted here as f and we assume that it is optimized when f is at its minimum.
Assume that the modified forward selection resulted in the string

$$S_F = 1, 14, 52, 72.$$

with fitness $f_F = 125$ and that the modified backward elimination resulted in the string

$$S_B = 1, 16, 39, 52, 72.$$

with fitness $f_B = 203$.
As a result, S_F is chosen since it has lower fitness value than S_B.

Mutation

Mutation was performed with probability p_m. When mutation occured, two different directions could be taken: either insertion or deletion of a marker from the string. Insertion and deletion were performed with equal probability. When adding a marker to the string we examined if there were any chromosomes that did not have any markers in the model. If such chromosomes existed, mutation was performed on one of them with equal probability of being selected. If there was an existing marker on all of the chromosomes, one chromosome was selected, with

equal probability, but the mutation site was kept at least 30 cM markers distant from other existing markers. In the case of deletion, one of the markers in the string was removed with equal probability.

The reason why the recombination probability was not set to $p_r = 1$ is that we don't want to exclude the possibility that a string from the population is changed only through mutation without previous recombination.

Updating the population

Recombination and mutation yield a new model, for which local improvement is performed. Fitness of the resulting model is compared with the fitness of the least fit member of the population. If the fitness of the new string is better, then the worst indvidual from the population is substituted by the new one. If the new individual does not improve the fitness of the population, no changes are made. Selection, recombination, mutation and updating comprises one iteration of the GA The algorithm terminates according to the following stopping criterion: We count the number of consecutive iterations for which no improvement of the population occured or when the new string is not among the B best models of the population. If the new string is among the B best models then the counter is set to zero. The parameter B will be called *best model index*. The algorithm terminates when the counter reaches the iteration limit I. Throughout all simulations, the stopping criterion parameter I was set to 1000.
A simple flow chart of our Genetic Algorithm is displayed in Figure 3.4.

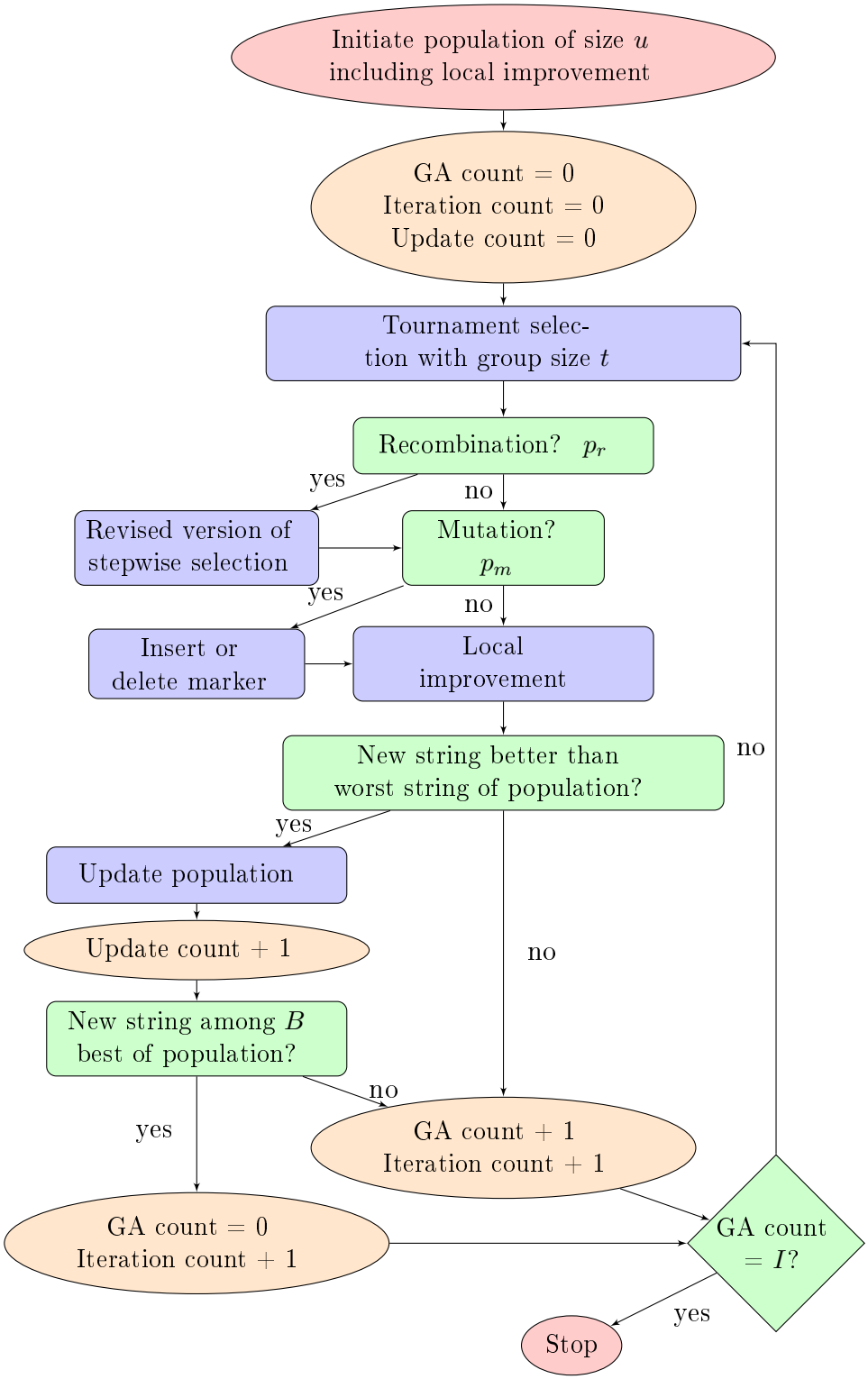

Figure 3.4: Our version of the Genetic Algorithm. Parameters to be tuned are u, t, p_r, p_m, B and I

Pool of models

Our aim is to investigate as many good models as possible. Therefore, whenever a mBIC value was calculated for a string, it is added to the so called pool of models, provided that the string did not exist in the pool already. To make sure that there were no duplicates in the pool we stored all of the visited models in a hashtable with the model string as a key and its mBIC value associated to that key. When using a hashtable, a hash function is used to calculate an index from the key and then the key and its associated value is added to an array according to this index. After running the GA we ended up with a rather large pool of models which could be used for calculating posterior probability for models as well as for markers.

Posterior probability

In Bayesian statistics, the posterior probability of a model M_ι is defined as

$$\mathrm{P}(M_\iota|Y) = \frac{\mathrm{P}(Y|M_\iota)\pi(M_\iota)}{\sum_{M\in\mathcal{M}} \mathrm{P}(Y|M)\pi(M)}$$

where $\pi(M)$ is the prior probability of model M and $\mathcal{M}$ is the set of all possible models. mBIC is derived as an approximation of the model posterior under the model prior $\pi(M_\iota) = p^{k_\iota}(1-p)^{m-k_\iota}$ (see Section 2.2.2). In particular the approximation

$$\mathrm{mBIC}(M_\iota) \approx -2\log\left[\mathrm{P}(Y|M_\iota)\cdot\pi(M_\iota)\right] + C$$

holds, where C is a constant (see Bogdan, *et al.* (2004)).
We therefore have

$$\exp\left\{-\frac{\mathrm{mBIC}(M_\iota)}{2}\right\} \approx \mathrm{P}(Y|M_\iota)\cdot\pi(M_\iota)\cdot\exp(-\frac{C}{2})$$

and

$$\begin{aligned}
\frac{\exp\left\{-\frac{\mathrm{mBIC}_{(M_\iota)}}{2}\right\}}{\sum_{M\in\mathcal{M}}\exp\left\{-\frac{\mathrm{mBIC}_{(M)}}{2}\right\}} &\approx \frac{\mathrm{P}(Y|M_\iota)\cdot\pi(M_\iota)\cdot\exp(-\frac{C}{2})}{\sum_{M\in\mathcal{M}}\mathrm{P}(Y|M)\cdot\pi(M)\cdot\exp(-\frac{C}{2})} \\
&= \frac{\mathrm{P}(Y|M_\iota)\cdot\pi(M_\iota)}{\sum_{M\in\mathcal{M}}\mathrm{P}(Y|M)\cdot\pi(M)} \\
&= \mathrm{P}(M_\iota|Y).
\end{aligned}$$

The Genetic Algorithm computes mBIC for a large number of models, and if many models of good quality have entered the pool it becomes reasonable to approximate

the model posterior probability based on the models of the pool. In particular we suggest that

$$\begin{aligned}\sum_{M\in\mathcal{M}} \mathrm{P}(Y|M)\cdot\pi(M) &\approx \sum_{M\in\mathcal{M}^{Pool}} \mathrm{P}(Y|M)\cdot\pi(M) \\ &\approx \sum_{M\in\mathcal{M}^{Pool}} \exp\left\{-\frac{\mathrm{mBIC}(M)}{2}\right\}\cdot\exp(\frac{C}{2}) \qquad (3.2)\end{aligned}$$

where $\mathcal{M}^{Pool}$ is the set of all models that are in the Pool.
To compute posterior probabilities for markers we define $\mathcal{M}_j$ as the set of all models containing marker j, denoted by μ_j, where $j = 1, ..., m$. Then the posterior probability for marker j is calculated as

$$\mathrm{P}(\mu_j = 1|Y) = \sum_{M\in\mathcal{M}_j} \mathrm{P}(M|Y), \qquad (3.3)$$

that is, the sum of the posterior probability of all models containing marker j. Then it holds that

$$\begin{aligned}\mathrm{P}(\mu_j = 1|Y) &= \sum_{M\in\mathcal{M}_j} \mathrm{P}(M|Y) = \frac{\sum\limits_{M\in\mathcal{M}_j} \mathrm{P}(Y|M)\pi(M)}{\sum\limits_{M'\in\mathcal{M}} \mathrm{P}(Y|M')\pi(M')} \\ &= \frac{\sum\limits_{M\in\mathcal{M}_j^{Pool}} \mathrm{P}(Y|M)\pi(M) + \sum\limits_{M\in\mathcal{M}_j\setminus\mathcal{M}_j^{Pool}} \mathrm{P}(Y|M)\pi(M)}{\sum\limits_{M'\in\mathcal{M}^{Pool}} \mathrm{P}(Y|M')\pi(M') + \sum\limits_{M'\in\mathcal{M}\setminus\mathcal{M}^{Pool}} \mathrm{P}(Y|M')\pi(M')}\end{aligned}$$

where $\mathcal{M}_j^{Pool}$ is the set of models that are in the Pool and contain marker j. $\mathcal{M}_j \setminus \mathcal{M}_j^{Pool}$ is then the set of models containing marker j that are not in the pool and $\mathcal{M} \setminus \mathcal{M}^{Pool}$ is the set of all models that are not in the pool.
In analogy to (3.2) we would like to approximate the marker posterior probability with

$$\mathrm{P}\left(\mu_j = 1|Y\right) \approx \frac{\sum\limits_{M\in\mathcal{M}_j^{Pool}} \mathrm{P}(Y|M)\pi(M)}{\sum\limits_{M'\in\mathcal{M}^{Pool}} \mathrm{P}(Y|M')\pi(M')} \approx \frac{\sum\limits_{M\in\mathcal{M}_j^{Pool}} \exp\left\{-\frac{\mathrm{mBIC}_{(M)}}{2}\right\}}{\sum\limits_{M'\in\mathcal{M}^{Pool}} \exp\left\{-\frac{\mathrm{mBIC}_{(M')}}{2}\right\}}. \qquad (3.4)$$

The quality of this approximation was investigated by Frommlet, *et al.* (2012) using simulation studies. Estimating QTL locations using the posterior probabilities gave better results than using only the best model found by the GA. Furthermore, they showed that more accurate estimates are made in a shorter time than with Zhu and Chipmans's (2006) PGA.

Chapter 4

Simulations

4.1 Genetic data simulations

Simulations were carrried out in Matlab for a backcross population using two different scenarios with equidistant markers. QTL and marker genotypes were simulated on $l = 10$ chromosomes with $k = 11$ markers equally distributed on each of them, with distance $d = 10$cM between the markers. The first marker was placed at genetic position 0 and the last at genetic position 100 on the chromosome. The total number of markers was $l \cdot k = 110$.

Marker genotypes were simulated, starting with the left end genotypes of each chromosome, denoted L_c, $c = 1, ..., l$, with equal probability of a chromosome starting with 0 (genotype AA) or 1 (genotype Aa). Recombination sites were then simulated. Since the distance between each marker is 10cM, the expected number of crossovers between marker at position 0 and marker at position 100 is 1. As discussed in Section 2.1.4 the number of crossovers on each chromosome follows a poisson process. Let X_R be the number of crossovers between marker at position 0 and marker at position 100. Then

$$\mathrm{P}(X_R = \kappa) = \frac{e^{-100\lambda}\,(100\lambda)^{\kappa}}{\kappa!}, \qquad \kappa = 0, 1, 2, ...$$

where $\lambda = \frac{1}{100}$ since the genetic distance is measured in cM.

We started by simulating the first crossover site $R_{c,1}$, $c = 1, ..., l$, where $R_{c,1} \sim \mathrm{Exp}\,(1/100)$. The second crossover site $R_{c,2}$ was simulated with $R_{c,2} = R_{c,1} + r_{c,2}$, where $r_{c,2} \sim \mathrm{Exp}\,(1/100)$. The following crossover locations were obtained with $R_{c,v} = R_{c,v-1} + r_{c,v}$, for $v = 2, .., \rho_c$, with $r_{c,v} \sim \mathrm{Exp}\,(1/100)$ and $c = 1, ..., l$. Here ρ_c is determined by the first crossover site that exceeds the last marker location. Recombination between two markers takes place when an odd number of crossovers

occurs between them.

Given the number of crossovers between markers, the marker genotype immediately follows. The value of the first genotype (0 or 1) of each chromosome is repeated until a crossover site is reached. Then the genotype is changed (to either 0 or 1) and repeated until the next crossover site. This procedure is continued until a genotype has been assigned to all markers.

Example

Assume that we want to simulate the marker genotypes for one individual with one chromosome which has 11 markers with 10cM distance. The genotype of the first marker on the left end of the chromosome is randomly selected with equal probability to be 0 or 1. Assume that

$$L_1 = 1.$$

The crossover sites are then simulated according to

$$R_{1,1} \sim \text{Exp}\,(1/100)$$

and

$$R_{1,v} = R_{1,v-1} + r_{1,v}, \quad \text{for } v = 2, ..., \rho_c$$

where $r_{1,v} \sim \text{Exp}\,(1/100)$. If the resulting crossover site vector is for example

$$R_1 = [16.9311, 52.5235, 90.9888, 165.1561]$$

then marker genotypes for the chromosome are

$$[1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 0].$$

A marker map for the example is displayed in Figure 4.1. Marker locations are shown in grey below the map, genotypes are above the map and recombination sites are displayed with red asterisks.

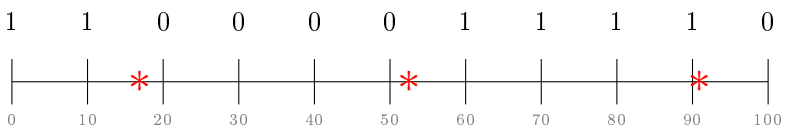

Figure 4.1: Marker map displaying genotypes and crossover locations

The genotypes of the QTLs are determined in a similar fashion as for the markers. If a QTL is located directly at a marker its genotype is the same as the marker genotype. For example, if QTLs were located at positions 20 and 70 in the example above, the genotypes would be [0 1].
If however a QTL is located between markers, the crossover sites are needed to determine the genotype. First, the genotype of the left flanking marker of the QTL is observed (0 or 1). If a crossover site is present between the left flanking marker and the QTL, then the genotype is changed (to 1 or 0). Assuming that there is a QTL at location 55 in the example above, then its genotype would be 1 since there is a crossover site at 52.5235.
Note that it is possible that the left and right flanking markers have the same genotype but a QTL between them has the opposite genotype. This happens, for example, when there are two crossovers between the markers, one on each side of the QTL (see Figure 4.2).

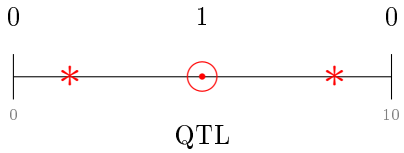

Figure 4.2: Two markers at positions 0 and 10 with QTL between them and crossover sites on each side of the QTL.

To simulate the quantitative trait, the sum of the effect size of each QTL which had Aa genotype and a random environmental noise was taken. That is, for the quantitative trait y_i

$$y_i = \sum_g \beta_g \cdot G_{i,g} + \epsilon_i$$

for each individual i, where g is the number of QTLs, β_g is the effect size of QTL g, $G_{i,g}$ is the genotype of QTL g (either 0 or 1) and ϵ_i the environmental noise with $\epsilon_i \sim \mathrm{N}(0, 1)$.

Two different scenarios were simulated in order to obtain the backcross populations. The first scenario is the same as scenario 12 from BOGDAN, *et al.* (2004). QTLs were placed on the chromosomes at the following positions [chromosome, position in cM, β (size of the effect)]: (1,20,0.76), (1,60,0.76), (2,20,0.76), (2,60,-0.76), (3,40,0.76), (4,20,0.76) and (5,0,0.76) (see Figure 4.3).

For the second scenario, the QTLs were positioned to examine the performance of the GA for a more complex model. The following positions and effect sizes were chosen for the QTLs: (1,30,1), (1,90,0.75), (2,25,1), (2,85,0.75), (3,5,1), (3,25,1), (4,5,1), (4,25,-1), (5,10,1), (5,45,1) and (5,70,1) (see Figure 4.4).

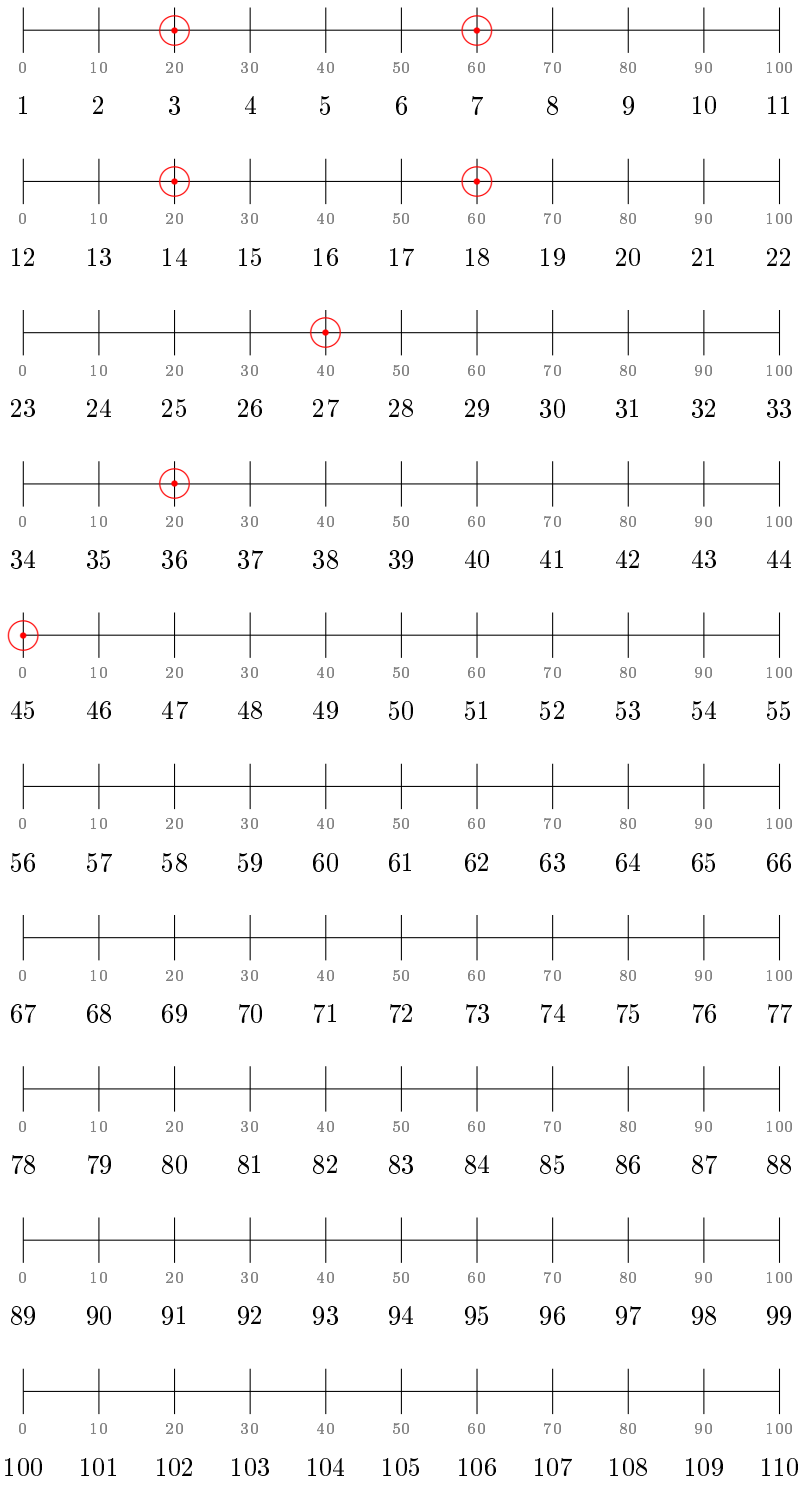

Figure 4.3: Scenario 1. Genetic locations are displayed in gray, markers 1-110 are displayed below the genetic location and QTLs are marked as circles along the chromosomes.

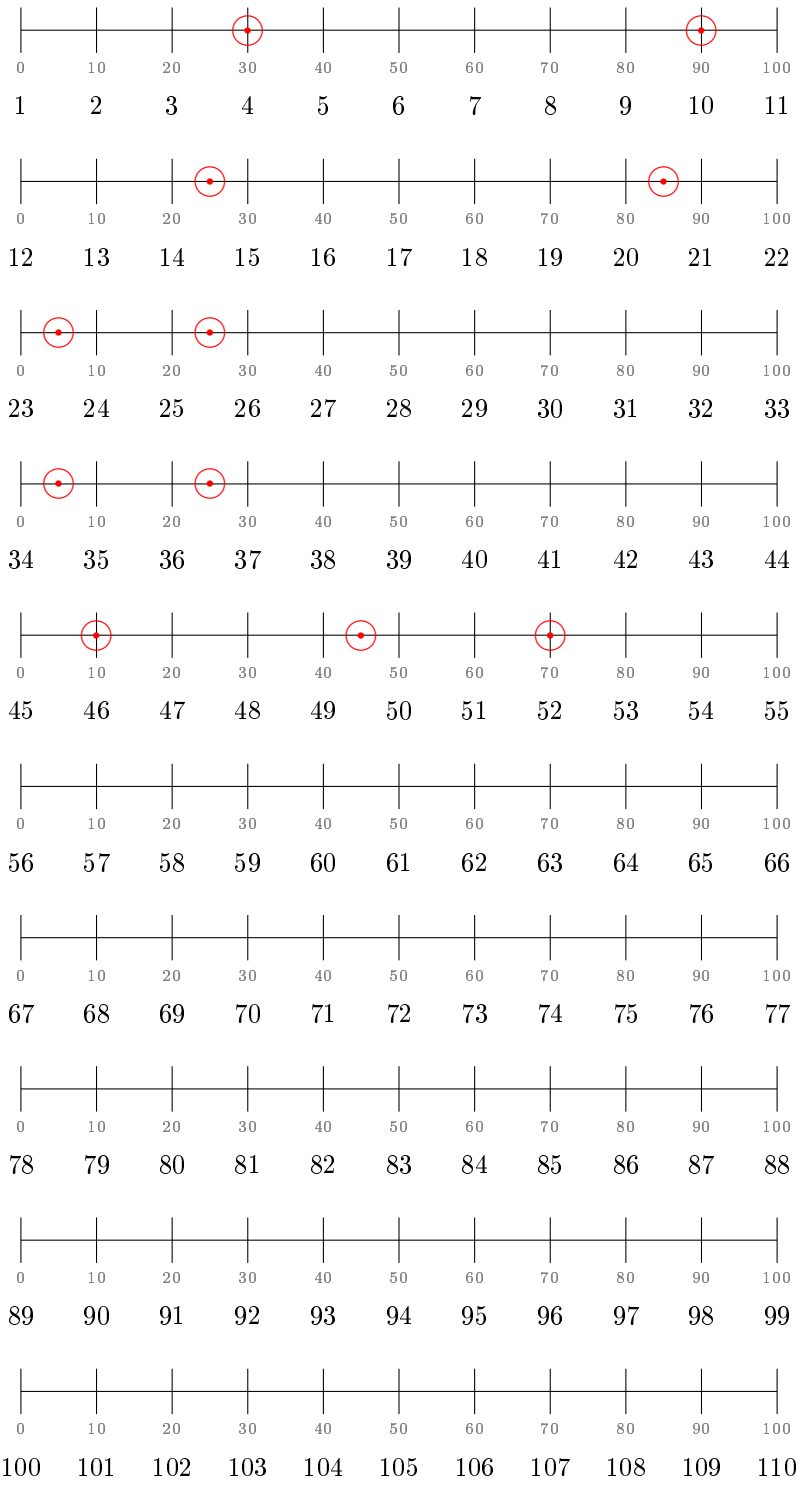

Figure 4.4: Scenario 2. Genetic locations are displayed in gray, markers 1-110 are displayed below the genetic location and QTLs are marked as circles along the chromosomes.

4.2 Genetic Algorithm simulations

4.2.1 Scenario 1

There are several parameters in the Genetic Algorithm that have to be tuned, namely, the recombination probability p_r, the mutation probability p_m, the size of the GA population u, the size of the tournament selection group t and the best model index B (see Figure 3.4). The number of iterations I without improvement of the population was kept fixed at $I = 1000$. All other parameter settings were systematically investigated. To analyze the performance of the GA for these different parameter settings we simulated 100 different data sets and applied GA to each of them with various parameter settings. When examining a particular parameter, we fixed all other parameters at default values which are typically used in the GA community, while the parameter in question was varied. For scenario 1 the default values were $p_r = 0.80$, $p_m = 0.050$, $u = 200$, $t = 2$ and $B = 10$.

Table 4.1 shows the systematic variation of individual parameters performed in the simulation study. Note that settings 3, 9, 13, 16 and 23 all contain the default parameter values. The set of parameter settings where one parameter varies while all others are kept fixed is called a parameter setting group. Five parameter setting groups were investigated.

Setting	p_r	p_m	u	t	B
1	**0.60**	0.050	200	2	10
2	**0.70**	0.050	200	2	10
3	**0.80**	0.050	200	2	10
4	**0.90**	0.050	200	2	10
5	**0.95**	0.050	200	2	10
6	**0.99**	0.050	200	2	10
7	0.80	**0.010**	200	2	10
8	0.80	**0.025**	200	2	10
9	0.80	**0.050**	200	2	10
10	0.80	**0.075**	200	2	10
11	0.80	**0.100**	200	2	10
12	0.80	0.050	**100**	2	10
13	0.80	0.050	**200**	2	10
14	0.80	0.050	**500**	2	10
15	0.80	0.050	**1000**	2	10
16	0.80	0.050	200	**2**	10
17	0.80	0.050	200	**3**	10
18	0.80	0.050	200	**5**	10
19	0.80	0.050	200	**7**	10
20	0.80	0.050	200	**10**	10
21	0.80	0.050	200	2	**1**
22	0.80	0.050	200	2	**5**
23	0.80	0.050	200	2	**10**
24	0.80	0.050	200	2	**20**

Table 4.1: Parameter settings examined for scenario 1

Runtime, pool size, iterations and updates

We start by reporting the mean and standard deviation of the runtime (measured in seconds), pool size, total number of iterations and total number of GA updates for each parameter setting based on 100 different simulation runs. Figures 4.5 and 4.6 display the change of mean runtime and mean poolsize between the different parameter settings.
It can be seen from those two figures that runtime and pool size show a somewhat similar behaviour. However, the increase of pool size for settings 12-15 is much more extreme than that of runtime.
The first line in Figures 4.5 and 4.6 display mean runtime and mean pool size respectively, when the recombination rate changes, that is, mean runtime and mean pool size for settings 1 - 6. Figure 4.5 shows that runtime clearly increases when

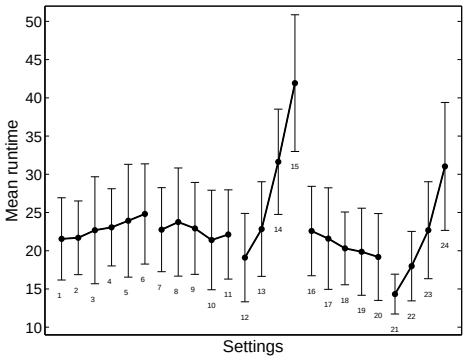

Figure 4.5: Mean runtime and standard deviation for 24 different parameter settings

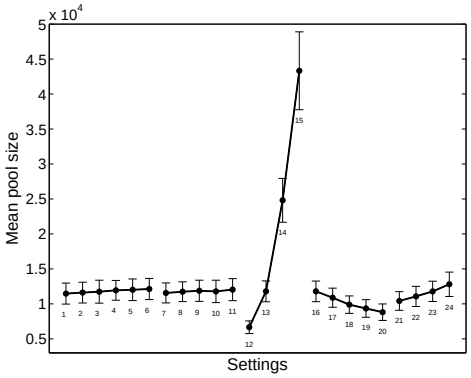

Figure 4.6: Mean pool size and standard deviation for 24 different parameter settings

the recombination probability is raised. The pool size also increases as can be seen in Figure 4.6. The reason for this increase is that if neither recombination nor mutation is performed, a GA iteration step will occur without finding a new string. The chances of such empty iterations are higher when the recombination rate is lower and since the mutation rate is quite low by default. Therefore, the stopping criterion can be reached quicker explaining why the runtime is smaller when the recombination rate is lower. The same applies to the pool size. Since no new model is added to the pool when an empty iteration occurs, the pool size is smaller when the recombination rate is lower.

The second line represents settings 7 - 11, or changes in the mutation rate. Figure 4.5 shows that the runtime is generally lower when the mutation rate is set higher. During recombination a rather good string is usually selected due to the stepwise selection inspired procedure. However, when mutation is performed, the mBIC value of the string typically deteriorates. The string is nevertheless often still good enough to make it to the population but might not be in the top B number of models. The stopping criterion is therefore reached quicker resulting in a lower runtime. Despite lower runtime, the pool size increases when the mutation rate is set higher. This is due to the fact that during mutation, strings that are new to the pool are frequently generated and during the ensuing local improvement, even more new strings are added to the pool.

The third line displays mean runtime and mean pool size when population size changes (settings 12 - 15). It is quite obvious that runtime and pool size grow when population size is larger. Increasing population size means increasing variety. The probability of selecting the same parents repeatedly is lower and therefore finding a new unique string is more likely, making it harder to reach the stopping criterion and consequently increasing the runtime. The pool size is higher for the

same reason but increases much more drastically due to the application of local improvement. For every new string numerous other strings can be added to the pool.
The fourth line shows how mean runtime and mean pool size react to changes in the number of individuals chosen in the tournament selection, that is, settings 16 - 20. Runtime and pool size decrease as the number of tournament individuals is set higher. As discussed in Section 3.2, t random members of the current population are chosen for the tournament group and then the string with the lowest mBIC is chosen from that group as a parent. When the tournament group is bigger, the probability that the same good strings are chosen reapeatedly as parents is higher causing loss in variety and therefore earlier GA termination. Therefore, runtime and pool size decrease when the number of individuals in the tournament group is increased.
The last line displays the mean runtime and mean pool size when the best model index B varies. As expected, runtime and pool size both increase when B is larger. The reason is that for smaller values of B it is harder to find a string that is among the top B strings.

Figures 4.7 and 4.8 show the mean and standard deviation of the total number of iterations and the total number of GA updates respectively (see "Iteration count" and "Update count" in Figure 3.4).

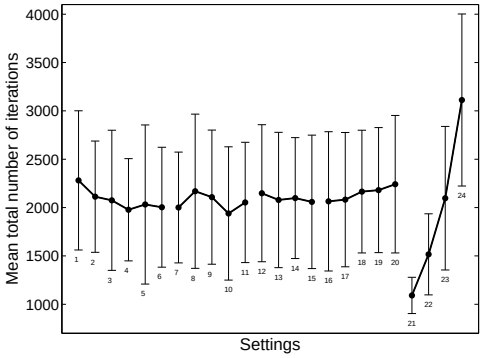

Figure 4.7: Mean total number of iterations and standard deviation for 24 different parameter settings

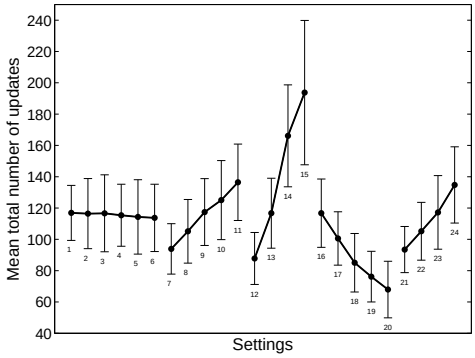

Figure 4.8: Mean total number of updates and standard deviation for 24 different parameter settings

When recombination probability is increased (settings 1-6), both the mean total number of iterations and the mean total number of updates decreases. As mentioned before, the probability of an empty iteration is higher when the recombination rate is lower. Therefore, some empty iterations may occur between iterations that result in updates of the population. For that reason, the number of

iterations grows with an increasing recombination rate. Even if no recombination occurs, mutation might still be performed. The mutated string can be good enough to update the population, despite not being as good as some strings produced by the recombination operator. Therefore, the number of updates is higher when the recombination rate is lower.
The number of iterations shows no clear trend as the mutation rate is increased (settings 7-11). Comparing Figures 4.5 and 4.7, it can be seen that runtime and the number of iterations behave almost identically for different mutation rate settings. The total number of updates shows a clear ascent as the mutation rate is increased for the same reason as for the increase of pool size.
As population size increases (settings 12-15), the number of iterations slightly shrinks but the number of updates grows. This might be because of a larger quantitiy of empty iterations for small population sizes which results from the fact that the probability of identical parents to be chosen is higher when the population size is lower.
The number of iterations increases with increasing tournament group size t (settings 16-20) while the number of updates shows a clear decrease. As for the population size, the probability that the same parents are chosen repeatedly is increased when the tournament group is larger causing increase in the number of iterations but loss in updates.
In analogy to runtime and pool size, the number of iterations and the number of updates both increase when the best model index B is set higher (settings 21-24).

Posterior probability plots

The posterior probability for each marker was computed according to equation (3.4). Figure 4.9 displays the average of the marker posterior over all 100 simulation runs for different parameter settings. Each graph contains the average posterior probability of each marker for all parameter settings with a single varying parameter.

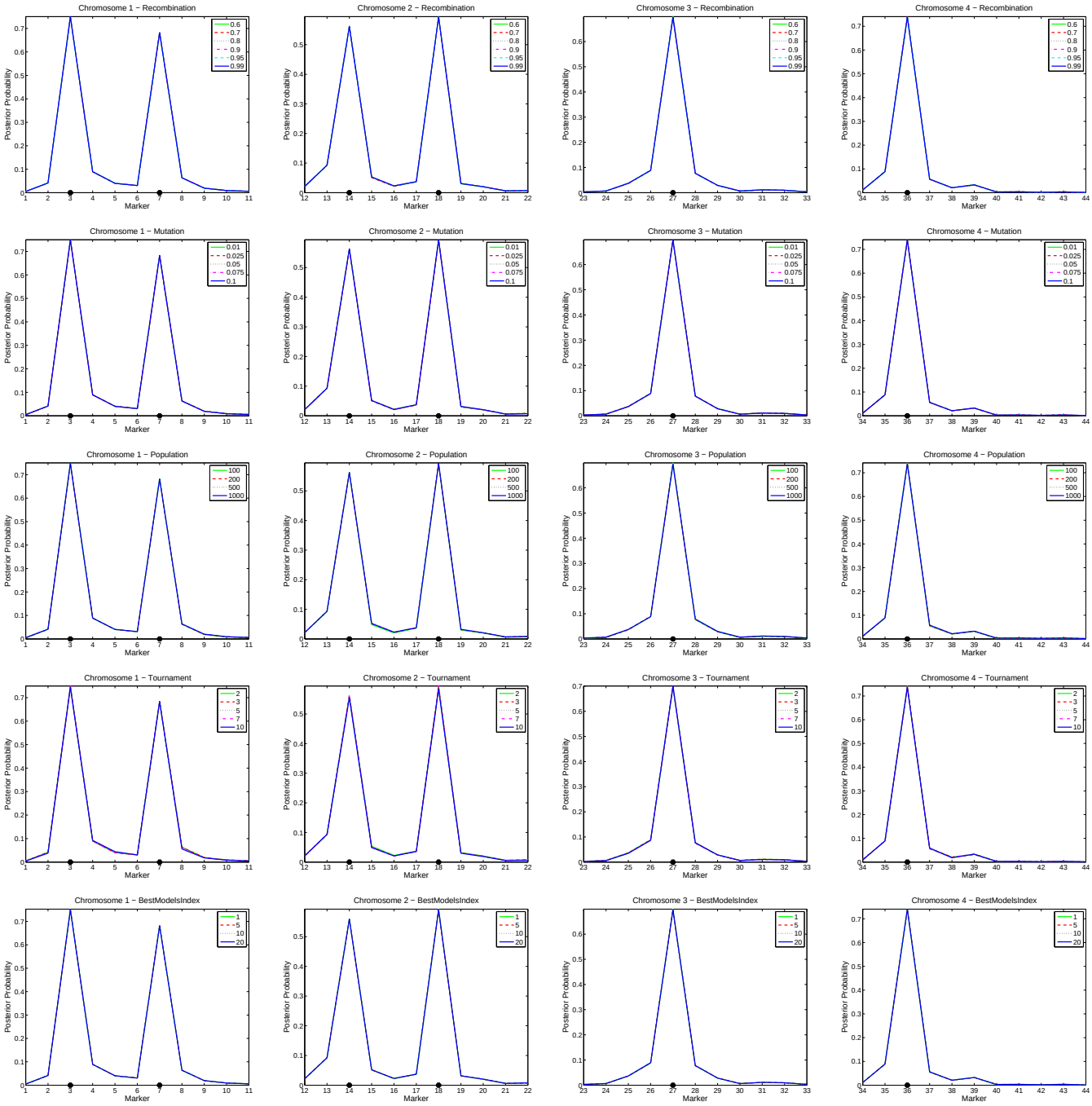

Figure 4.9: Mean posterior probability of each marker and each parameter setting based on all 100 data sets for scenario 1. QTL positions are indicated by a dot at the bottom.

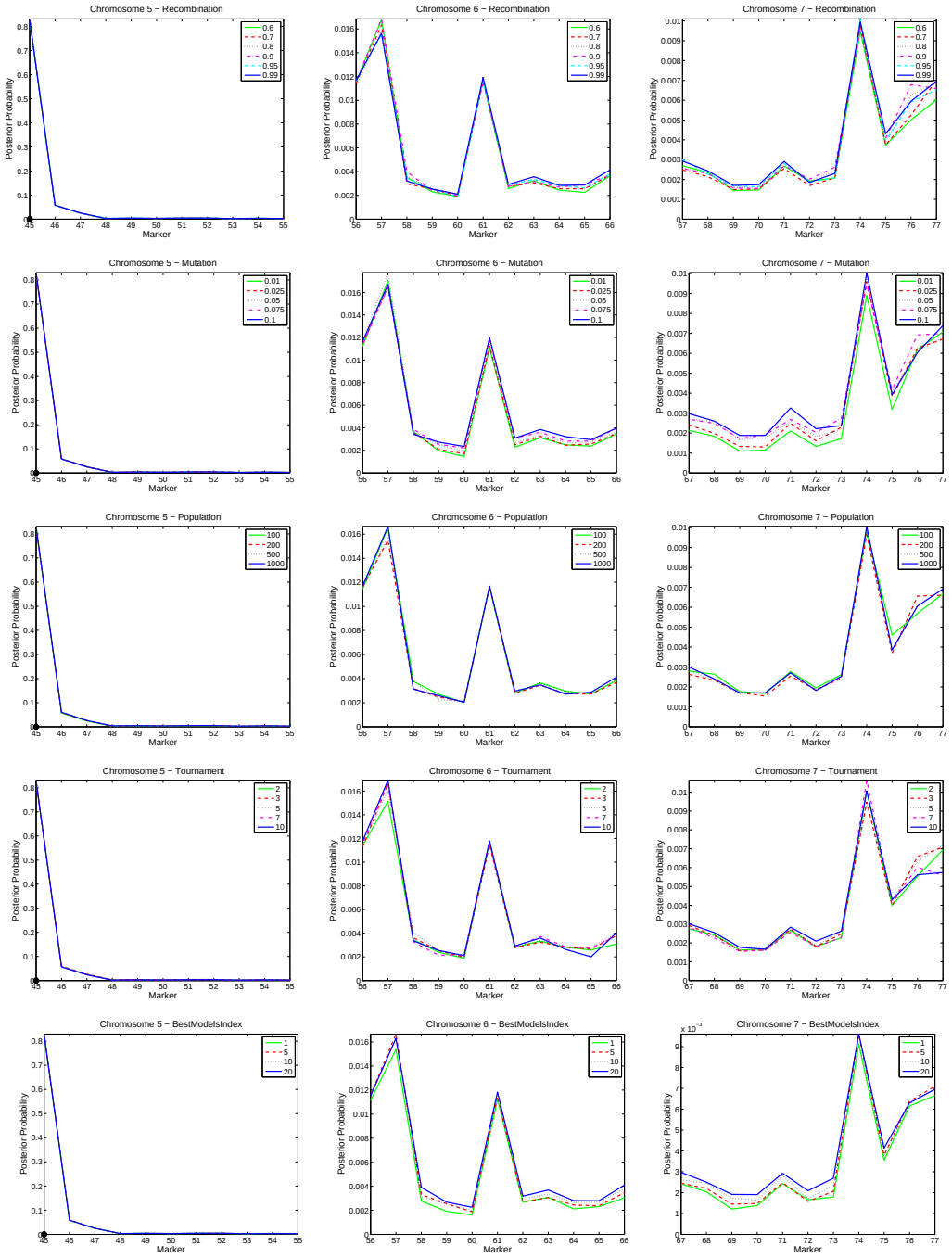

Figure 4.9: Mean posterior probability of each marker and each parameter setting based on all 100 data sets for scenario 1. QTL positions are indicated by a dot at the bottom.

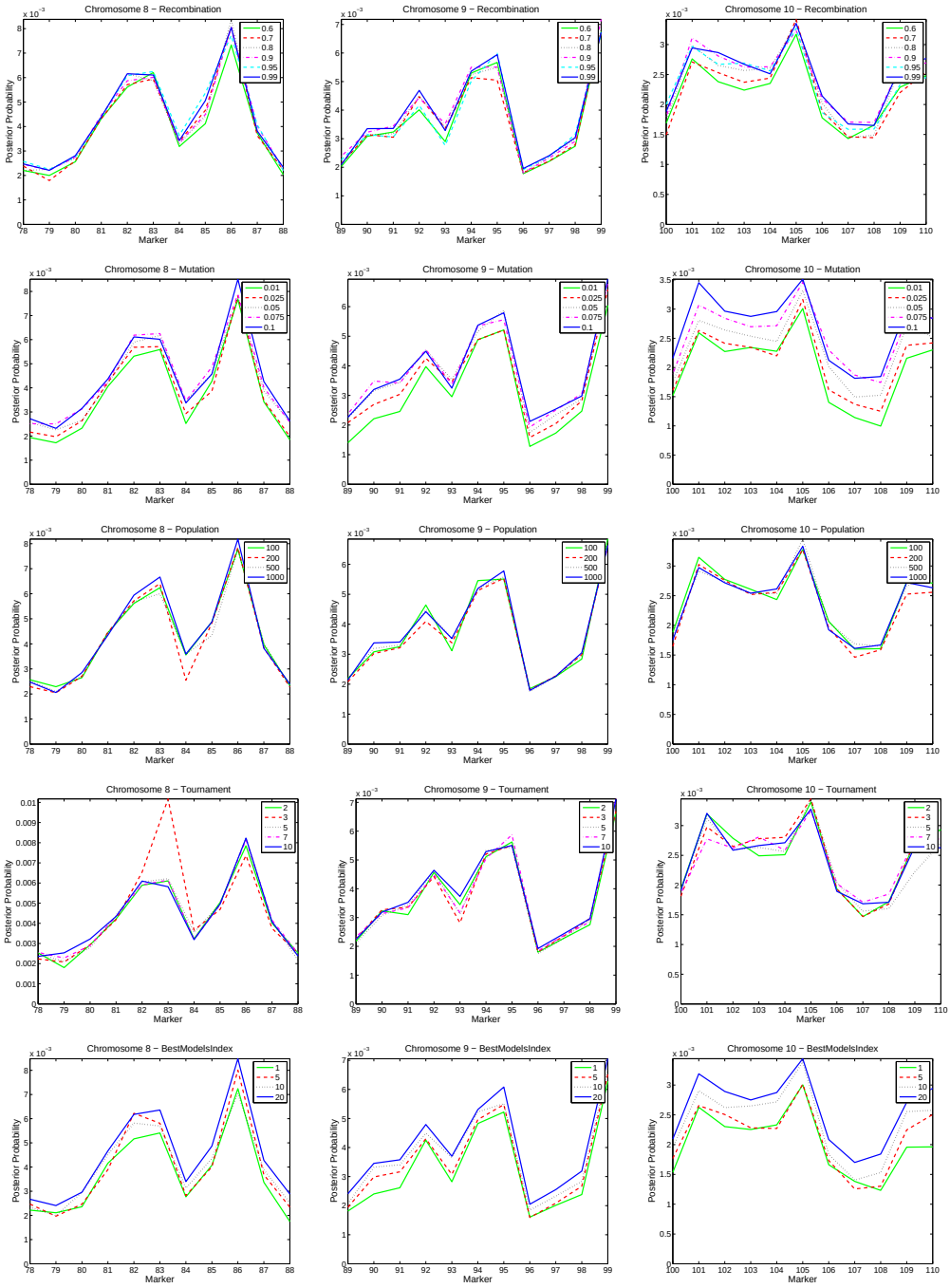

Figure 4.9: Mean posterior probability of each marker and each parameter setting based on all 100 data sets for scenario 1. QTL positions are indicated by a dot at the bottom.

In this thesis, we consider two different criteria to classify markers as QTLs: either the marker posterior is above 0.5 (the classical Bayesian approach) or the posterior is above 0.3. The search cutoff takes into account that the probability weight of the posterior might be distributed between neighbouring markers. On average all QTLs are easily identified in Figure 4.9 since the peaks of the posterior probability where QTLs are present are always quite high. On average no major difference can be observed in the posterior probability between different parameter settings. On chromosome 6-10 there are no QTLs present. As a result, the posterior probability at these chromosomes is very low and can be assumed to be due to noise.

To investigate the differences of the mean posterior probability for the parameter settings more closely, the difference of each parameter setting and the default parameter setting was determined (see Figure 4.10). When calculating the difference for each parameter setting group, the default setting within that particular group was used. For example, for the recombination parameter setting group, setting 3 was used to calculate the difference of the mean posterior probability.

In Figure 4.10, it is interesting to observe differences in the posterior probability at true QTL locations. However, the differences between parameter settings are very small, and of the same order like mean posterior probabilities at chromosomes which contain no QTLs (see chromosomes 6-10 in Figure 4.9). This suggests that the difference between the parameter settings does not reflect any hint of superior behaviour, but is rather due to random fluctuations.

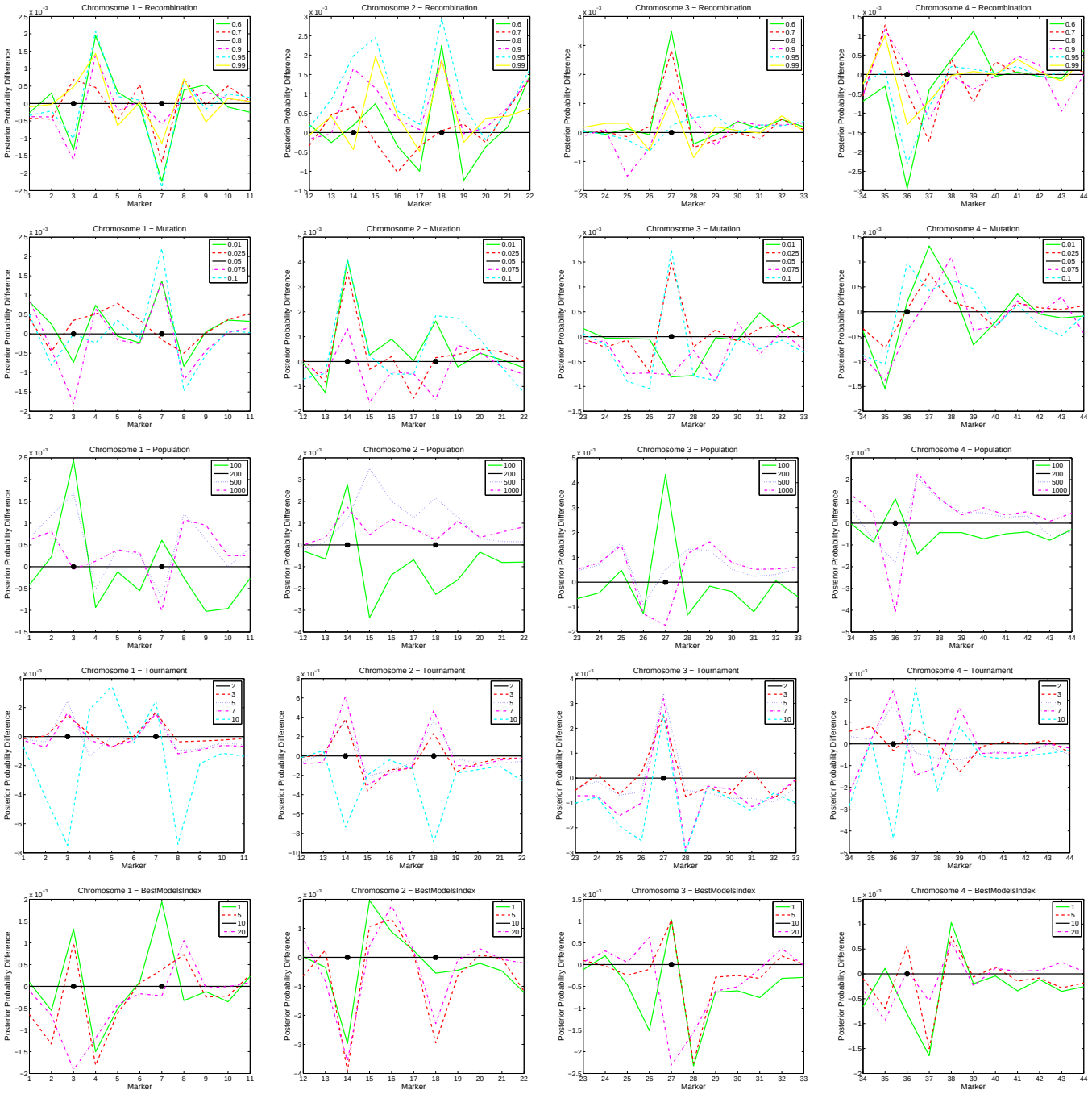

Figure 4.10: Difference of mean posterior probability for each marker and each parameter setting based on all 100 data sets for scenario 1. QTL positions are indicated by a dot at the bottom.

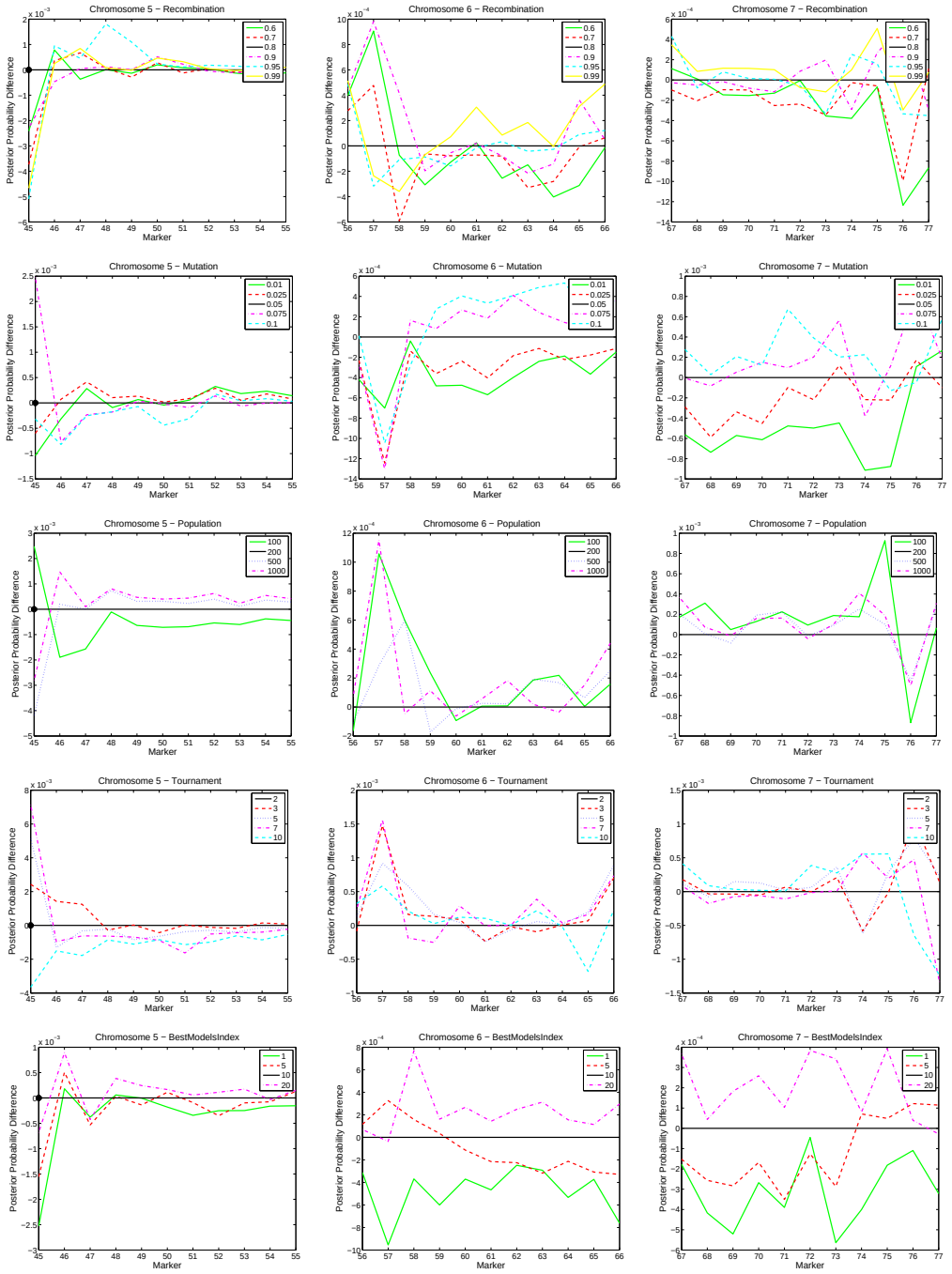

Figure 4.10: Difference of mean posterior probability for each marker and each parameter setting based on all 100 data sets for scenario 1. QTL positions are indicated by a dot at the bottom.

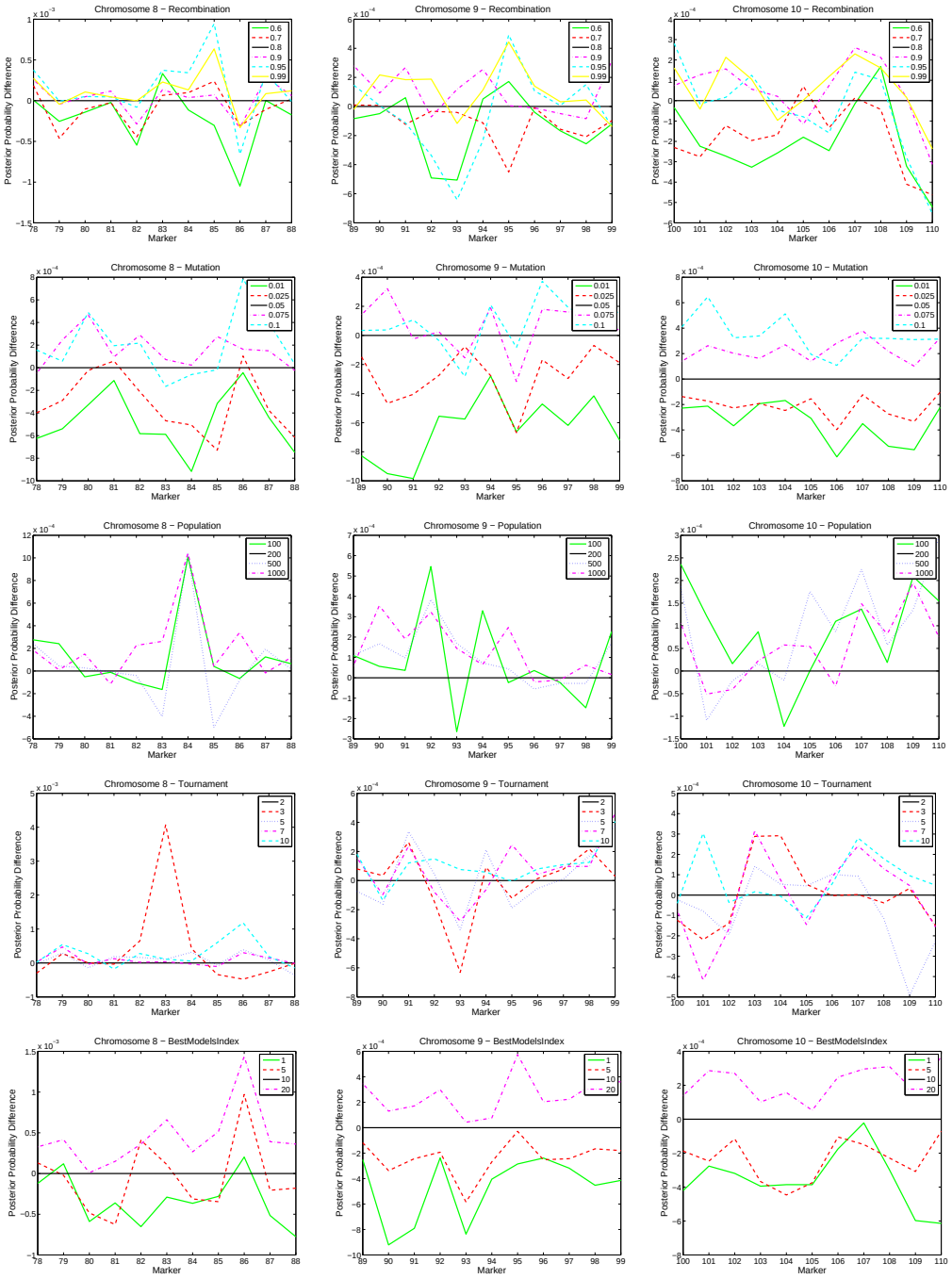

Figure 4.10: Difference of mean posterior probability for each marker and each parameter setting based on all 100 data sets for scenario 1. QTL positions are indicated by a dot at the bottom.

Since the differences of the posterior probability between the parameter settings within each group are so small it is interesting to investigate the difference between the five default settings which all have the same parameter settings but went through a GA simulation run independently. Figure 4.11 displays the posterior probability difference between all default parameter settings. When calculating the difference, the posterior probability of each setting was subtracted from the posterior probability of the first default setting, namely parameter setting 3.

Again, there are some peaks in the posterior probability difference where true QTLs are located, but in Figure 4.11 we know that any differences must be due to random fluctuations. The size of the difference is very similar to both the posterior probability differences between parameter settings in Figure 4.10 and the posterior probability of chromosomes that have no QTLs in Figure 4.9. This confirms that the difference of posterior probability between parameter settings is due to noise and one cannot deduce that any parameter setting would perform superior to others.

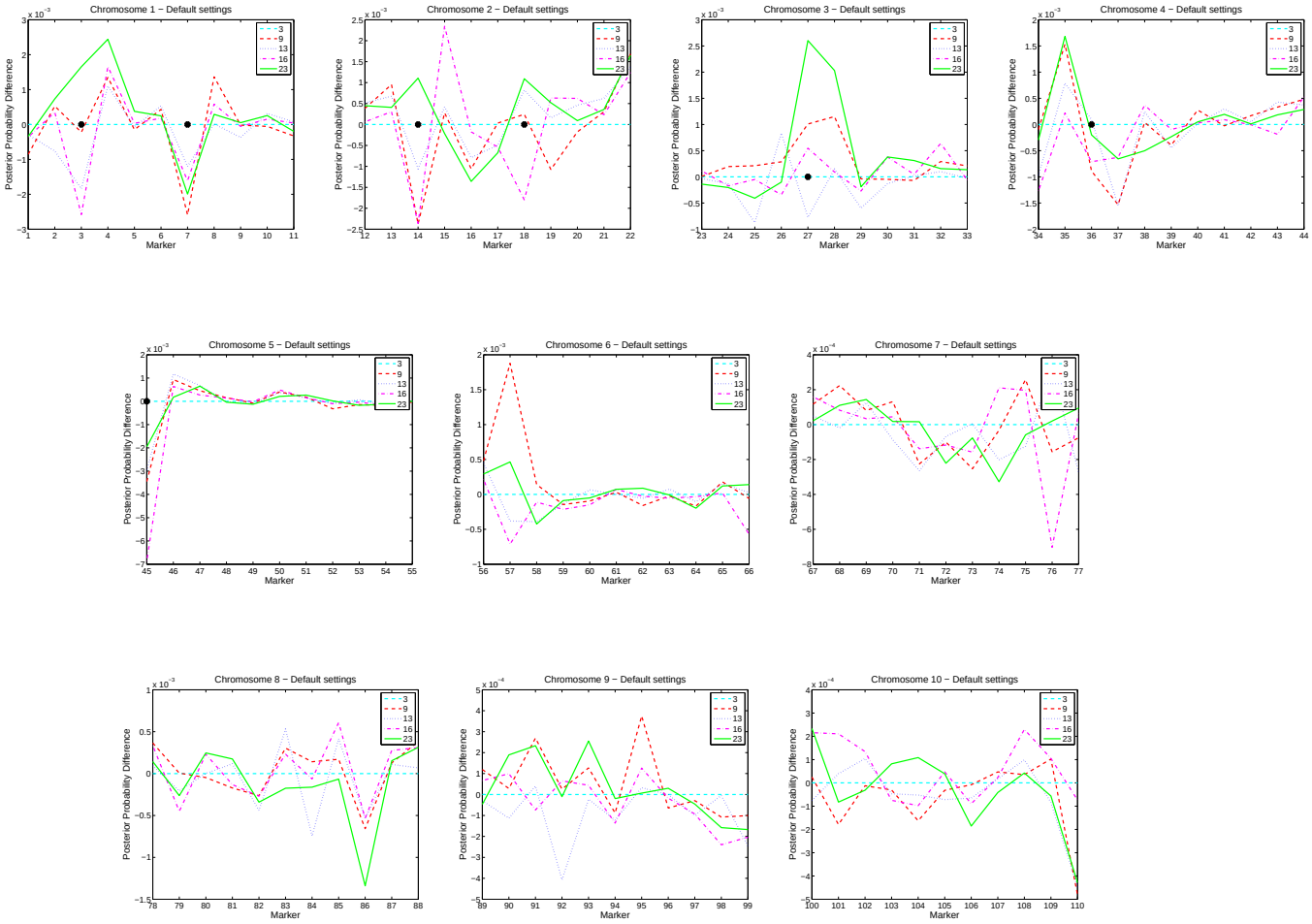

Figure 4.11: Difference of mean posterior probability for all default settings. QTL positions are indicated by a dot at the bottom.

When the mean of the marker posterior probability is calculated over 100 data sets, all QTLs are easily detected and the different parameter settings perform almost identically. However, if each data set is analysed separately, not all QTLs are easily identified. As an example, plots of the posterior probabilities of markers for data sets 7 and 88 are displayed in Figures 4.12 and 4.13.

Data set 7 (see Figure 4.12) is an example of a data set where all QTLs (that is, at markers 3, 7, 14, 18, 27, 36, 45) are easily detectable when observing the posterior probability plots. However, on chromosome 7, two peaks can be observed, a small one and one that has a posterior probability close to 0.45. These are false positives as there are no QTLs present on chromosome 7. Again, there is virtually no difference in the posterior estimates obtained with different parameter settings.

Data set 88 (see Figure 4.13) provides an example where several markers are difficult to detect. The QTL at marker 3 is not detectable, whereas the posterior probability at marker 4 is very high. The posterior probability of marker 7 is very low so that the QTL is not detected. Marker 14 has high posterior probability and thus a rather easily identifiable QTL. The QTL at marker 18 is not detected since its posterior probability is very low. The posterior probability at marker 27 is very low but the combined posterior probability of markers 25 and 26 is high enough to consider the QTL to be detected although its location is not entirely correct. This behaviour is to be expected due to the high correlation of markers which are close to each other. On chromosome 3 the posterior probability has a peak at marker 32, although there is no QTL at that location. Marker 36 has a rather low posterior probability. However, if the sum of markers 35 and 36 would be considered, that location could be identified as a possible QTL site. Although the peak of the posterior probability is correctly located for the QTL at marker 45, this posterior probability value is quite low for a possible QTL detection.

No big difference between various parameter settings can be observed for neither data set 7 nor data set 88. The correct identification of QTLs rather depends on the data set and on mBIC, and not on the specific choice of parameters. We now want to quantify more systematically the performance of GA to compute posterior probabilities correctly.

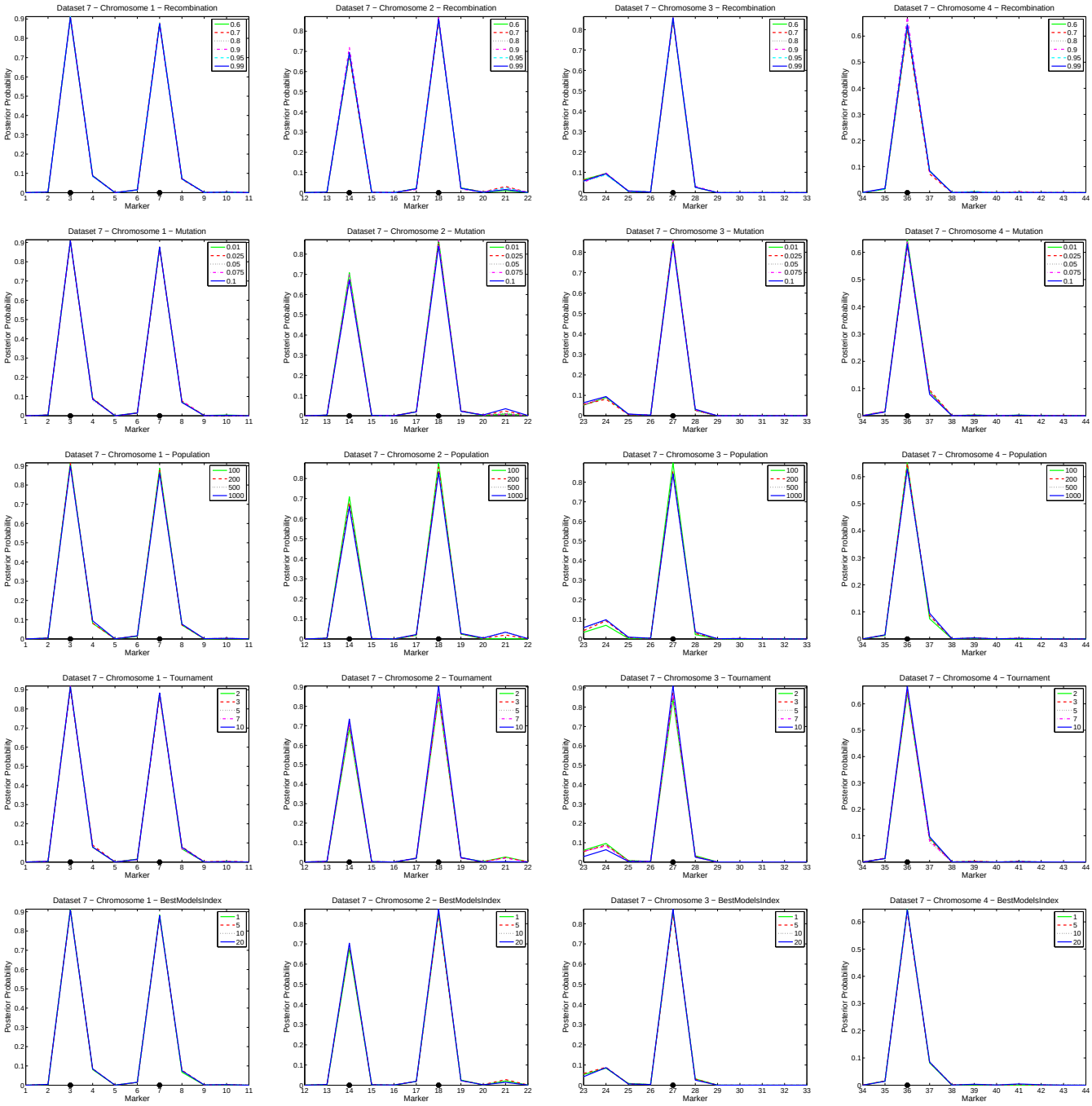

Figure 4.12: Posterior probability of each marker and each parameter setting for data set 7. QTL positions are indicated by a dot at the bottom.

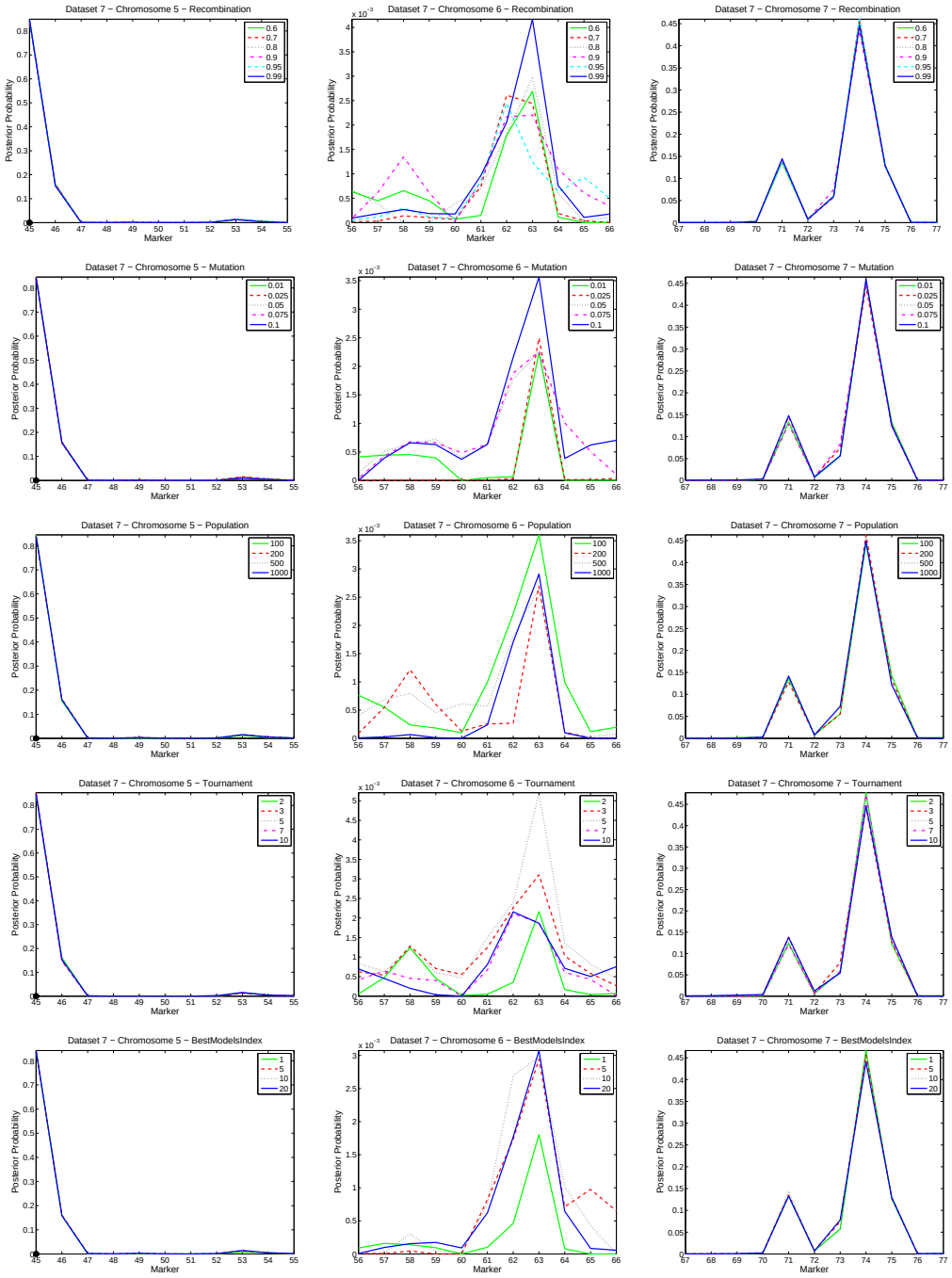

Figure 4.12: Posterior probability of each marker and each parameter setting for data set 7. QTL positions are indicated by a dot at the bottom.

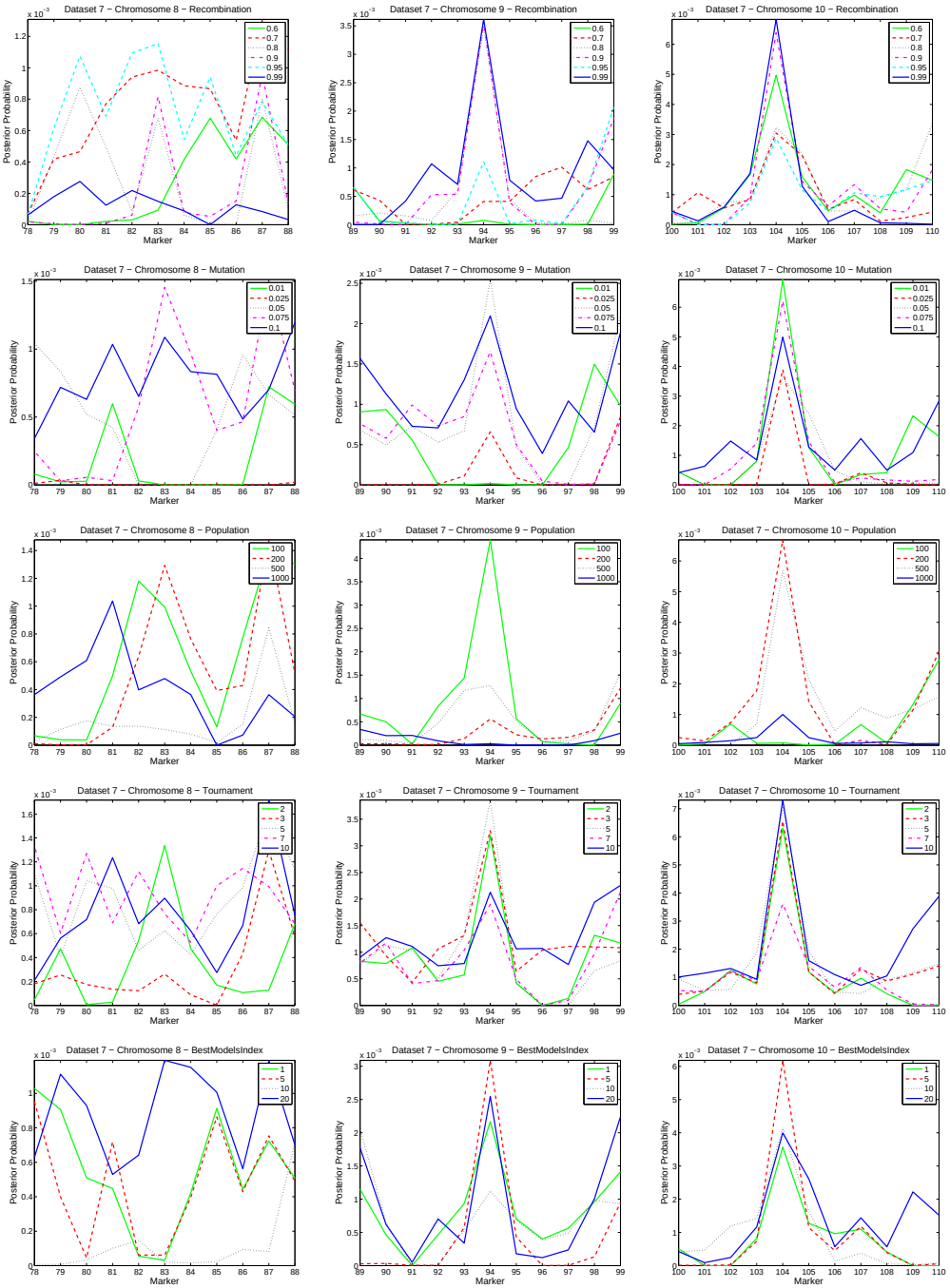

Figure 4.12: Posterior probability of each marker and each parameter setting for data set 7. QTL positions are indicated by a dot at the bottom.

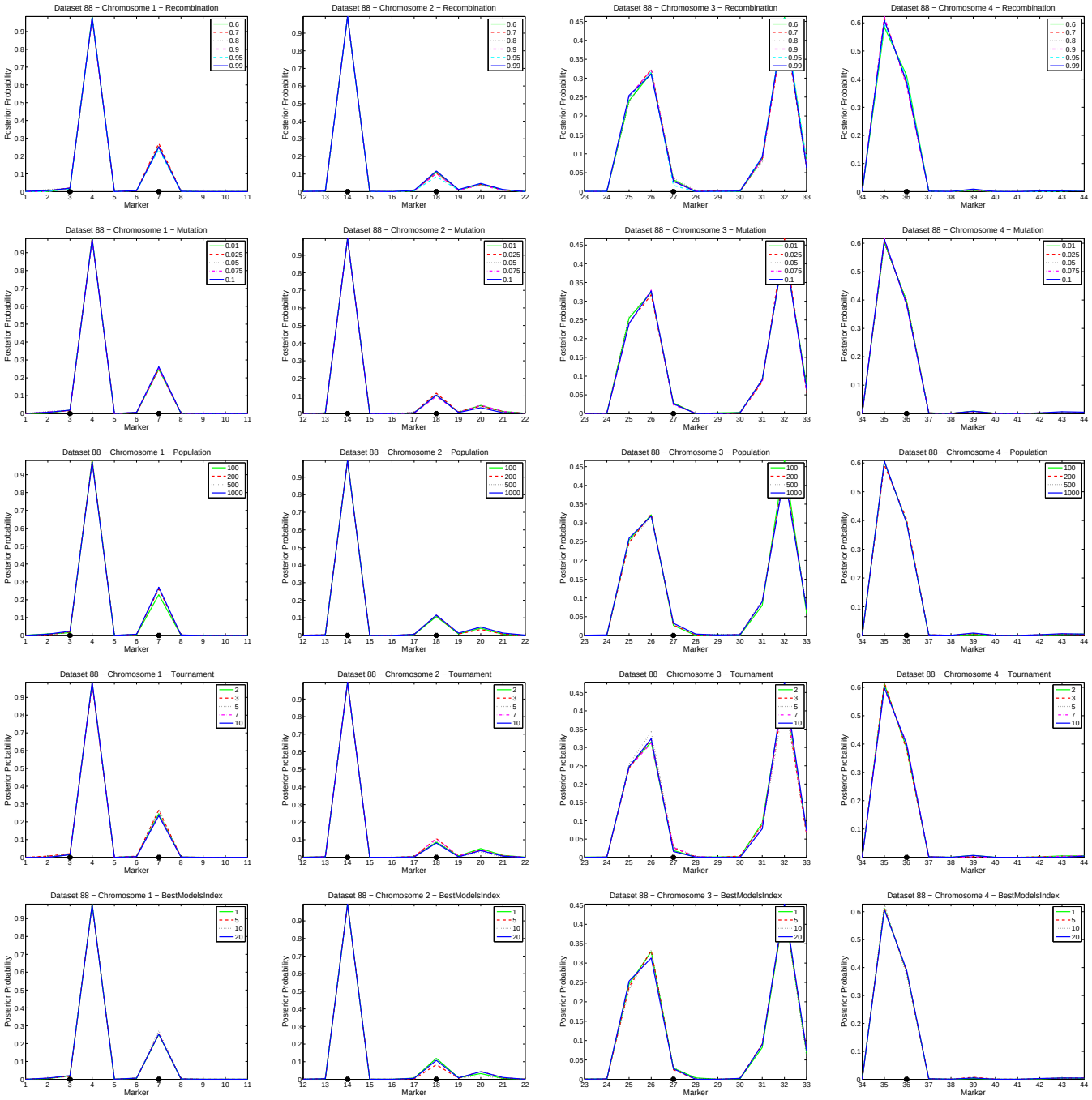

Figure 4.13: Posterior probability of each marker and each parameter setting for data set 88. QTL positions are indicated by a dot at the bottom.

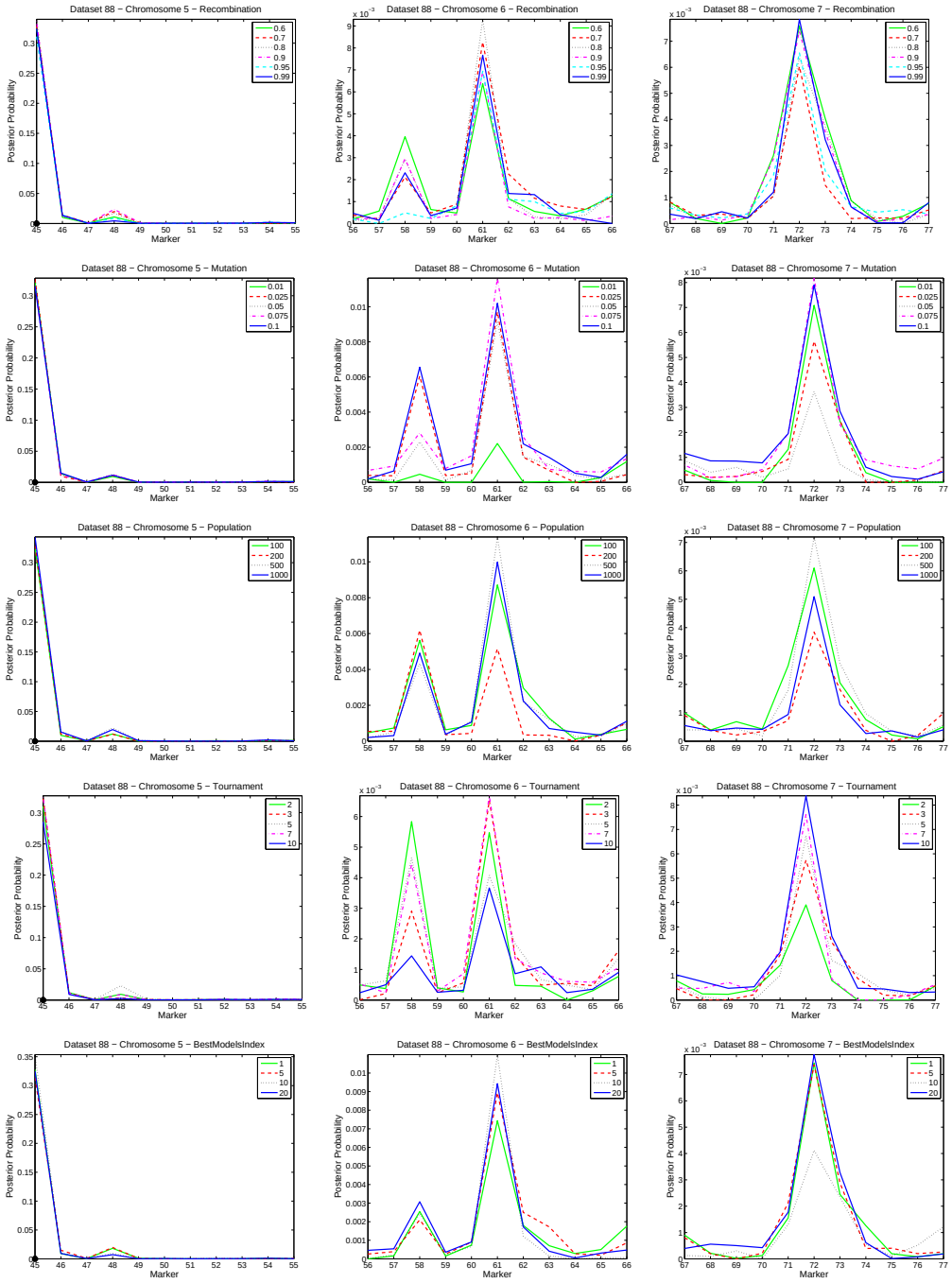

Figure 4.13: Posterior probability of each marker and each parameter setting for data set 88. QTL positions are indicated by a dot at the bottom.

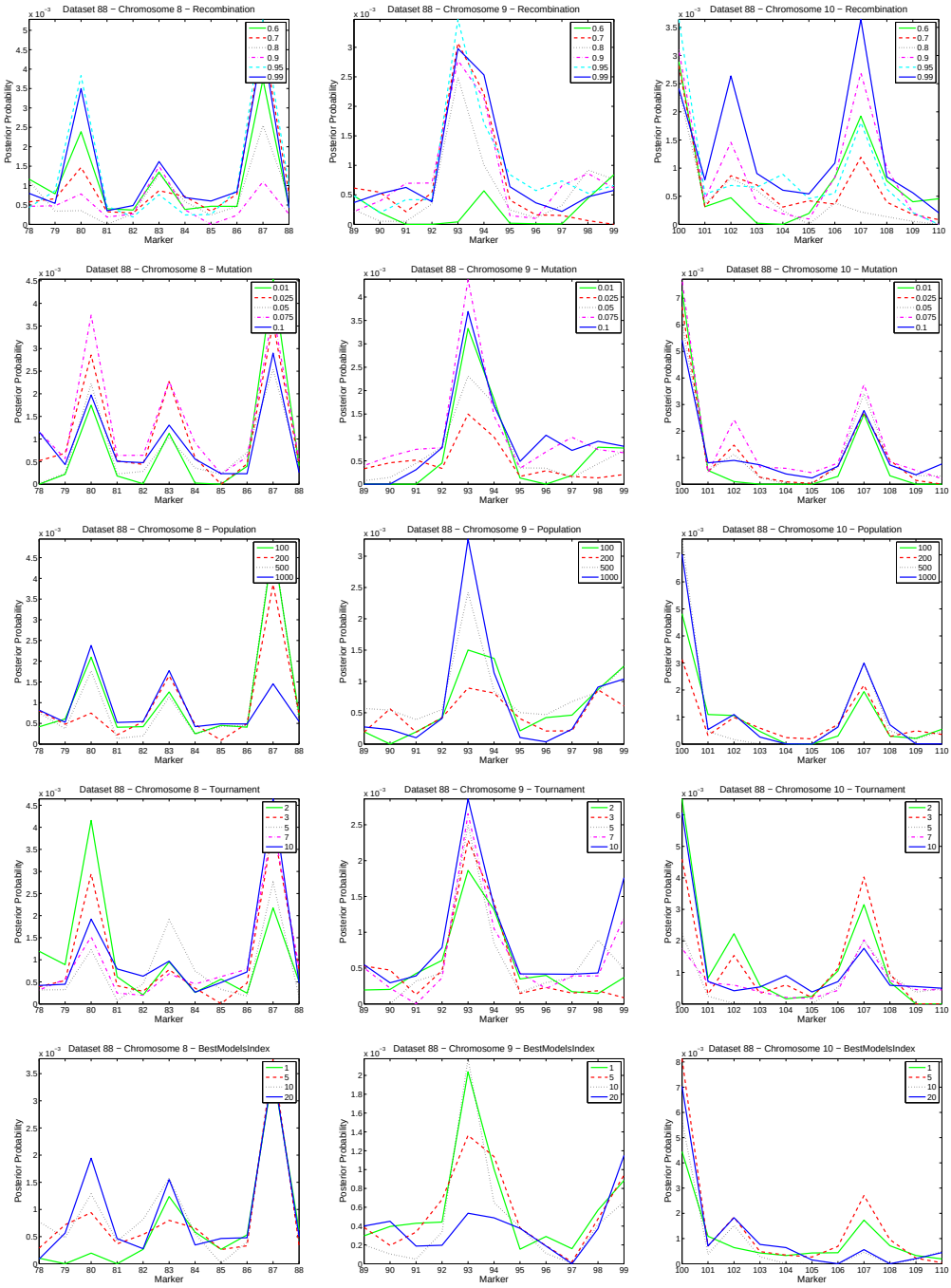

Figure 4.13: Posterior probability of each marker and each parameter setting for data set 88. QTL positions are indicated by a dot at the bottom.

Quality measures for posterior probabilities

A possible measure of the quality of a QTL mapping algorithm is the following L_2-norm

$$L_2 = \sqrt{\sum_{j \in M^*} (1 - \pi_j)^2 + \sum_{j \notin M^*} (\pi_j)^2} \tag{4.1}$$

where π_j is the posterior probability for marker j as defined in equation (3.4) and M^* is the correct model according to the true location of the QTLs. L_2 is the L_2 norm of the discrepancy between the posterior probability vector and the index vector of the true model. The mean L_2 and standard error of the mean were calculated over all 100 data sets for each parameter setting. The results can be seen in Figure 4.14.

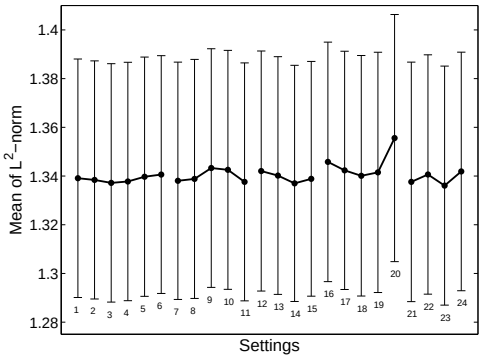

Figure 4.14: Mean L_2 and standard error of the mean for 24 different parameter settings

The difference of the mean L_2 between different parameter settings is quite small. Additionally, the mean value of each setting is within the standard error of the mean of all other settings. The small differences of the mean seem to be due to random fluctuation.
To confirm this, a repeated measures ANOVA F-test was performed for the null hypothesis

$$\mathrm{H}_{0P_q} : \text{the mean of } L_2 \text{ is the same for all parameter settings in } P_q$$

where P_q is the set of parameter settings in group $q = 1, ..., 5$. Remember that each parameter setting group includes the settings where one parameter varies while the others are kept fixed. For example $P_1 = \{1, ..., 6\}$ represents different recombination rate settings and $P_3 = \{12, ..., 16\}$ different population size settings.

Settings	1-6	7-11	12-15	16-20	21-24
p-value	0.8556	0.1715	0.2189	0.1120	0.2353

Table 4.2: p-values from repeated measures ANOVA F-test on the mean of L_2 for all parameter setting groups

The resulting p-values are displayed in Table 4.2. According to these p-values and a significance level of $\alpha = 0.01$, the null hypotheses cannot be rejected. This indicates that no one parameter setting performs better than any other in terms of the L_2.

In the posterior probability plots of Figures 4.12 and 4.13 it is interesting to observe that sometimes the posterior probability of the left and right flanking markers of a QTL is quite high. Since we do not want to miss those areas of possible QTLs, although their location is not fully accurate, we additionally report another quality measure, which we will call Q. We define it as

$$Q = \sqrt{\sum_{j \in M^*} (1 - \sum_{i \in U_j} \pi_i)^2 + \sum_{j \notin M^* \cup U_j} \pi_j^2} \tag{4.2}$$

where
π_i is the posterior probability for marker i,
M^* is the correct model according to true QTL locations,
U_j is the set of markers in the area around marker j defined as

$$U_j = \begin{cases} \{j-1, j, j+1\} & \text{if marker } j \text{ has left and right flanking markers,} \\ \{j-1, j\} & \text{if marker } j \text{ has only a left flanking marker, that is,} \\ & \text{it is located at the right end of a chromosome,} \\ \{j, j+1\} & \text{if marker } j \text{ has only a right flanking marker, that is,} \\ & \text{it is located at the left end of a chromosome.} \end{cases}$$

Therefore Q is again the L_2-norm of the discrepancy between the posterior probability vector and the index vector of the true model, but now the markers within each environment U_j are merged. The mean of Q and the standard error of the mean was calculated for each parameter setting and plotted in Figure 4.15.

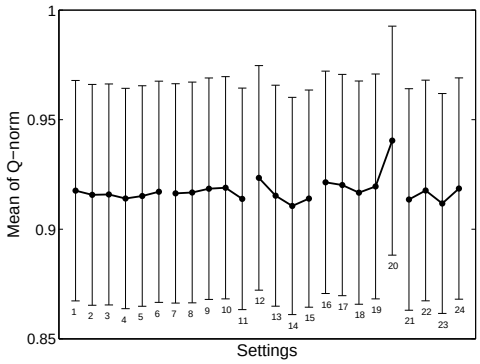

Figure 4.15: Mean Q and standard error of the mean for 24 different parameter settings

Mean and standard error of the Q are very similar between parameter settings. We again perform a repeated measures ANOVA F-test with the null hypotheses

$$\mathrm{H}_{0P_q} : \text{the mean of } Q \text{ is the same for all parameter settings in } P_q.$$

The resulting p-values are displayed in Table 4.3. According to these p-values and a significance level of $\alpha = 0.01$, only the null hypothesis for parameter setting group P_3 can be rejected. This indicates that the difference of the mean Q is significant between population size settings. In Figure 4.15, it seems that the mean Q is declining as population size is increased. For all other parameter setting groups no one parameter setting performs better than any other in terms of Q.

Settings	1-6	7-11	12-15	16-20	21-24
p-value	0.8867	0.4915	0.0030	0.1616	0.1044

Table 4.3: p-values from repeated measures ANOVA F-test on the mean of Q for all parameter setting groups

Power, FWER and FPN

To further investigate the performance of the GA with different parameter settings, the power, family wise error rate (FWER) and the total number of false positives and false negatives (FPN) were estimated.
A QTL is classified as detected if a marker which is within 15 cM of the QTL is selected. In classical Bayesian variable selection a marker is selected if its posterior probability is larger than 0.5. However, if a QTL is located between two markers, its posterior probability will be divided between the two flanking markers. Therefore, we also report power, FWER and FPN when a marker is selected with posterior probability larger than 0.3.

An estimate of power (number of true positives divided by the total number of QTLs) based on the 100 simulation runs is reported. The difference of the power between parameter settings can be seen in Figures 4.16 and 4.17.

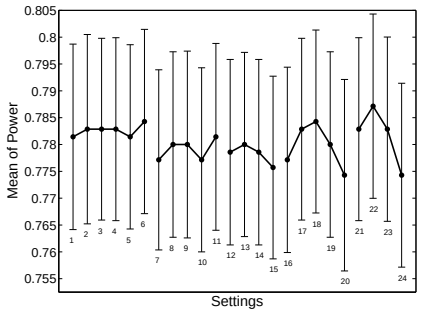

Figure 4.16: Mean power for selected marker posterior probability larger than 0.5 and standard error of the mean for 24 different parameter settings

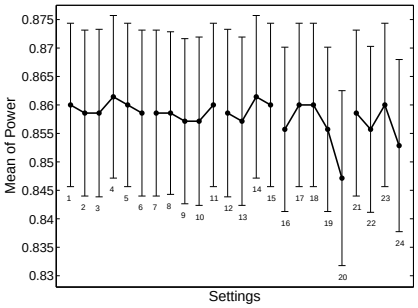

Figure 4.17: Mean power for selected marker posterior probability larger than 0.3 and standard error of the mean for 24 different parameter settings

The mean of the power does not show any obvious trend within parameter setting groups. Furthermore, the mean is quite similar among all parameter settings and each setting is within the standard error of the mean of the other settings.

An estimate of the family wise error rate (percentage of simulation runs where at least one false positive was detected) is reported for selected markers with posterior probability larger than 0.5 and 0.3 (see Figures 4.18 and 4.19).
The total number of false positives and false negatives is reported for selected markers that have posterior probability larger than 0.5 and 0.3 (see Figures 4.20 and 4.21).

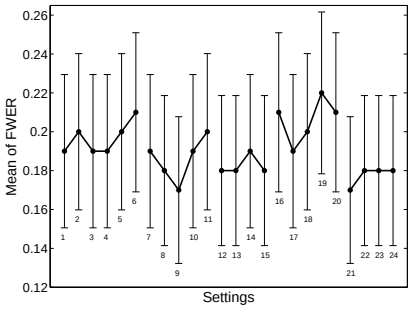

Figure 4.18: Mean family wise error rate (FWER) for selected marker posterior probability larger than 0.5 and standard error of the mean for 24 different parameter settings

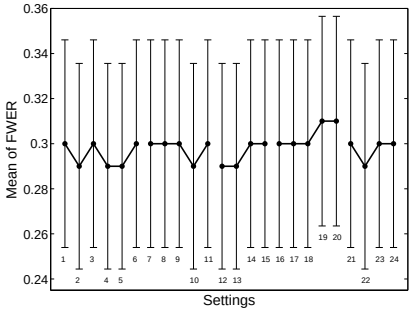

Figure 4.19: Mean family wise error rate (FWER) for selected marker posterior probability larger than 0.3 and standard error of the mean for 24 different parameter settings

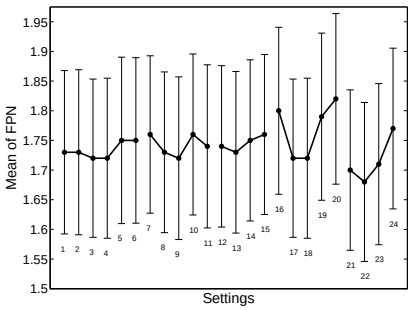

Figure 4.20: Mean total number of false positives and false negatives (FPN) for selected marker posterior probability larger than 0.5 and standard error of the mean for 24 different parameter settings

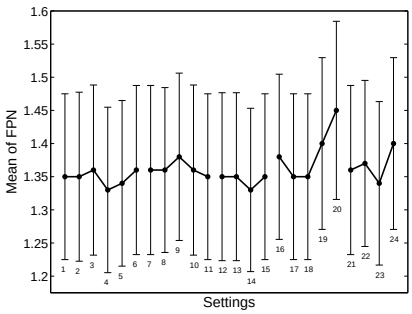

Figure 4.21: Mean total number of false positives and false negatives (FPN) for selected marker posterior probability larger than 0.3 and standard error of the mean for 24 different parameter settings

Just as for power, mean FWER and mean FPN do not show any obvious trend, and there are very little differences between the mean FWER and between the mean FPN for the various parameter settings (see Figures 4.18-4.21). We therefore conclude that no parameter setting performs better than any other when considering FWER and FPN.

Cumulative sum of $\exp(-\frac{\text{mBIC}}{2})$

The performance of each parameter setting was further investigated by calculating the cumulative sum of $\exp(-\frac{\text{mBIC}_i}{2})$ for each model i. That is, first the models were sorted in an ascending order according to their mBIC value, the cumulative sum of $\exp(-\frac{\text{mBIC}_i}{2})$ was determined and then the mean value over all data sets for each parameter setting was calculated (see Figures 4.22 - 4.26). This was done to compare the performance of different parameter settings in finding good models. The length of the lines in the plots is determined by the smallest pool among the 100 data sets for each parameter setting.

Figures 4.22 and 4.23 show no obvious trend in the cumulative sum, although the highest recombination rate and mutation rate seem to lead to the largest number of good models. When observing the plot for population size (see Figure 4.24) it can be seen that the cumulative sum increases with growing population size, with the exception that it is a bit higher for $u = 500$ than for $u = 1000$. The cumulative sum decreases as tournament size rises (see Figure 4.25), suggesting that $t = 2$ is performing best. As for the best model index, the cumulative sum grows as the index is increased (see Figure 4.26).

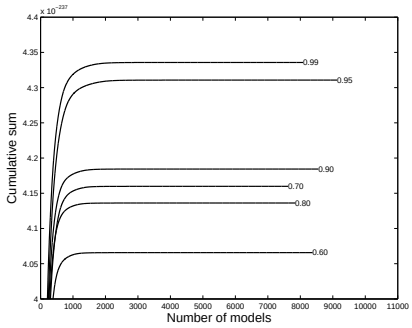

Figure 4.22: The cumulative sum of $\exp(-\frac{\text{mBIC}}{2})$ for different recomb. rates

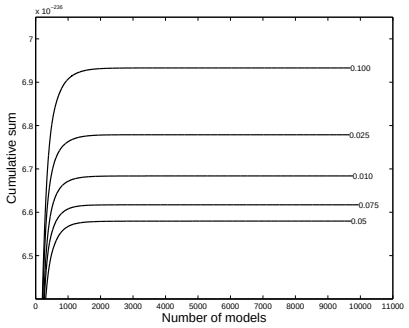

Figure 4.23: The cumulative sum of $\exp(-\frac{\text{mBIC}}{2})$ for different mutation rates

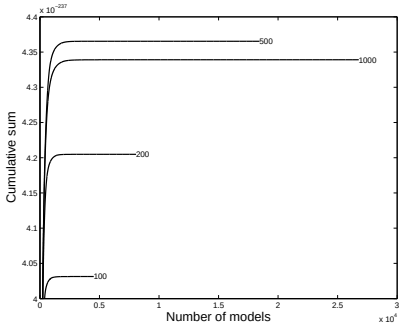

Figure 4.24: The cumulative sum of $\exp(-\frac{\text{mBIC}}{2})$ for different population sizes

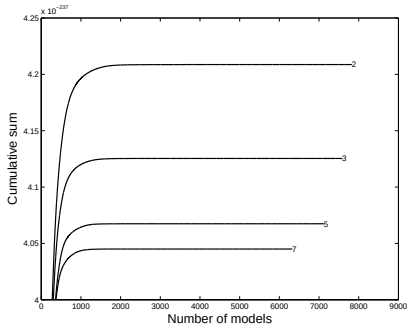

Figure 4.25: The cumulative sum of $\exp(-\frac{\text{mBIC}}{2})$ for different tourn. sizes

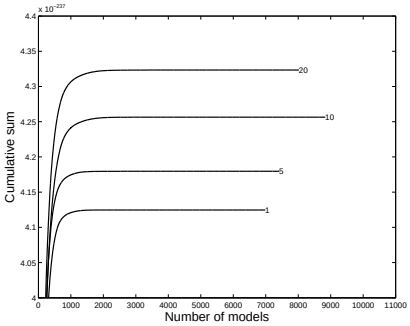

Figure 4.26: The cumulative sum of $\exp(-\frac{\text{mBIC}}{2})$ for different best model indices

Summary

The posterior probability plots displayed in Figure 4.9 show that on average all QTLs are easily detected and very few false positives occur. Therefore, using the GA to obtain these posterior probabilities seems to be a well performing procedure to localize QTLs.
According to the previously reported posterior probability plots, L_2, power, FWER and FPN, no one parameter setting performs better than any other. Only Q suggests that larger population sizes perform better than small ones.
Finally, the cumulative sum plots indicate that larger population sizes ($u = 500$ and $u = 1000$) and tournament size $t = 2$ perform best. For all other settings choosing parameters that minimize runtime is therefore sensible.

4.2.2 Scenario 2

For scenario 2 (see Figure 4.4) 100 data sets were again simulated. The performance of the GA was analysed with parameter settings slightly different from those used in scenario 1. The median or the values around the median of the previously studied parameter settings were chosen as default values for scenario 2, namely, recombination rate $p_r = 0.90$, population size $u = 200$, tournament size $t = 5$ and best model index $B = 10$. To increase variation and the total number of GA updates, the default value for mutation rate $p_m = 0.100$ was chosen. When examining a particular parameter, all other parameters were kept fixed at their default setting. Table 4.4 displays the 24 different settings that were used in the simulation. Note that settings 4, 11, 13, 18 and 23 all contain the identical default parameter values.

Setting	p_r	p_m	u	t	B
01	**0.60**	0.100	200	5	10
02	**0.70**	0.100	200	5	10
03	**0.80**	0.100	200	5	10
04	**0.90**	0.100	200	5	10
05	**0.95**	0.100	200	5	10
06	**0.99**	0.100	200	5	10
07	0.90	**0.010**	200	5	10
08	0.90	**0.025**	200	5	10
09	0.90	**0.050**	200	5	10
10	0.90	**0.075**	200	5	10
11	0.90	**0.100**	200	5	10
12	0.90	0.100	**100**	5	10
13	0.90	0.100	**200**	5	10
14	0.90	0.100	**500**	5	10
15	0.90	0.100	**1000**	5	10
16	0.90	0.100	200	**2**	10
17	0.90	0.100	200	**3**	10
18	0.90	0.100	200	**5**	10
19	0.90	0.100	200	**7**	10
20	0.90	0.100	200	**10**	10
21	0.90	0.100	200	5	**1**
22	0.90	0.100	200	5	**5**
23	0.90	0.100	200	5	**10**
24	0.90	0.100	200	5	**20**

Table 4.4: Parameter settings examined for scenario 2

Runtime, pool size, iterations and updates

Just as for scenario 1 we start with reporting the mean and standard deviation of the runtime (measured in seconds), pool size, total number of iterations and total number of GA updates for each parameter setting based on 100 different simulation runs. Figure 4.27 and 4.28 display the variation of runtime and pool size for different parameter settings. Runtime and pool size behave very similarly in scenario 1 and 2.

When the recombination rate is increased (see the first line or settings 1-6) the runtime and pool size grow. As in scenario 1, the reason is that the chance of an empty iteration is higher when the recombination rate is lower. The stopping criterion can thus be reached faster resulting in a lower runtime.

Results of changes in the mutation rate are displayed by the second line (settings 7-

11). Like for scenario 1 the runtime goes down when the mutation rate is increased but the pool size goes up. In case of mutation, the mBIC is usually increased, and the new string might not be good enough to be among the top B models of the population. The stopping criterion is therefore reached quicker, while the pool size increases due to many new models visited in the local improvement step.
The third line (settings 12-15) shows variations for different population size settings. Increasing population size means that the chance of selecting the same parents repeatedly is lower and therefore finding new strings to enter the top B number of models is more likely. Therefore, both runtime and pool size increase as population size grows.
As can be seen in the fourth line (settings 16-20), an increase in tournament size results in lower runtime and smaller pool size. The reason is that when more strings are chosen for the tournament group, the chances that the same parents are chosen repeatedly from that group is higher leading to faster GA termination. Again, as to be expected, runtime and pool size increase when the best model index is increased (see the last line or settings 21-24).

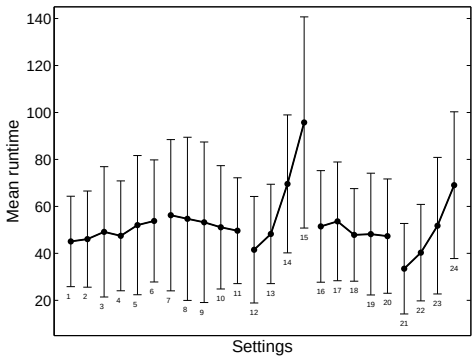

Figure 4.27: Mean runtime and standard deviations for 24 different parameter settings

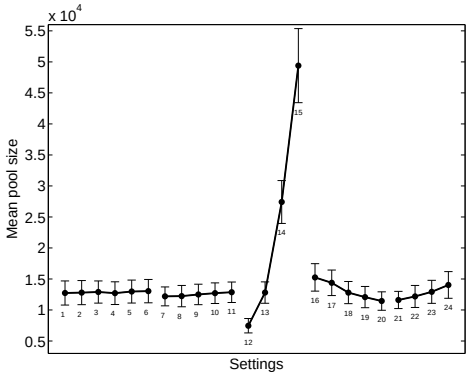

Figure 4.28: Mean pool size and standard deviations for 24 different parameter settings

Figures 4.29 and 4.30 display the mean and standard deviation of the total number of iterations and the total number of GA updates (see "Iteration count" and "Update count" in Figure 3.4). Again, the observed behaviour is very similar to scenario 1.
The first line (settings 1-6) shows the reactions to changes of the recombination rate. Since the chance of an empty iteration is lower when the recombination rate is higher both iterations and updates decrease when the recombination rate is increased.
As is shown by the second line (settings 7-11), the number of iterations slightly decreases while the number of updates increases when the mutation rate grows.

The reason is that the GA stopping criterion can be reached faster as the mBIC of a string deteriorates during mutation. However, since mutation usually yields a string that is new to the population the number of updates increases.
When population size (settings 12-15) is increased it is not clear if the number of iterations increases or decreases but the number of updates displays notable increase. When the population size is larger the probability of new strings entering the population is higher. Therefore, the number of updates accumulates.
The fourth line, which reflects change in tournament size (settings 16-20), seems to ascend for the number of iterations but descend for the number of updates. When tournament group is larger the probability that the same parents are chosen repeatedly is higher leading to an increase in the number of iterations but a decrease in the number of updates.
Increase of best model index (settings 21-24) means again rise in both the number of iterations and the number of updates.

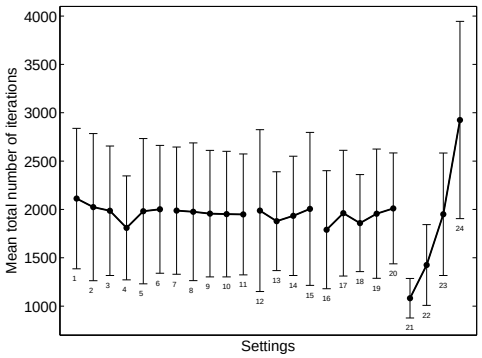

Figure 4.29: Mean total number of iterations and standard deviations for 24 different parameter settings

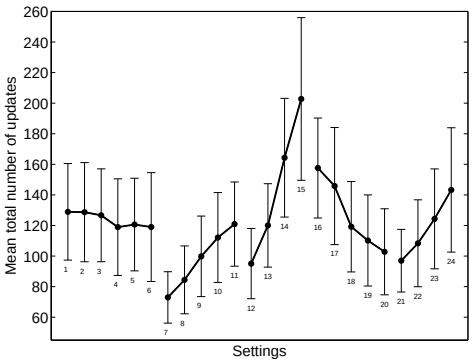

Figure 4.30: Mean total number of updates and standard deviations for 24 different parameter settings

Posterior probability plots

Figure 4.31 displays the mean posterior probability for each marker and every parameter setting computed according to equation (3.4).

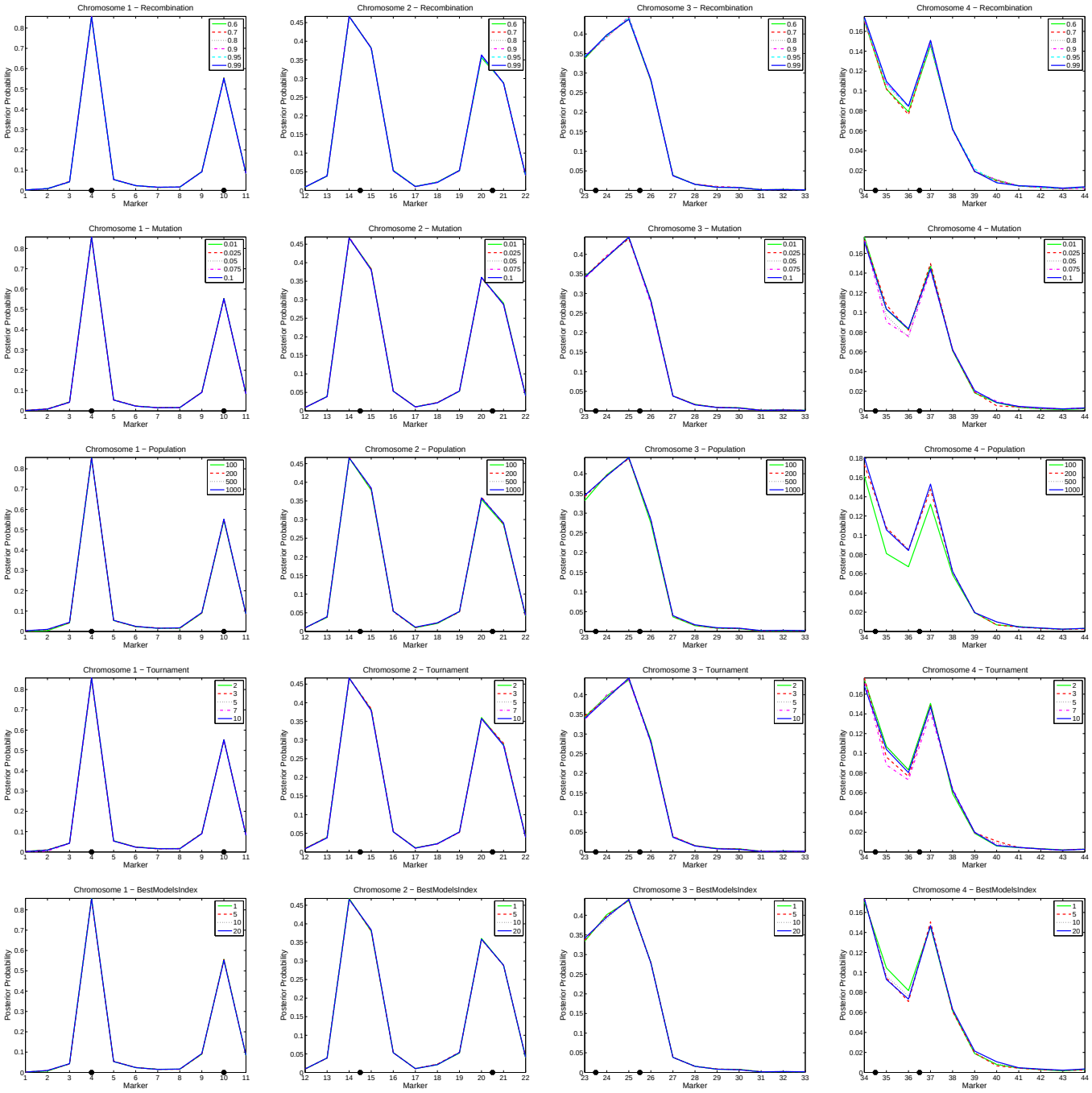

Figure 4.31: Mean posterior probability of each marker and each parameter setting based on all 100 data sets for scenario 2. QTL positions are indicated by a dot at the bottom.

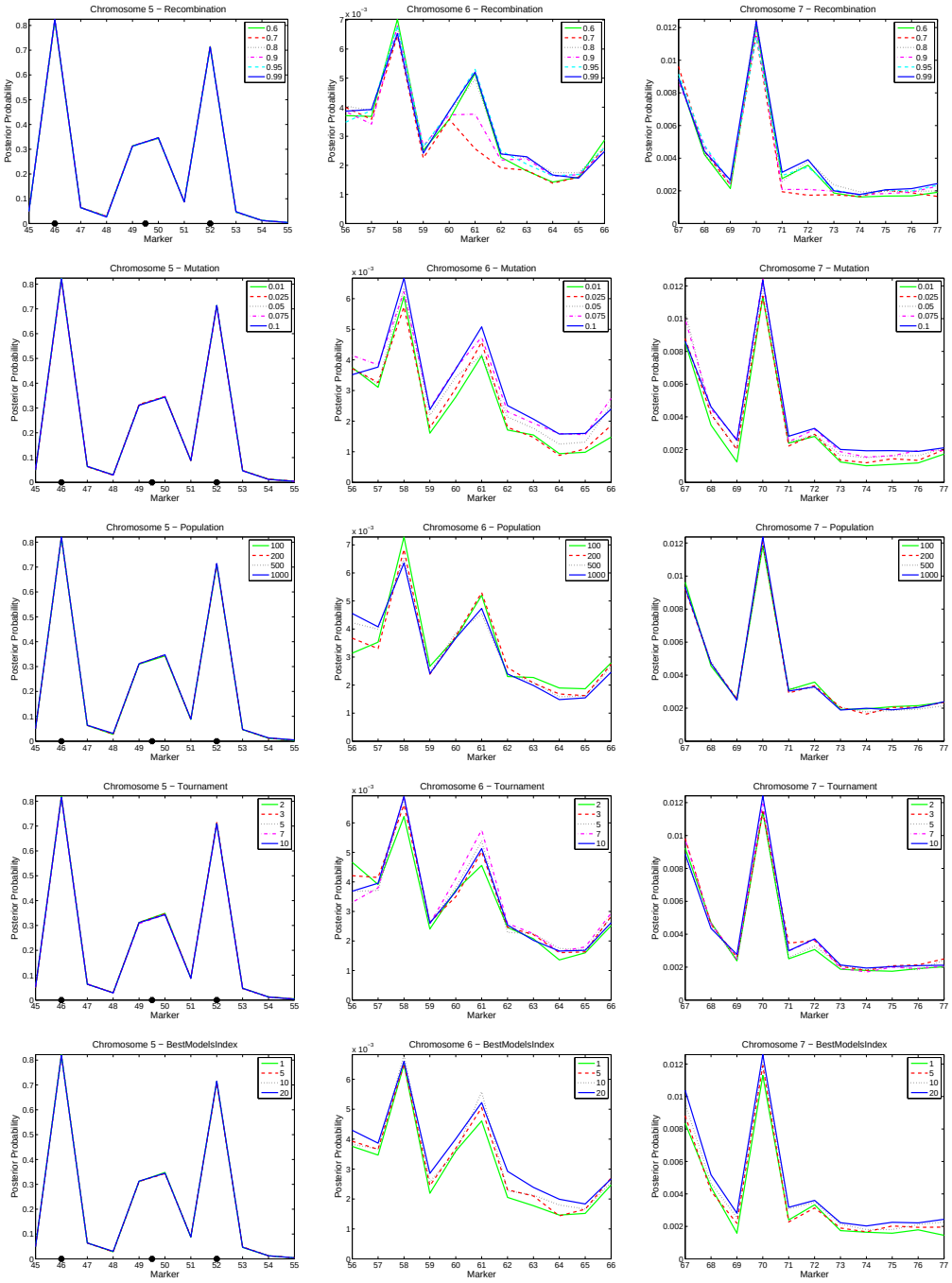

Figure 4.31: Mean posterior probability of each marker and each parameter setting based on all 100 data sets for scenario 2. QTL positions are indicated by a dot at the bottom.

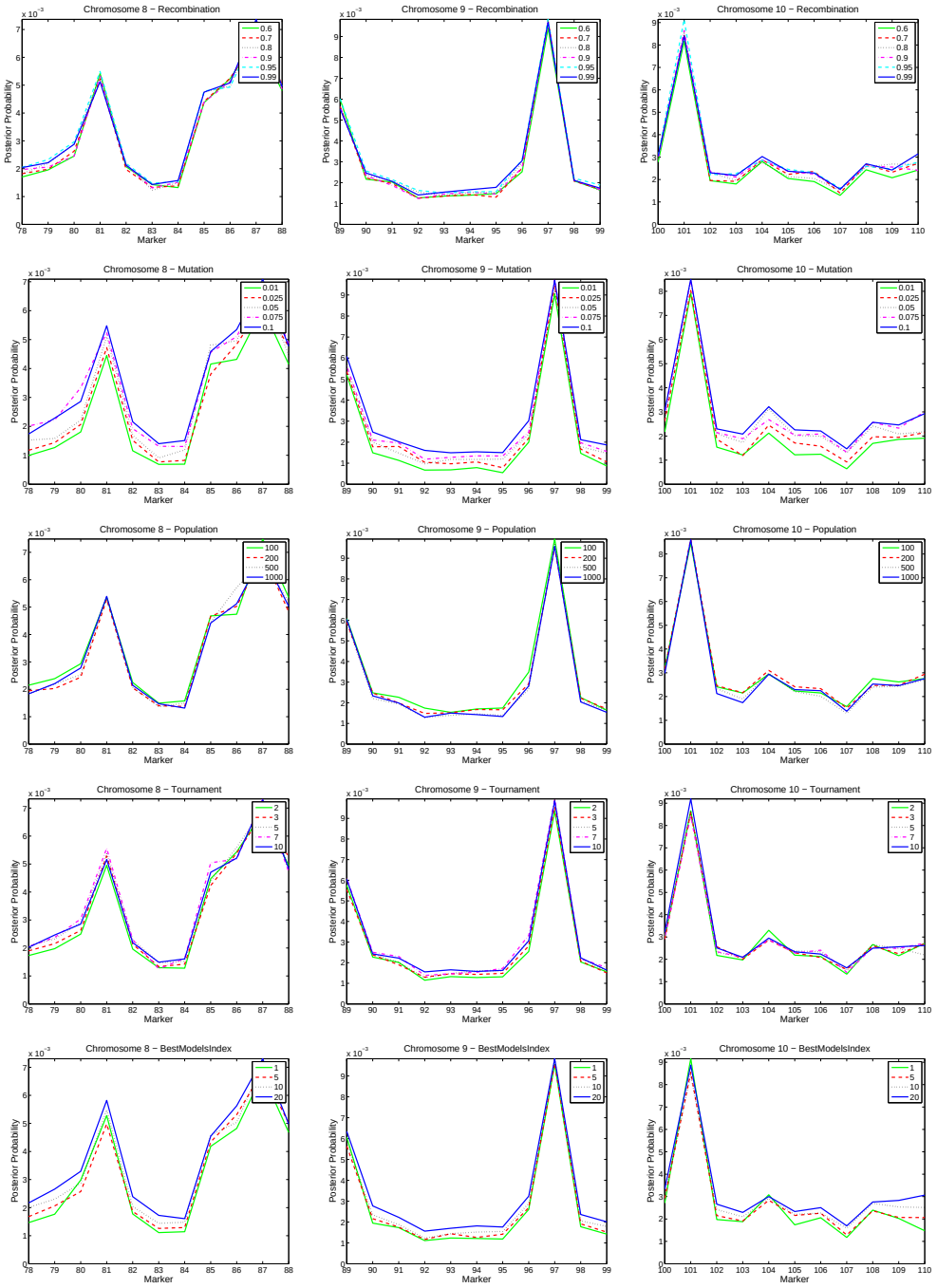

Figure 4.31: Mean posterior probability of each marker and each parameter setting based on all 100 data sets for scenario 2. QTL positions are indicated by a dot at the bottom.

On chromosome 1 QTLs are located directly at markers 4 and 10. Those QTLs have effect sizes $\beta = 1$ and $\beta = 0.75$ respectively. The QTL at marker 4 is easily detectable but the QTL at marker 10 has considerably lower posterior probability due to lower effect size.
Chromosome 2 also has two QTLs which are placed between markers 14 and 15 and markers 20 and 21 with effect sizes $\beta = 1$ and $\beta = 0.75$. The posterior probability peaks are not very high, in fact, it is lower than 0.5. However, the posterior probability of the QTLs are divided between the two flanking markers. If the sum of the posterior probability of these two markers is considered, the correct location would be detected as a possible QTL site.
On chromosome 3 there are QTLs located between markers 23 and 24 and between markers 25 and 26 both with effect size $\beta = 1$. Here, the QTLs are so close together that it is not possible to distinguish them. However, the posterior probability sum is high enough for the locations at markers 23-24, 24-25 and 25-26 to be detected as QTL sites.
The QTLs on chromosome 4 are as close to each other as the ones on chromosome 3 but have different effect sizes, namely $\beta = 1$ and $\beta = -1$ respectively. Here, two peaks are noticable in the posterior probability, although their posterior probability is quite low. The posterior probability sum of markers 34 and 35 and of markers 36 and 37 would not be high enough to be detected even if the posterior probability minimum for detecting a QTL were 0.3.
Chromosome 5 has three QTLs at marker 46, between markers 49 and 50 and at marker 52, all with effect sizes $\beta = 1$. Both QTLs placed directly on a marker are easily detectable. The QTL between 49 and 50 has its posterior probability divided on the flanking markers, making the sum of their posterior probability high enough to be considered as a QTL site.
Chromosomes 6-10 have no QTLs located on them and they all have very low posterior probabilities which are due to noise. Differences between different parameter settings can only be observed on chromosome 4. The posterior probability of all parameter settings is pretty similar, except for population size $u = 100$, which performs notably worse than other population size settings. However, the posterior probabilities on chromosome 4 are in general quite low, which makes the difference between parameter settings more visible than for other chromosomes.

In order to examine the difference between the parameter settings better, the difference of each parameter setting and the default setting within each parameter setting group was determined. Figure 4.32 displays the differences of the mean posterior probabilities. The posterior probability differences appear to be pretty random although there are some peaks around the true QTL locations. No clear trend in any parameter setting group can be detected, except for the population size setting group. There, population size $u = 100$ generally has lower posterior probability than other population size settings, in particular at QTL locations. This suggests that population size $u = 100$ might be too low. The magnitude of the posterior probability differences is similar to those of chromosomes 6-10 where no QTLs are present. This suggests that the differences between the parameter settings are due to random fluctuations and not performance dissimilarities.

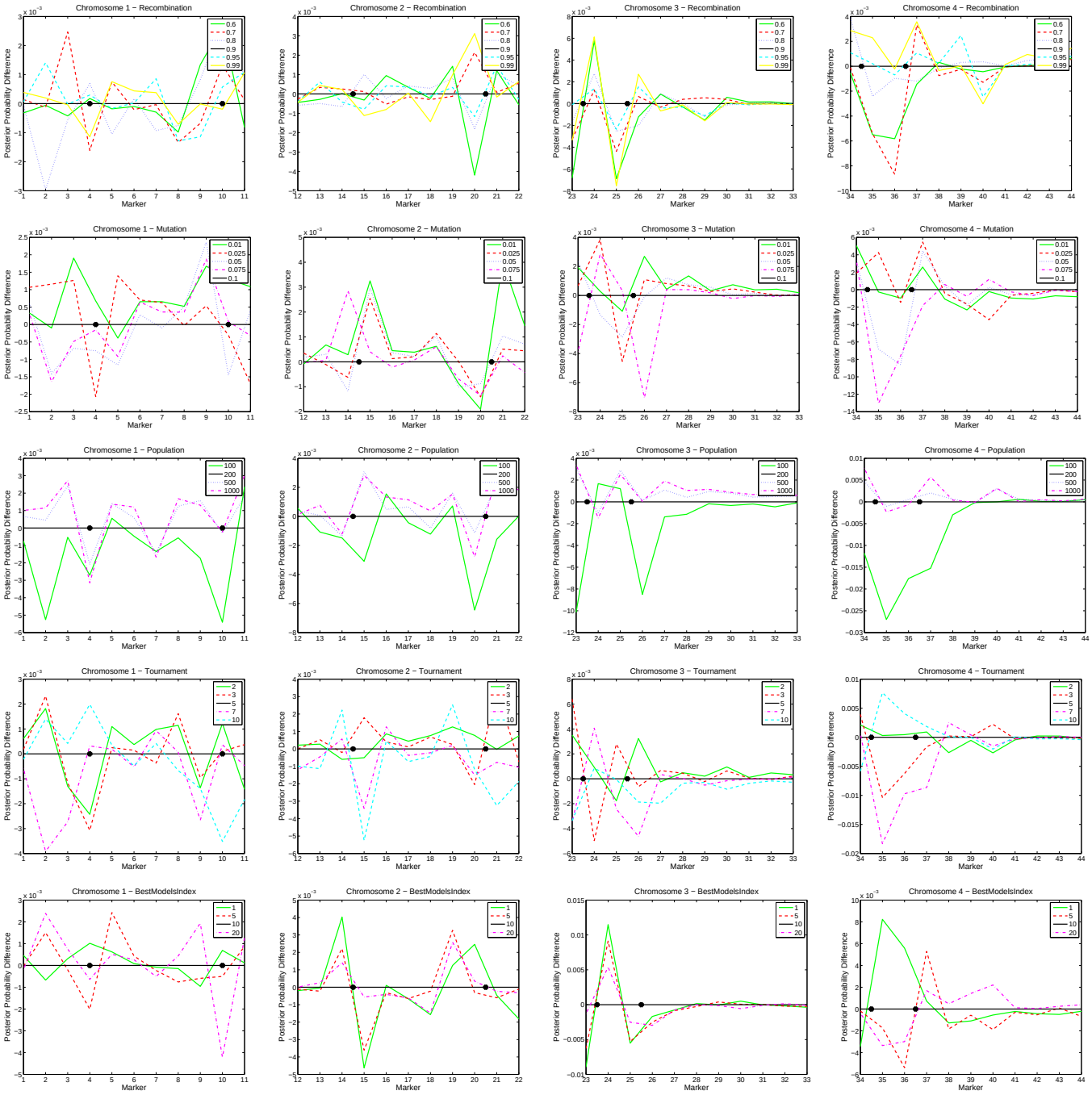

Figure 4.32: Difference of mean posterior probability for each marker and each parameter setting based on all 100 data sets for scenario 1. QTL positions are indicated by a dot at the bottom.

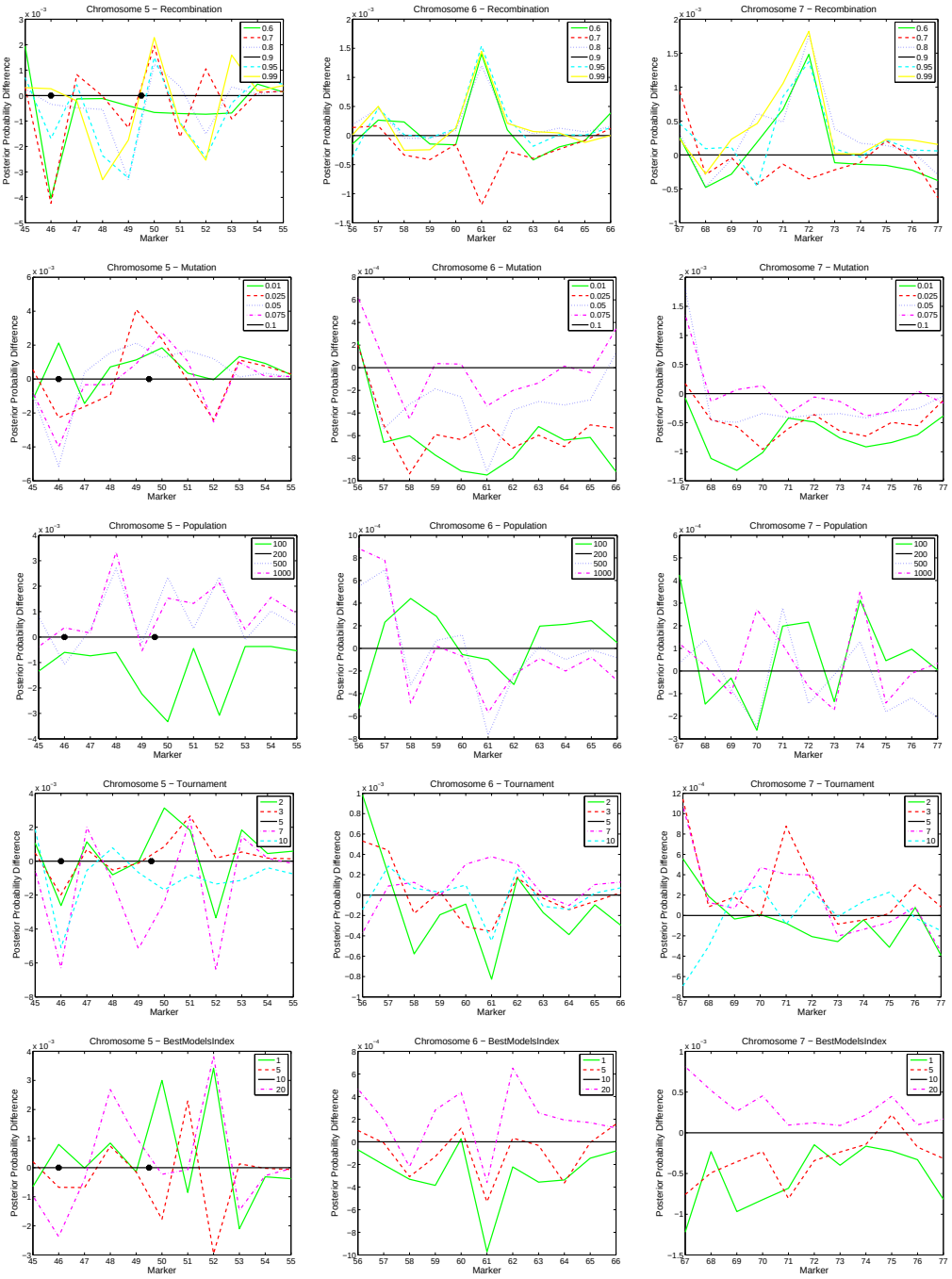

Figure 4.32: Difference of mean posterior probability for each marker and each parameter setting based on all 100 data sets for scenario 1. QTL positions are indicated by a dot at the bottom.

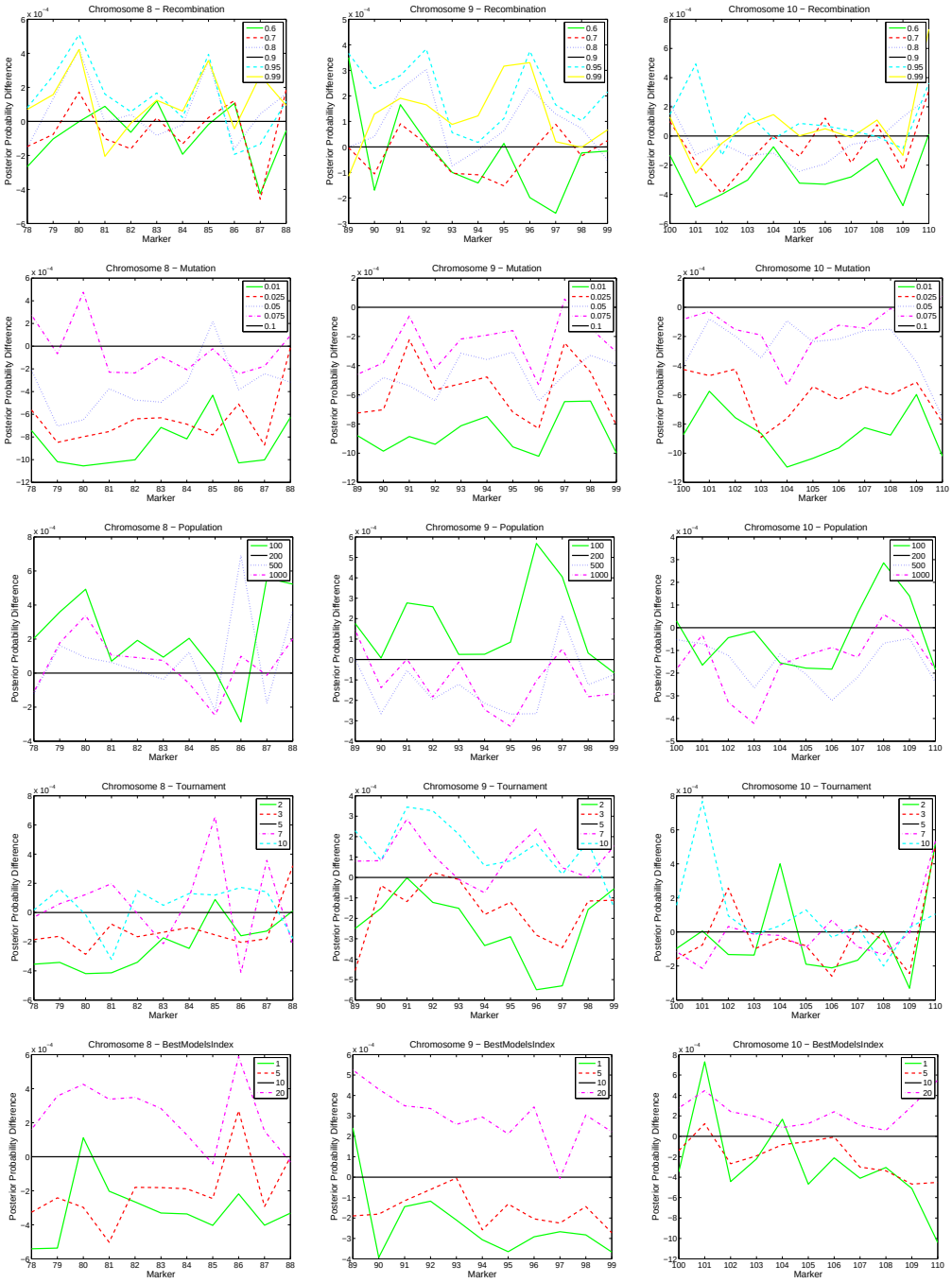

Figure 4.32: Difference of mean posterior probability for each marker and each parameter setting based on all 100 data sets for scenario 1. QTL positions are indicated by a dot at the bottom.

To examine the random fluctuations in more detail, the difference of the mean posterior probability between the default settings was determined (see Figure 4.33). The differences were calculated by subtracting the mean posterior probability value of the first default parameter setting from each of the other default parameter settings.
These differences are similar to the differences between parameter settings and, likewise, between parameter settings on chromosomes without QTLs. This confirms our belief that no parameter setting, besides population size, performs better than other.

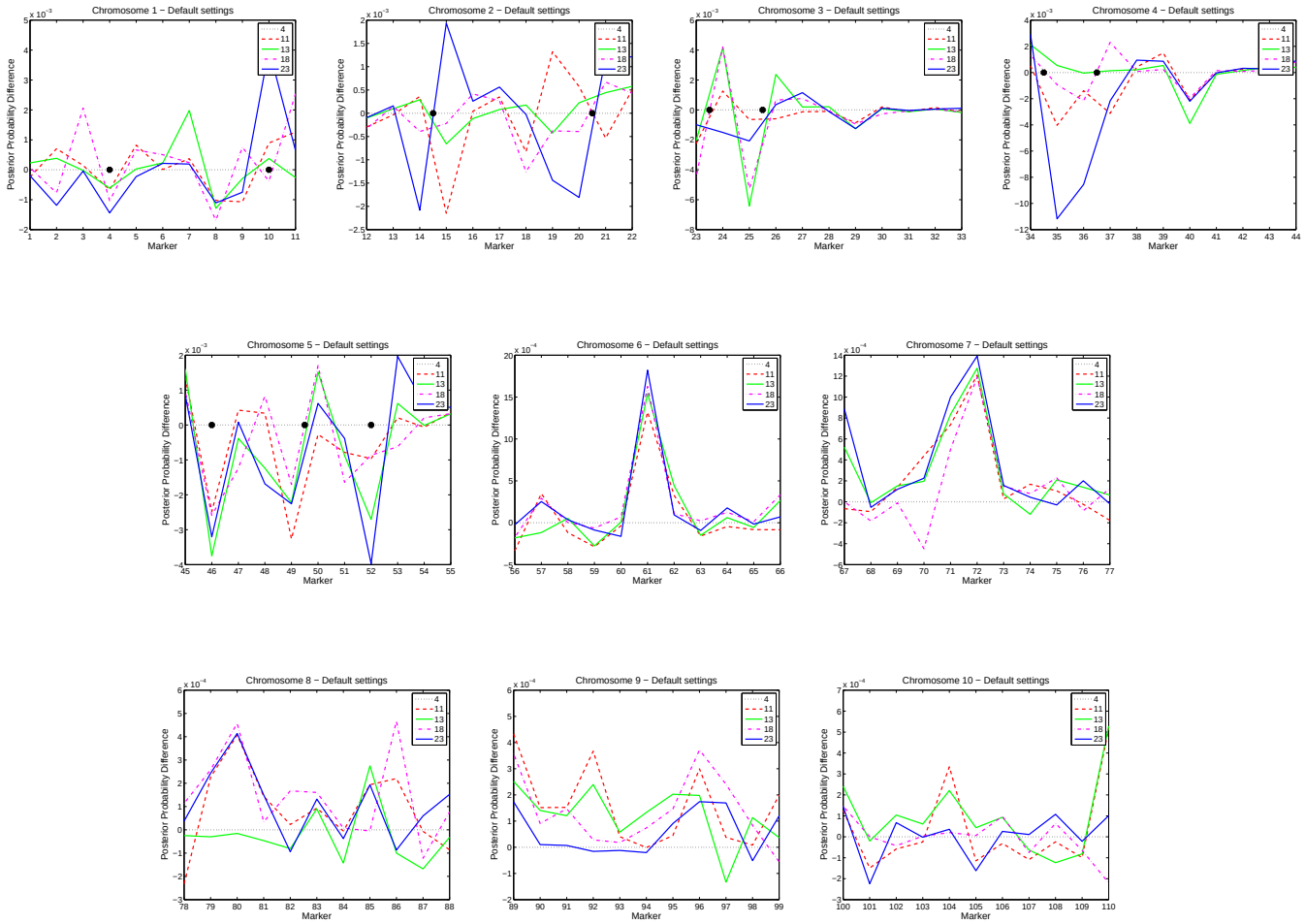

Figure 4.33: Difference of mean posterior probabilities for all default settings. QTL positions are indicated by a dot at the bottom.

Just as for scenario 1 we will now quantify the differences between the performance of parameter settings more systematically.

Quality measures for posterior probabilities

Since scenario 2 includes QTLs that are located between two markers, a new definition for L_2 is needed. That is

$$L_2 = \sqrt{\sum_{q \in R} (1 - \sum_{i \in U_q} \pi_i)^2 + \sum_{i \in V \setminus \bigcup_{q \in R} U_q} \pi_i^2}$$

where π_i is the posterior probability for marker i,
R is the true location of the QTLs, for scenario 2, that is,
$R = \{4, 10, 14.5, 20.5, 23.5, 25.5, 34.5, 36.5, 46, 49.5, 52\}$,
V is the set of all markers, that is, $V = \{1, 2, ..., 110\}$
and

$$U_q = \begin{cases} \{q\} & \text{if } q \text{ is located at a marker} \\ \{\lfloor q \rfloor, \lceil q \rceil\} & \text{if } q \text{ is located between markers.} \end{cases}$$

In other words, if a QTL is located directly at a marker, the posterior probability directly at that marker is used in the calculations of L_2, but if a QTL is located between two markers, the sum of the posterior probabilities of the two flanking markers is used to calculate L_2.

Figure 4.34 shows the mean L_2 and the standard error of the mean for different parameter settings.

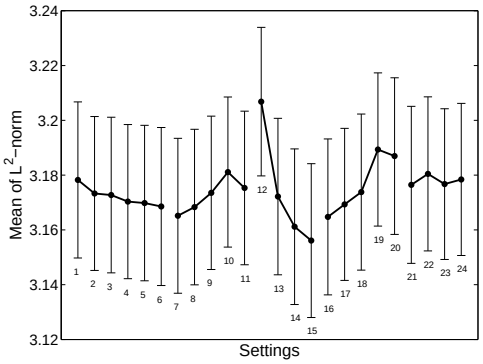

Figure 4.34: Mean L_2 and standard error of the mean for 24 different parameter settings

Within most parameter setting groups, the difference of the mean L_2 is rather small. The largest difference can be seen between population size settings (settings

12-15). To test if the mean L_2 is different between parameter settings we perform a repeated measures ANOVA F-test for the null hypotheses

$$\mathrm{H}_{0P_q} : \text{the mean of } L_2 \text{ is the same for all parameter settings in } P_q$$

where P_q is the set of parameter settings in group $q = 1, ..., 5$. The resulting p-values are displayed in Table 4.5.

Settings	1-6	7-11	12-15	16-20	21-24
p-value	0.2487	0.0311	$2.2091 \cdot 10^{-17}$	$1.9454 \cdot 10^{-5}$	0.9343

Table 4.5: p-values from repeated measures ANOVA F-test on the mean of L_2 for all parameter setting groups

These p-values and a significance level of $\alpha = 0.01$ indicate that the mean L_2 is different within parameter setting groups P_3 (population size) and P_4 (tournament size). According to these results and Figure 4.34 we can assume that the L_2 declines when population size is increased and that the L_2 grows when tournament size is increased. Therefore, population size $u = 1000$ and tournament size $t = 2$ seem to perform best.
The p-value for parameter setting group P_2 is also quite small, although it is not smaller than $\alpha = 0.01$. According to Figure 4.34, mutation rate $p_m = 0.010$ could be performing better than other mutation rate settings.

Power, FWER and FPN

As for scenario 1 the power, family wise error rate (FWER) and the total number of false positives and false negatives (FPN) were estimated. As before, a QTL is classified as detected if a marker within 15 cM of the QTL is selected and a marker is classified as selected if its posterior probability is larger than 0.5 or 0.3 respectively. The difference of the power between different parameter settings can be seen in Figures 4.35 and 4.36.

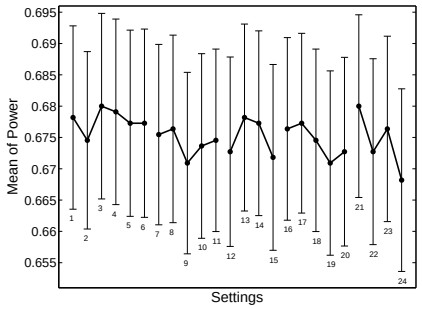

Figure 4.35: Mean power for selected marker posterior probability larger than 0.5 and standard error of the mean for 24 different parameter settings

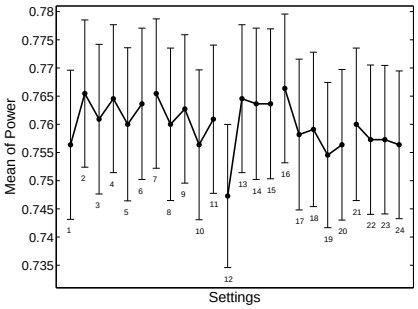

Figure 4.36: Mean power for selected marker posterior probability larger than 0.3 and standard error of the mean for 24 different parameter settings

The mean power is very similar among all parameter settings and the standard error of the mean is quite large for each setting. Only setting 12 seems to be performing considerably worse than others in Figure 4.36. A paired t-test for the hypothesis that the mean power of settings 12 and 13 are the same gives a p-value of 0.0005. We therefore conclude that setting 12 (population size $u = 100$) performs worse than other population size settings in terms of power when a marker is selected with posterior probability larger than 0.3.

An estimate of FWER is reported for selected markers with posterior probability larger than 0.5 and 0.3 (see Figures 4.37 and 4.38).
Again, no big differences can be seen between different parameter settings. The mean FWER of each parameter setting is very similar and within the standard error of the mean of other parameter settings.

The total number of false positives and false negatives are reported for detected markers with posterior probability larger than 0.5 and 0.3 (see Figures 4.39 and 4.40).

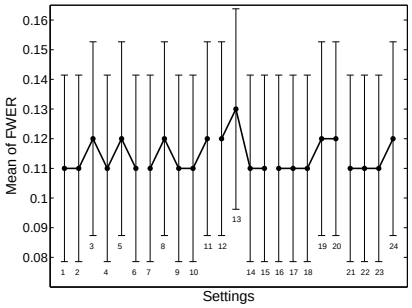

Figure 4.37: Mean family wise error rate (FWER) for selected marker posterior probability larger than 0.5 and standard error of the mean for 24 different parameter settings

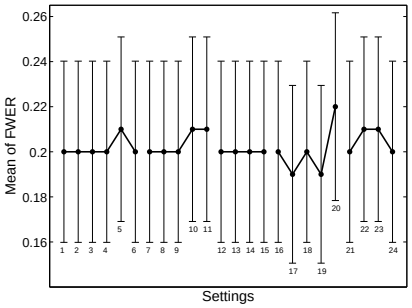

Figure 4.38: Mean family wise error rate (FWER) for selected marker posterior probability larger than 0.3 and standard error of the mean for 24 different parameter settings

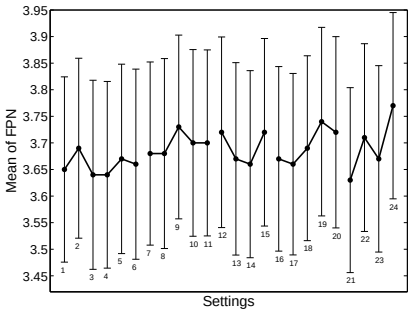

Figure 4.39: Mean total number of false positives and false negatives (FPN) for selected marker posterior probability larger than 0.5 and standard error of the mean for 24 different parameter settings

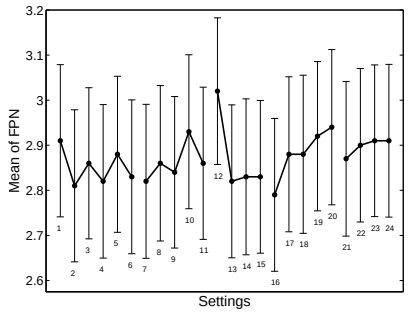

Figure 4.40: Mean total number of false positives and false negatives (FPN) for selected marker posterior probability larger than 0.3 and standard error of the mean for 24 different parameter settings

The mean FPN is very similar among all parameter settings although it is considerably higher for parameter setting 12 (population size $u = 100$) than for the other settings in Figure 4.40.

Cumulative sum of $\exp\left(-\frac{\mathbf{mBIC}}{2}\right)$

The mean cumulative sum of $\exp\left(-\frac{\text{mBIC}}{2}\right)$ for each parameter setting is reported in Figures 4.41 - 4.45 as was done for scenario 1.

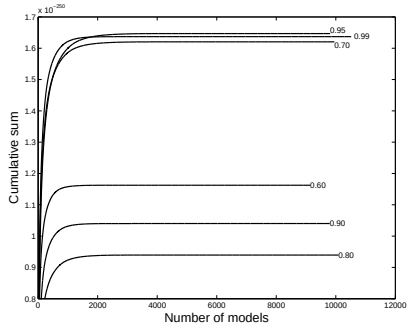

Figure 4.41: The cumulative sum of $\exp(-\frac{\text{mBIC}}{2})$ for different recomb. rates

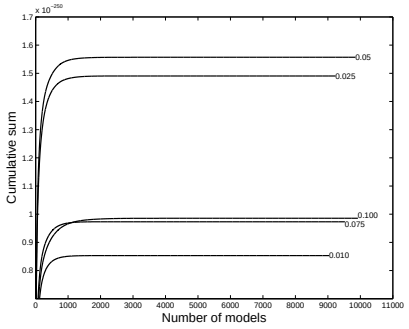

Figure 4.42: The cumulative sum of $\exp(-\frac{\text{mBIC}}{2})$ for different mutation rates

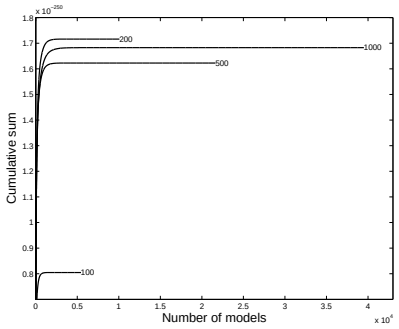

Figure 4.43: The cumulative sum of $\exp(-\frac{\text{mBIC}}{2})$ for different population sizes

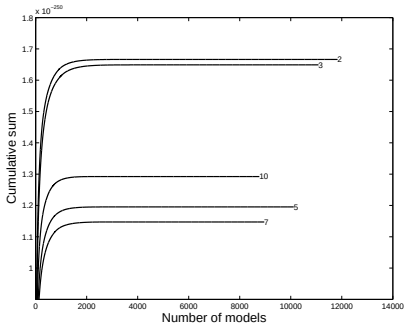

Figure 4.44: The cumulative sum of $\exp(-\frac{\text{mBIC}}{2})$ for different tourn. group sizes

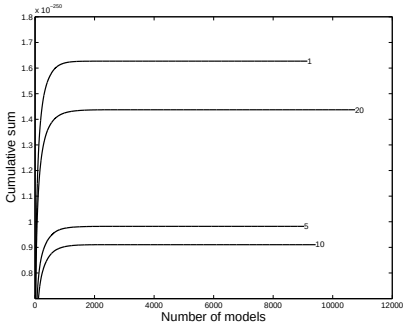

Figure 4.45: The cumulative sum of $\exp(-\frac{\text{mBIC}}{2})$ for different best model indices

No clear trend in the cumulative sum can be observed among the different parameter settings. However, larger population sizes (u=200, u=500 and u=1000) and smaller tournament sizes ($t = 2$ and $t = 3$) seem to be superior. The best performing settings are recombination rate $p_r = 0.95$, mutation rate $p_m = 0.05$, population size $u = 200$, tournament size $t = 2$ and best model index $B = 1$.

Summary

The posterior probability plots displayed in Figure 4.31 show that on average, most QTLs are detected, provided that they are not located too close to each other. QTLs on chromosome 3 are not distinguished and QTLs on chromosome 4 are not detected due to their proximity. For chromosome 4, population size $u = 100$ performs worse than larger population size settings.
According to the previously reported posterior probability plots, L_2 and power it is clear that when population size is increased, the GA performs better. However, as population size grows the overall runtime of the GA increases drastically. According to the L_2 and the cumulative sum plots lower tournament sizes are superior to others. The runtime of the GA does not increase considerably as tournament size is set lower and therefore we suggest using tournament size $t = 2$. For recombination rate, mutation rate and the best model index choosing parameters that minimize runtime is sensible.

4.2.3 Further investigation on population size

According to the previously reported quality measures for scenario 1, only tournament size $t = 2$ and larger population sizes seem to be superior. For all other parameters no one setting performs better than other. Since the runtime of different recombination, mutation and best model index settings does not change drastically, any of the previously examined values could be chosen for those parameters.
However, since the runtime increases greatly when population size is raised, it is tempting to choose the lowest value possible for population size. To investigate the performance of the GA when population size is decreased even further we simulated 1000 new data sets and applied the GA to each of them with different population size parameter settings.
Table 4.6 displays the different settings that were used. Simulations of the data sets were executed using scenario 1.

Setting	p_r	p_m	u	t	B
01	0.80	0.05	**10**	2	10
02	0.80	0.05	**25**	2	10
03	0.80	0.05	**50**	2	10
04	0.80	0.05	**100**	2	10
05	0.80	0.05	**200**	2	10

Table 4.6: Different population size parameter settings

Runtime, pool size, iterations and updates

Mean and standard deviation of runtime (measured in seconds), pool size, total number of iterations and total number of updates for different parameter settings based on 1000 simulation runs are reported. Figures 4.46 and 4.47 display changes in runtime and pool size respectively for different population size settings. Both runtime and pool size seem to increase linearly when population size grows, although pool size increases more rapidly. The reason is that for every extra individual in the population, local improvement is performed which produces a number of new models for the pool.

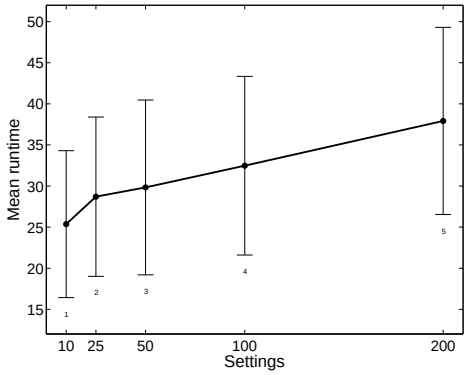

Figure 4.46: Mean runtime and standard deviation for 5 different population size parameter settings

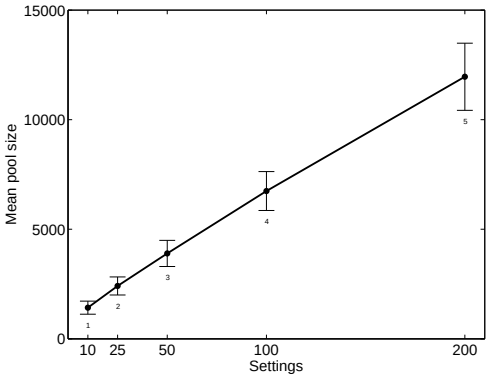

Figure 4.47: Mean pool size and standard deviation for 5 different population size parameter settings

Figures 4.48 and 4.49 display the mean and standard deviation of the total number of iterations and the total number of updates respectively (see "Iteration count" and "Update count" in Figure 3.4). Iterations decrease slightly when population size grows while updates increase. The probability that the same parents are chosen repeatedly is lower when population size is larger and therefore iterations decrease but updates increase.

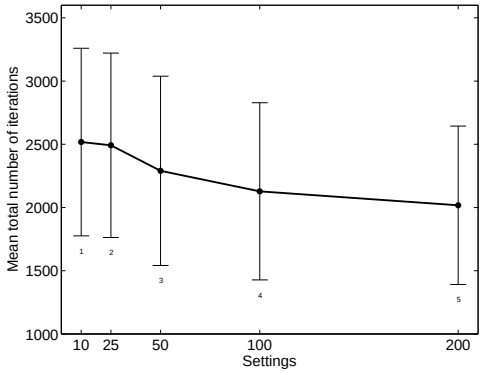

Figure 4.48: Mean total number of iterations and standard deviation for 5 different population size parameter settings

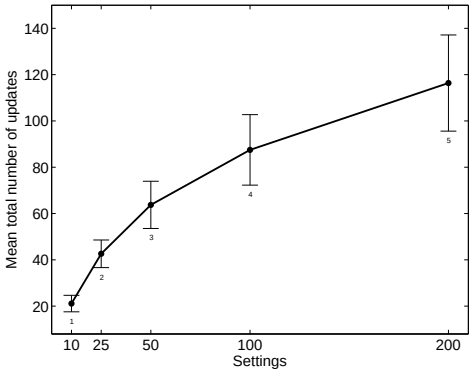

Figure 4.49: Mean total number of updates and standard deviation for 5 different population size parameter settings

Posterior probability plots

Posterior probability for each marker was computed according to equation (3.4). The mean of this marker posterior probability is displayed for every population size setting in Figure 4.50.
All QTLs are easily detected when we observe the mean posterior probability plots in Figure 4.50. No major differences between the population size settings can be observed. The posterior probability for chromosomes 6-10 is very low and can be assumed to be resulting from noise. It is interesting to see that on average all QTLs are easily detected even when the population size is very low.

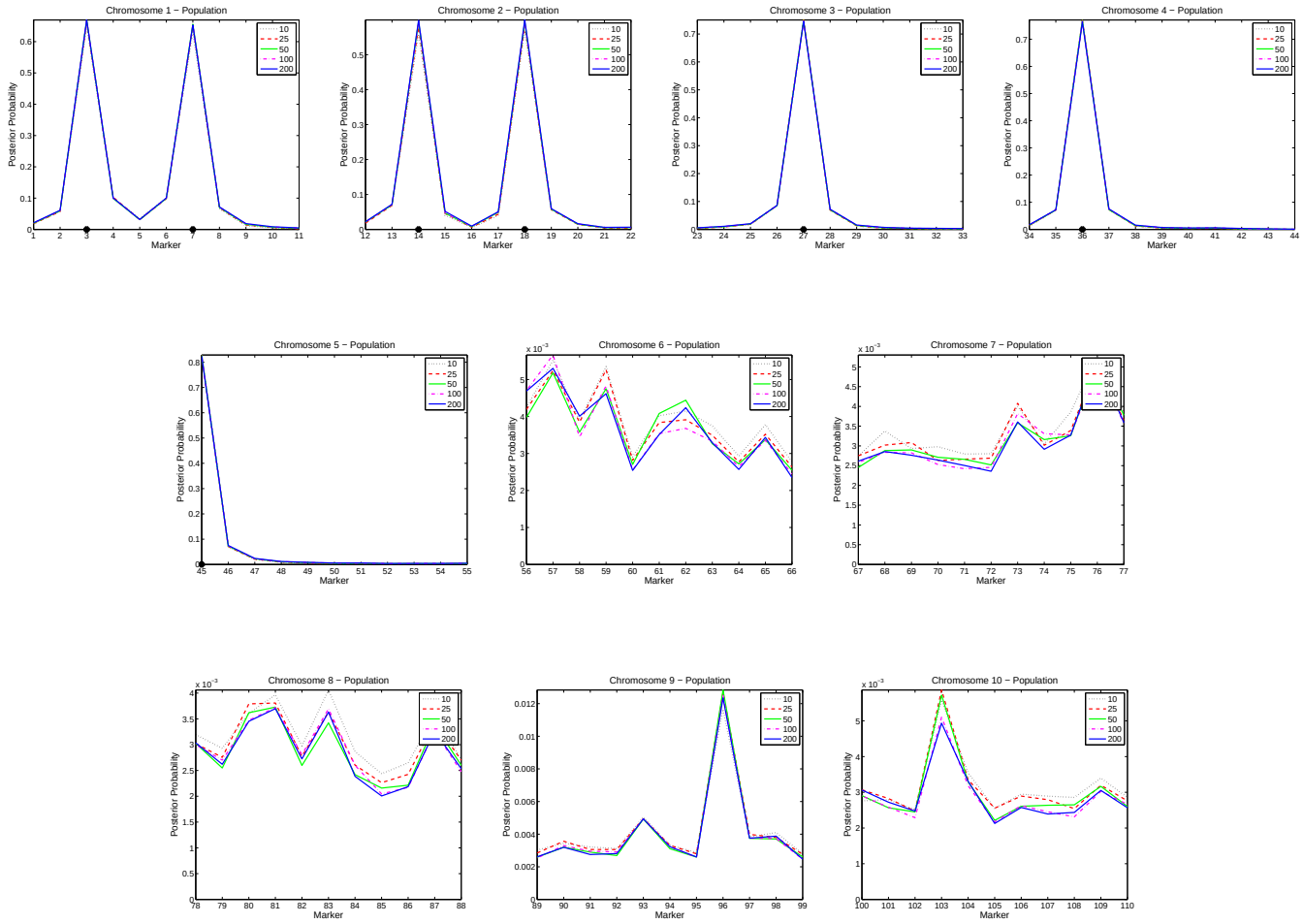

Figure 4.50: Mean posterior probability for different population settings. QTL positions are indicated by a dot at the bottom.

To examine the difference between the population size settings in more detail, mean posterior probability for each parameter setting was subtracted from the mean posterior probability of setting 1, that is, population size $u = 10$ (see Figure 4.51).

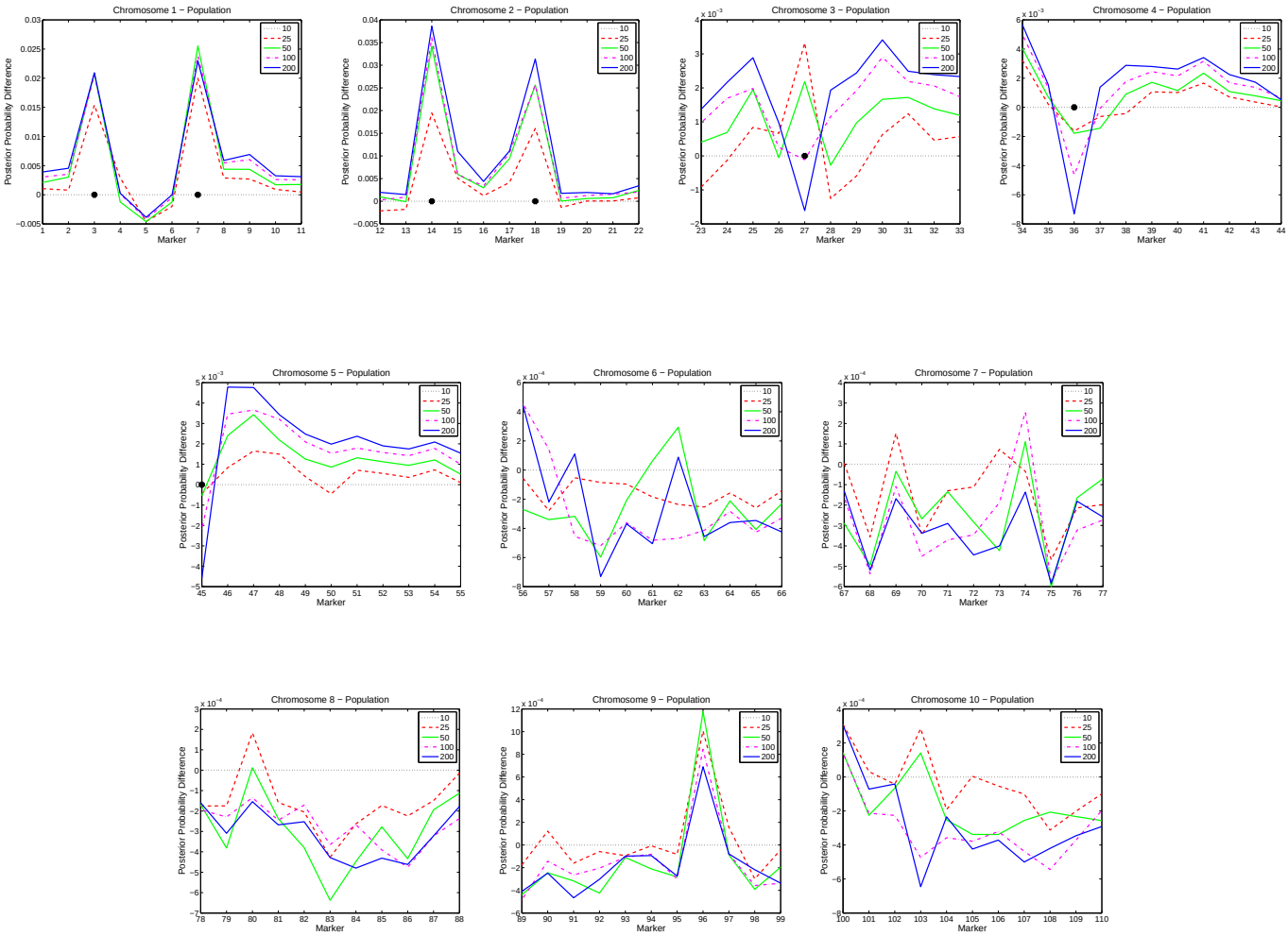

Figure 4.51: Difference of mean posterior probability for different population settings. QTL positions are indicated by a dot at the bottom.

In Figure 4.51 the difference between population size settings becomes more visible. At QTL locations, a peak in the difference is visible. On chromosomes 1, 2 and 5, the difference seems to increase as population size grows. The posterior probability differences on chromosomes 6-10 (no QTLs) is quite low and is assumed to be due to noise.

Quality measures for posterior probabilities

The L_2 and Q were computed according to equations (4.1) and (4.2) respectively. Figures 4.52 and 4.53 display the mean and standard error of the mean of the L_2 and Q for different population size settings.

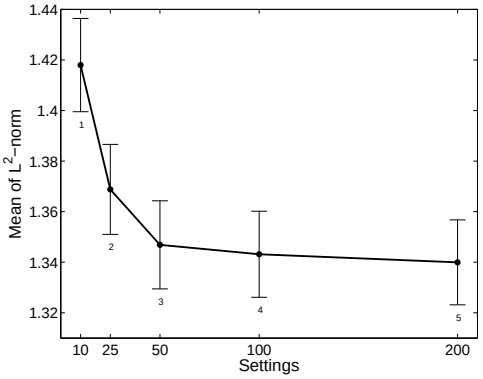

Figure 4.52: Mean of L_2 and standard error of the mean for 5 different population size parameter settings

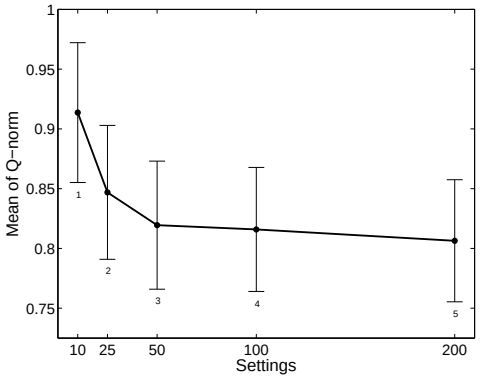

Figure 4.53: Mean of Q and standard error of the mean for 5 different population size parameter settings

The mean of L_2 and Q both decrease significantly when population size is increased from $u = 10$ to $u = 50$. It is interesting to see that the L_2 and Q decrease less and less with growing population size. One might conclude that for scenario 1 a population size of $u = 50$ or $u = 100$ is to some extent optimal, because the L_2 and the Q have reached their minimum, and the runtime only increases when further increasing the population size for this scenario.

To verify that both L_2 and Q improve when population size grows, a repeated measures ANOVA F-test is performed for the null hypothesis that the mean L_2 is the same for all population size settings and for the null hypothesis that the mean Q is the same for all population size settings. Table 4.7 exhibits the p-values of the F-tests.

	L_2	Q
p-value	$3.0571 \cdot 10^{-93}$	$1.0928 \cdot 10^{-98}$

Table 4.7: p-values from repeated measures ANOVA F-test of mean L_2 and mean Q

According to the p-values in Table 4.7, the null hypotheses can be rejected and we therefore conclude that the mean L_2 and mean Q is different between population size settings.

Power, FWER and FPN

Estimates of power, family wise error and the total number of false positives and false negatives are reported for a selected marker posterior probability larger than 0.5 and 0.3 respectively. The mean of estimated power is displayed in Figures 4.54 and 4.55.

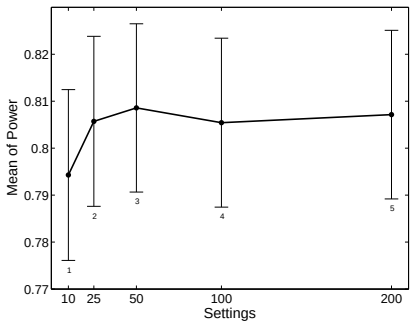

Figure 4.54: Mean power for selected marker posterior probability larger than 0.5 and standard error of the mean for 5 different population size parameter settings

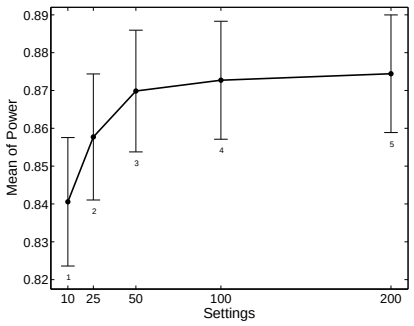

Figure 4.55: Mean power for selected marker posterior probability larger than 0.3 and standard error of the mean for 5 different population size parameter settings

In general, power increases as population size grows. Again, one might conclude that optimal power has been reached with population size $u = 50$ or $u = 100$.

Mean FWER is displayed in Figures 4.56 and 4.57.

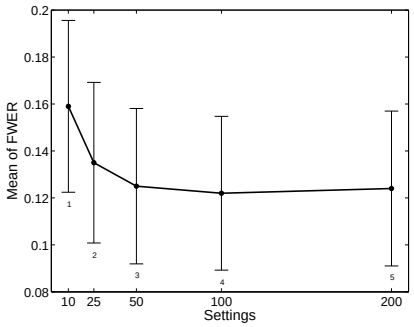

Figure 4.56: Mean FWER for marker posterior probability larger than 0.5 and standard error of the mean for 5 different population size parameter settings

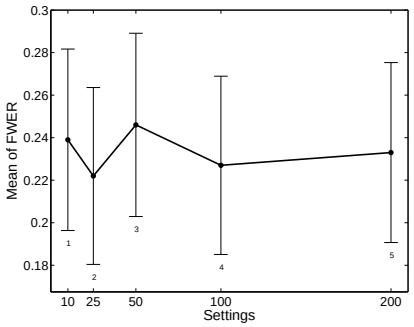

Figure 4.57: Mean FWER for marker posterior probability larger than 0.3 and standard error of the mean for 5 different population size parameter settings

For selected markers with posterior probability larger than 0.5, the mean FWER decreases as population size grows. The FWER for selected markers with posterior probability higher than 0.3 shows a different trend which might be a sign of random fluctuation.

Mean FPN are reported and displayed in Figures 4.58 and 4.59.

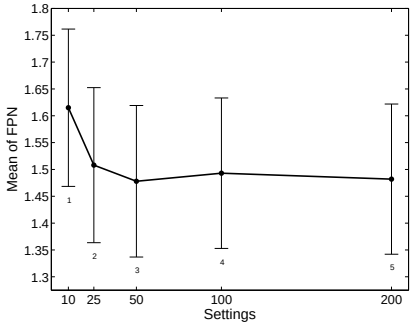

Figure 4.58: Mean FPN for marker posterior probability larger than 0.5 and standard error of the mean for 5 different population size parameter settings

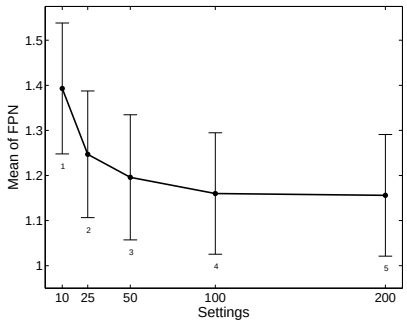

Figure 4.59: Mean FPN for marker posterior probability larger than 0.3 and standard error of the mean for 5 different population size parameter settings

Mean FPN decreases when population size increases. Again, one might decide, based on the total number of false positives and false negatives, that choosing population size $u = 50$ or $u = 100$ is appropriate.

Summary

According to various quality measures used throughout this paper, the performance of the GA decreases drastically when the population size is too small. To minimize runtime without loosing the performance power of the GA, using population size $u = 50$ or $u = 100$ for scenario 1 is suggested.
For more complicated scenarios larger population sizes might be beneficial as seen in scenario 2. Our general recommendation would be $u = 100$ or $u = 200$.

Chapter 5

Conclusion

In this thesis, a Genetic Algorithm (GA) was proposed to identify QTL locations. Our version of the GA includes problem dependent recombination and mutation operators as well as a local improvement strategy. The inclusion of the local improvement procedure increases the speed of convergence and provides a large pool of good models. This pool allows us to estimate the posterior probabilities of the marker-trait association which is possible because the modified version of the Bayesian Information Criterion is used as a fitness function in the GA.

In an extensive simulation study, various GA parameter settings were investigated using two different QTL scenarios. Runtime, pool size, total number of iterations, total number of updates, posterior probablity plots, L_2, Q, power, family wise error rate, total number of false positives and negatives and cumulative sum of $\exp\left(-\frac{\text{mBIC}}{2}\right)$ were reported and utilized to compare the GA's performance quality for different parameter settings. The cumulative sum plots for scenario 1 suggested that tournament size $t = 2$ performed better than other tournament size settings. This indication was confirmed in scenario 2, since the L_2 and cumulative sum plots showed the same results. Since runtime does not increase considerably as tournament size is set lower, we suggest using tournament size $t = 2$ when applying our version of the GA. For both scenarios the GA performance improves with increasing population size although the runtime grows drastically. For recombination rate, mutation rate and best model index, no one parameter setting performed better than any other. For those parameters we therefore suggested choosing the parameter setting that minimizes runtime.
Since population size influences runtime the most, further investigations on different population size settings were carried out using scenario 1 in order to determine how small the population size can be set without reducing the performance power of the GA. The result was that for rather small population sizes, the performance

of the GA deteriorates drastically. An optimal population size was hence suggested.

In this thesis, only backcross design was considered. Further analysis of the GA's performance quality using an intercross population could be quite easily executed using the previously described methods.
The simulations were carried out under scenarios where the trait is modelled by a normal distribution and without interactions between genes. Further research could include using the GA to localize additive and interaction effects for traits modelled by Generalized Linear Models.
Recently, FROMMLET, *et al.* (2012) proposed mBIC2 as a model selection criterion, which keeps the false discovery rate (FDR) for $n \geq 200$ below 10%. Analysis of the GA and its various parameter settings, using this FDR-controlling criterion is a candidate for further research.

Bibliography

[1] Akaike, H., 1974 *A new look at the statistical model identification.* IEEE Transaction Automatic Control **19**: 716-723.

[2] Baierl, A., M. Bogdan, F. Frommlet and A. Futschik, 2006 *On Locating Multiple Interacting Quantitative Trait Loci in Intercross Designs.* Genetics **173**: 1693-1703.

[3] Baierl, A., A. Futschik, M. Bogdan and P. Biecek, 2007 *Locating multiple interacting quantitative trait loci using robust model selection.* Computational Statistics & Data Analysis **51**: 6423-6434.

[4] Bogdan, M., J. K. Gosh and R. W. Doerge, 2004 *Modifying the Schwarz Bayesian Information Criterion to Locate Multiple Interacting Quantitative Trait Loci.* Genetics **167**: 989-999.

[5] Bogdan, M., J. K. Gosh and M. Zak-Szatkowska, 2008 *Selecting Explanatory Variables with the Modified Version of the Bayesian Information Criterion.* Quality and Reliability Engineering International **24**: 627-641.

[6] Broman, K. W. and S. Sen, 2009 *A Guide to QTL Mapping with R/qtl.* Springer, New York, USA, first edition.

[7] Broman, K. W., 2001 *Review of statistical methods for QTL mapping in experimental crosses.* Lab Animal **30(7)**: 44-52.

[8] Broman, K. W. and T. P. Speed, 2002 *A model selection approach for the identification of quantitative trait loci in experimental crosses.* J. R. Stat. Soc. B **64**: 641-656.

[9] Carlborg, Ö., L. Andersson and B. Kinghorn, 2000 *The Use of a Genetic Algorithm for Simultaneous Mapping of Multiple Interacting Quantitative Trait Loci.* Genetics **155**: 2003-2010.

[10] Dempster, A. P., N. M. Laird and D. B. Rubin, 1977 *Maximum likelihood from incomplete data via the EM algorithm (with discussion).* J. R. Statist. Soc. B **39**: 1-38.

[11] Doerge R. W., Z. B. Zeng and B. S. Weir, 1997 *Statistical issues in the search for genes affecting quantitative traits in experimental populations.* Statistical Science **12**: 195-219.

[12] Frommlet, F., A. Chakrabarti, M. Murawska and M. Bogdan, 2011 *Asymptotic Bayes optimality under sparsity for generally distributed effect sizes under the alternative.* Technical report, arXiv 1005.4753.

[13] Frommlet, F., I. Ljubic, H. B. Arnardóttir and M. Bogdan, 2012 *QTL mapping using a memetic algorithm with modifications of BIC as fitness function.* Statistical Applications in Genetics and Molecular Biology **11(4)**:Article 2

[14] Goldberg D. E., 1989 *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Publishing Company, Inc, Reading Massachusetts.

[15] Jansen, R. C., 1993 *Interval Mapping of Multiple Quantitative Trait Loci.* Genetics **135**: 205-211.

[16] Jansen, R. C. and P. Stam, 1994 *High Resolution of Quantitative Traits Into Multiple Loci via Interval Mapping.* Genetics **136**: 1447-1455.

[17] Kao, C. H., Z. B. Zeng and R. D. Teasdale, 1999 *Multiple interval mapping for quantitative trait loci.* Genetics **152**: 1203-1216.

[18] Lander, E. S. and D. Botstein, 1989 *Mapping Mendelian factors underlying quantitative traits using RFLP linkage maps.* Genetics **121**: 185-199.

[19] Lynch M. and B. Walsh, 1998 *Genetics and Analysis of Quantitative Traits.* Sinauer Associates, Inc. Publishers, Sunderland, Massachusetts, 01375 U.S.A.

[20] Michalewicz, Z. and D. B. Fogel, 2000 *How to Solve It: Modern Heuristics.* Springer-Verlag Berlin Heidelberg

[21] Mukhopadhyay, S., V. George and H. Xu, 2010 *Variable selection method for quantitative trait analysis based on parallel genetic algorithm.* Ann Hum Genet. **74**(1): 88-96.

[22] Nakamichi, R., Y. Ukai and H. Kishino, 2001 *Detection of Closely Linked Multiple Quantitative Trait loci Using a Genetic Algorithm.* Genetics **158**: 463-475.

[23] Namkung, J., J.-W. Nam and T. Park, 2007 *Identification of expression quantitative trait loci by the interaction analysis using genetic algorithm.* BMC Proceedings **1**(Suppl 1):S69.

[24] Obitko M, 1998 *Introduction to Genetic Algorithms.* Retrieved June 8 2011, http://www.obitko.com/tutorials/genetic-algorithms/.

[25] Schwarz, G., 1978 *Estimating the dimension of a model.* Annals of Statistics **6**: 461-464.

[26] Siegmund, D. O. and B. Yakir, 2010 *The Statistics of Gene Mapping (Statistics for Biology and Health).* Springer

[27] Soller, M., T. Brody and A. Genizi, 1976 *On the power of experimental designs for the detection of linkage between marker loci and quantitative loci in crosses between inbred lines.* Theoret. Appl. Genet. **47**: 35-39.

[28] Zeng, Z. B., 1993 *Theoretical basis for separation of multiple linked gene effects in mapping quantitative trait loci.* Proc. Natl. Acad. Sci. USA **90**: 10972-10976.

[29] Zeng, Z. B., 1994 *Precision Mapping of Quantitative Trait Loci.* Genetics **136**: 1457-1468.

[30] Zhu, M. and H. A. Chipman, 2006 *Darwinian evolution in parallel universes: A parallel genetic algorithm for variable selection.* Technometrics **48**: 491-502.

[31] Whitley, D., 1994 *A Genetic Algorithm Tutorial.* Statist. Comput. **4**: 65-85.

[32] Wu R. L., C. X. Ma and G. Casella, 2007 *Statistical Genetics of Quantitative Traits: Linkage, Maps and QTL.* Springer, New York, USA, first edition, 2007.

Zusammenfassung

In dieser Arbeit wird das Problem der Identifikation von "Quantitative Trait Loci" (QTLs) untersucht. In experimentellen Populationen können QTLs mithilfe multipler Regressionsanalyse lokalisiert werden. In diesem Zusammenhang hat sich in der Vergangenheit die Anwendung einer modifizierten Version des Bayesschen Informationskriteriums (mBIC) als Auswahlkriterium gut etabliert. Bisher wurden schrittweise Auswahlverfahren eingesetzt, um das Modell mit dem minimalen Wert des Auswahlkriteriums und infolge die mutmaßlichen QTLs zu finden. Obwohl schrittweise Auswahlverfahren in relativ geringer Zeit Lösungen produzieren, sind sie jedoch häufig nicht in der Lage, die global minimale Lösung zu finden. In dieser Arbeit wird ein Genetischer Algorithmus (GA) als mBIC minimierendes Verfahren vorgeschlagen. Desweiteren wird der Bayessche Ursprung des Auswahlkriteriums dazu genutzt, für alle vom GA untersuchten Modelle Marker-A-posteriori-Wahrscheinlichkeiten zu berechnen. Der GA hängt von verschiedenen Parametern ab. In umfangreichen Simulationsstudien auf Basis zweier verschiedener QTL-Szenarien wurden unterschiedliche Einstellungen dieser Parameter untersucht. In beiden Szenarien verbessert sich die Leistungsfähigkeit des GA mit steigender Populationsgröße sowie sinkender Turniergruppengröße. In einer finalen Simulationsstudie wird der Einfluss der Populationsgröße genauer untersucht um herauszufinden, wie klein dieser Paramater gesetzt werden kann, ohne dass die Fähigkeit des GA beeinträchtigt wird, die korrekten QTLs zu detektieren. Für die meisten anderen Parameter wird die Wahl jener Einstellungen vorgeschlagen, die die Laufzeit des GA minimieren.

CURRICULUM VITAE

Education

2008 - 2013 **University of Vienna**
Master studies in Statistics

2002 - 2005 **University of Iceland**
Bachelor of Science in Mathematics with a minor in Molecular Biology

1998 - 2002 **The Commercial College of Iceland**
Matriculation Exam from mathematical stream

1988 - 1998 **The Elementary School in Borgarnes, Iceland**

Academic experience

2007 - 2008 **University of Iceland**
Research Assistant at the Statistical Institute

2002 - 2005 **University of Iceland**
Teaching Assistant at the Mathematical Department

Publications

F. Frommlet, I. Ljubic, H.B. Arnardóttir, M. Bogdan
QTL Mapping Using a Memetic Algorithm with Modifications of BIC as Fitness Function
Statistical Applications in Genetics and Molecular Biology. Volume 11, Issue 4, May 2012.