UiO : Department of Informatics
University of Oslo

PyBayenv: A framework for interpreting, testing and optimizing Bayenv analyses

Kristoffer H. Ring

Master's Thesis Spring 2015

PyBayenv: A framework for interpreting, testing and optimizing Bayenv analyses

Kristoffer H. Ring

4th May 2015

Abstract

Loci involved in local adaptation may potentially be identified by the correlation between population allele frequencies and environmental variables. Several statistical methods for this purpose have been developed and a relatively new method known as BAYENV has become a popular and consequently receiving a lot of attention. By using a set of presumed neutral SNPs as a null model, BAYENV attempts to control for the effects of population structure when testing for correlation to environmental variables. BAYENV has proven to perform well when compared to the alternatives in studies evaluating differential based methods. However, there are several challenges associated with the BAYENV method. The use of Markov Chain Monte Carlo (MCMC) algorithms to evaluate complex statistical models makes the method vulnerable to a high run-to-run variability. Hence, it is recommendable to compare the results from several independent runs of the algorithm before drawing conclusions. Moreover, the method presents its results on the form of a Bayes Factor whose interpretation is not as well known as its frequentistic counterpart, the p-value - especially not in the context of multiple hypothesis testing. Additionally, the extensive use of MCMC algorithms, as well as a multi-step procedure for carrying out the analysis, makes BAYENV both time intensive and cumbersome to use.

Here we address several of the issues regarding the use of BAYENV and interpretation of its results. We propose an automated method to assign a significance level for an empirical distribution Bayes factors. The method, named the *Second Difference Method* (SDM), make use of the second difference to detect where the distribution has a significant change in the positive direction. By using SDM on the results from two SNP datasets, we find the method to be more reliable than conventional methods such as a percentage or static cutoff in terms of FDR. As a measure to reduce the overall time consumption of BAYENV, we suggest a method where SNPs with low allele frequency difference between populations are excluded from the test phase of BAYENV. This method showed promising results when tested on a dataset containing SNP data from Atlantic cod (Gadus morhua L.). To make the BAYENV analysis more user friendly and to test our hypotheses, we developed a wrapper program for BAYENV named PYBAYENV. Among other features in PYBAYENV, we implemented a mode where several instances of BAYENV were allowed to run in parallel. By parallelizing the process we were able to greatly reduce the time spent when performing multiple BAYENV analyses.

Contents

List of Tables

List of Figures

List of Algorithms

List of abbreviations

BF(s): Bayes factor(s)

DNA: Deoxyribonucleic Acid

FDR: False discovery rate

FN(s): False negative(s)

FP(s): False positive(s)

FPR: False positive rate

HWE: Hardy-Weinberg Equilibrium

LFMM: Latent Factor Mixed Model

MAFD: Maximum allele frequency difference

MCMC: Markov chain Monte Carlo

PCA: principal component analysis

SDM: Second difference method

SNP(s): Single Nucleotide Polymorphism(s)

TN(s): True negative(s)

TP(s): True positive(s)

TSS: Total significance set

Acknowledgements

I wish to express my sincere thanks to my supervisors, Anja Bråthen Kristoffersen and Karin Lagesen for valuable guidance and feedback throughout the research and writing process of the thesis.

A very special thanks goes to my third supervisor and colleague Ola Tveitereid Westengen. First of all for introducing me to this very interesting field of research by including me in his projects, but also for valuable help during the writing process of the thesis.

I am also grateful to Paul Ragnar Berg for not only giving me access to the SNP dataset used in this thesis, but also for including me in his research projects.

A warm thanks goes to my colleagues at Centre for Development and the Environment (SUM) for a lot of support throughout my part time studies. In particular I'm very grateful to my colleagues in the administrative staff at SUM for always cheering and encouraging me.

A special thanks goes to my colleague and fried Karen Syse for proof reading parts of the manuscript.

Finally, I want to thank my family for all the support, without them this thesis would not have seen the light of day.

Kristoffer Hofaker Ring

Oslo, May 2015

Chapter 1

General introduction

1.1 Introduction

Understanding the process of how species adapt to diverging environments is fundamental in ecology and evolution. The process is known as local adaptation and is an interplay between evolutionary forces such as selection, gene flow, genetic drift and mutation, where selection plays a leading role. Local adaptation is likely to take place if selection is spatially heterogeneous and strong compared to other evolutionary forces (Blanquart et al. 2013). By natural selection, some traits evolve to function better in a given context and thereby provide a higher degree of fitness to the affected population. Formally, a population is said to have been locally adapted if it exhibits higher average fitness in its native habitat than any other population introduced to the same habitat (Kawecki and Ebert 2004). Local adaptation is an important response to varying environmental conditions and may promote subdivision of a species into ecotypes, which again can potentially lead to the emergence of new biological species (e.g. Sobel et al. 2010). If the outcome of the process is a new species, the process is known as *speciation*.

Knowledge of genetic differentiation between populations may provide important functional information in the fields of agronomy and biomedical science. Such research can potentially identify interesting loci that can prove to be beneficial for the work on cultivated plants, livestock and humans (e.g decease loci) in particular (Bonhomme et al. 2010). Genetic shifts correlated with global warming have also been observed (e.g. Bradshaw and Holzapfel 2001). Hence, knowledge of genetic variation caused by local adaptation may prove crucial to ensure food security in a world undergoing rapid climate change (Lobell et al. 2008, 2011; Westengen et al. 2012).

The characterisation of genetic loci involved in local adaptation is central to understand phenotypic variation along environmental and

geographic gradients (known as clines - Huxley 1938) or between discrete environments caused by genetic differentiation has been observed in many animals (e.g. Nielsen et al. 2009) and plants (e.g. Alberto et al. 2013). However, despite many studies showing evidence of local adaptation, the genetic basis of local adaptation remains poorly understood (Savolainen, Lascoux and Merila 2013; Schlötterer 2002).

A particular challenge when looking for evidence of local adaptation is that it is hard to separate the complex effects of genetic drift (the random factor) and gene flow between populations from selection. In populations with low gene flow, genetic drift decrease local adaptation. However, if the gene flow is high, genetic drift has no effect on local adaptation (Blanquart, Gandon and Nuismer 2012). Hence, the balance between gene flow and selection is decisive for the extent of local adaptation (Savolainen, Pyhäjärvi and Knürr 2007). Another problem is that local adaptation may often result in subtle shifts in allele (an alternative form of a DNA segment at a specific locus) frequency at many loci where all make a small contribution to one particular phenotype (Hancock et al. 2010a). Identification of such loci (known as polygenic quantitative trait loci) that controls these traits is a challenging task (Savolainen, Lascoux and Merila 2013). Third, adjacent neutral genomic loci may be linked (alleles that are inherited together) and thus be hitchhiking to fixation along with loci under selection. Distinguishing between such loci may prove difficult.

There are several methods available for detecting molecular evidence of selection. (for a review see Nielsen 2005). In this thesis we will be focusing on a method that uses genetic differentiation among population. Genetic differentiation is the difference in allelic frequencies between populations that are caused by evolutionary forces such as genetic drift or selection. Currently there are three main differentiation based methods for detecting molecular footprints of local adaptation (Savolainen, Lascoux and Merila 2013): 1) Detection of population differentiation through scans of *Wright's fixation index* (F_{ST}, see Section 2.1.18), 2) $F_{ST} - Q_{ST}$ comparison (Q_{ST} measures the amount of genetic variance among populations relative to the total genetic variance in the trait. The $F_{ST} - Q_{ST}$ comparison is used to infer the action of natural selection on complex phenotypic traits. See Leinonen et al. 2013 for a review of this method) and 3) Correlation between allele frequencies and environmental variables.

Lately, several correlation based methods that accounts for the covariance of allele frequencies between populations have been developed (e.g. Bonhomme et al. 2010; Coop et al. 2010; Duforet-Frebourg, Bazin and Blum 2014; Frichot et al. 2013; Guillot et al. 2014) By controlling for neutral covariance it is easier to separate the effects of local adaptation from those due to shared population history and gene flow. In a recent simulation study, correlation based methods that accounted for underlying allele frequency structure proved to be more powerful and resulted in less false positive results than the methods based on F_{ST} (De Mita et al. 2013). In

particular, the method implemented in the program BAYENV (Coop et al. 2010) showed the highest statistical power (sensitivity) in some of the tests.

The BAYENV method uses a large set of presumed neutral loci to estimate the empirical pattern of covariance in allele frequencies between populations that accounts for the shared population history and gene flow. Given this neutral covariance as a null model, BAYENV test whether an alternative model that assumes a linear correlation between the population allele frequencies at a given locus and an environmental variable, is more probable than expected. The program uses a Monte Carlo Markov Chain (MCMC) scheme to estimate the null model (a covariance matrix) and to perform the tests for environmental correlation. For each locus, a Bayes factor (BF) is calculated as a measure of support for the alternative model. A high BF indicates higher correlation between the allele frequencies and the environmental variable than expected given the null model. BAYENV uses population allele frequencies on biallelic SNP (single nucleotide polymorphism, see Section 2.1.12) markers as input for the program.

BAYENV has since release in 2010 become a widely used and cited software in studies investigating genomic loci under selection in the context of local adaptation (e.g. Chen et al. 2012; Eckert et al. 2010; Evans et al. 2014; Fumagalli et al. 2011; Hancock et al. 2010b; Hancock et al. 2008; Hancock et al. 2011a; Heerwaarden, Hufford and Ross-Ibarra 2012; Westengen et al. 2014b). Moreover, the method has produced results that agrees with other differentiation based methods such as BAYESCAN (Foll and Gaggiotti 2008) and LFMM (Frichot et al. 2013) (see e.g. Berg et al. 2015, *in review* and Villemereuil et al. 2014). However, there are several issues with the BAYENV method that need further discussion.

First, the method provides its results on the form of BF. Interpretation of the BF is not as well known as its frequentistic counterpart, the p-value, particularly not in the context of multiple hypothesis testing. A common approach is to use a percentage or static cutoff on the empirical distribution of BFs. However, a percentage cutoff may often lead to many false positive results. Tables for different interpretation of BF values exist, but none of these are making any adjustments for multiple comparison tests.

Second, in a recent study (Blair, Granka and Feldman 2014) the stability of the BAYENV method was questioned. Data from an earlier study (Hancock et al. 2011b) was reanalysed and the results showed that some of the SNPs reported as highly significant in the original article showed no signals when it was rerun by Blair, Granka and Feldman 2014. Moreover, the study showed that there were in general a high variability between independent runs of BAYENV and the authors warned against making conclusion based on a single run alone.

The use of MCMC is known to be causing unstable results. MCMC algorithms needs to converge to a stable state in order be functioning as intended. How long the algorithm must run before this state is reach is

debated and a field of ongoing research. A common approach is to run as many iterations that is practically possible to be certain that the algorithm reaches equilibrium space. However, this often makes programs using MCMC algorithms particularly time consuming.

Some bioinformatics methods have developed a de facto standard for how many MCMC iterations and replicate runs that are needed to get a stable result. For example, for the program STRUCTURE (Falush, Stephens and Pritchard 2003; Pritchard, Stephens and Donnelly 2000), which is widely used to infer the population structure using allele frequencies, there have been published independent guidelines (e.g. Porras-Hurtado et al. 2013) and there is now a consensus for how the program should be run. A search in Google scholar shows that STRUCTURE is cited by about 14,000 studies (April 2015), a success caused by the quality of the method, but also by the user-friendliness of the original program and several supporting methods and programs (e.g. Earl and vonHoldt 2012; Evanno, Regnaut and Goudet 2005; Jakobsson and Rosenberg 2007).

By looking in the literature on how BAYENV is used, the lack of a uniform method for how the program should be run and how to interpret the results is evident. Moreover, the program is run in a multi-step procedure which makes it cumbersome and time consuming to use. The BAYENV specific file format may also cause difficulties for users without programming skills.

In this thesis we aim to address three particular challenges with the BAYENV program. First, given the challenge of interpreting the results, we aim to provide an automated method for assigning dynamic significance levels for distributions of BFs in the context of multiple hypothesis testing. We make use of the property of second difference to detect where the BF distribution makes a substantial jump in the positive direction and thereby separate significant from non-significant results. Thus the method was named the second difference method (SDM). Second, given the various practises of BAYENV (e.g. Blair, Granka and Feldman 2014; Chen et al. 2012; Fumagalli et al. 2011), we want to find an ideal set of settings for the BAYENV algorithm. We do this by examining the results from using different run length for the MCMC algorithms and by comparing different number of independent runs. Third, considering the fact that a full BAYENV analysis is very time consuming (De Mita et al. 2013), we propose a method of only including "interesting" SNPs as a time saving measure when carrying out the test for environmental correlation. By "interesting" we mean SNPs that exhibits a high difference in allele frequencies across populations. SNPs with a uniform allele frequencies are less interesting in this context since BAYENV is testing for a linear relationship to an environmental variable. We use the measurement maximum allele frequency difference (MAFD) between the populations to select the SNPs that are more likely to be the target of selection.

To help accomplish this, we developed a wrapper program, PYBAYENV, which in addition to streamlining the BAYENV procedures, serves as a

tool for testing our hypotheses. PYBAYENV has four main functions: 1) Conversion from a common file format (the GENEPOP format; Raymond and Rousset 1995) to the program specific BAYENV format. 2) Save time by running all the steps required by BAYENV in one go and parallelizing multiple runs to take advantage of today's multi-core CPUs. 3) Interpret the results from one or more BAYENV analyses by implementing our proposed second difference method (SDM). 4) Function for excluding SNPs from testing based on the maximum allele frequency difference (MAFD) between populations.

In this thesis we have used PYBAYENV to carry out BAYENV analyses on two different datasets to explore the strength and weaknesses of the BAYENV method and to test our hypotheses. The main data for the thesis is a dataset consisting of 8809 SNPs genotyped from Atlantic Cod (Gadus morhua L.). Parts of the results from these analyses were presented in the article *Adaptation to low salinity promotes genomic divergence in Atlantic cod* (Berg et al. 2015, *in review*). We also analysed a smaller dataset consisting of 135 SNPs from African Maize (Zea mays L.). Results from this analysis were published in the article *Modern maize varieties going local in the semi-arid zone in Tanzania* (Westengen et al. 2014b).

Chapter 2

Background theory

2.1 Basic biology

This section provides a brief introduction to the basics of biology necessary to understand the biological and evolutionary part of this thesis. For further details on the subject, the reader is referred to standard textbooks such as *Biology* (Campbell et al. 2008).

2.1.1 DNA

The basic unit in every living organism is the cell. The cell contains *DNA* (deoxyribonucleic acid), which is a chain of *nucleotides* that holds the genetic information that makes the organism what it is. DNA is *transcribed* into *RNA*, which again is *translated* into protein which are the building blocks of all organisms. The DNA molecule is organized as two parallel strands of nucleotides attached to each other by hydrogen bonds. This structure is referred to as a double helix. In DNA there are four types of nitrogenous bases: *Adenine* (A), *Guanine* (G), *Thymine* (T), and *Cytosine* (C)). The two strands in the double helix structure contain paired nucleotides that are *complementary* to each other. The base A is always paired with T, whereas G is paired with C. Such a pair is referred to as a *base pair*. This property implies that each strand contain all information that are present on the other strand. The DNA in the nucleus does not change and remains the same throughout life.

2.1.2 Organization of genome

The *genome* is the genetic material of an organism. The DNA molecule is curled and organized in *chromosomes* in the cell nucleus. The human genome is divided into 46 chromosomes which again can be divided into

23 pairs where one of each pair is inherited from each parent. Out of these pairs there are 22 *autosome* pairs that are identical between sexes and two *sex chromosomes* that differ between sexes and are sex determinant. The *ploidy* level denotes how many sets of chromosomes that are in the cell: *Haploid* has one set, *diploid* has two, *triploid* has three, etc. The somatic cells in humans are diploid, whereas the *gametes* are haploid. The term *polyploid* is used for cells with three or more chromosome sets.

2.1.3 The inheritance of the genome

The genetic material is passed on from one cell to another through cell division. There are two kinds of cell division: *mitosis* and *meiosis*. Mitosis occur in cells in somatic tissue where the genetic material is passed from the parent cell to the daughter cell through a process called DNA *replication*. Meiosis is a specialized division process that is used to produce gamete cells that only contains half the diploid complement of the genetic material making a haploid cell. The merging of two gametes in the fertilization process restores the diploid complement in the *zygote*. After Mitosis, the daughter cell is an exact replica of its parent cell (mutations do happen e.g. in the case of cancer). The gametes produced by meiosis, however, undergoes a process called *recombination* where paternal and maternal chromosomal homologs align and exchange DNA segments. This process is also known as *crossing over*.

2.1.4 Linkage Disequilibrium

Linkage Disequilibrium (LD) is the nonrandom correlation between specific alleles at different loci. It is a measure of recombination at the population level. A high LD between two loci indicates that these are seen more often together than would be expected by chance alone. There are several statistical descriptors of LD, where the most commonly used summaries are $\mathbf{D}$, $\mathbf{D'}$ and $\mathbf{r}^2$ (see Nordborg and Tavaré 2002 for a review).

2.1.5 Genotype and phenotype

The *genotype* of an organism is the genetic material (DNA) that is inherited from the parents. The genotype is constant for the entire lifetime and somatic mutations can (but not always) cause cancer or other diseases.

Phenotype is the set of traits of an individual. Examples of such traits can be eye color, skin color, diseases, etc. In addition to be influenced by the genotype, phenotype can also be affected by environmental factors like temperature, nutrition and diseases. Unlike the genotype, the phenotype may change throughout the lifetime of the organism. Furthermore, different genotype can also result in the same phenotype.

2.1.6 Genes

A gene is a subsequence of the DNA molecule, normally a couple of thousand nucleotides long, which codes for a protein that is used by the cell. In a human cell there are approximately 20000 genes. Sometimes there is a simple connection between a gene and a trait, eg. people with blue eyes lacks a protein that makes brown pigment, however, in most cases the connection between genes and traits are more complex. External conditions, the environment, can to a greater or lesser extent affect gene expression and thereby the traits they contribute to. Despite having over 20000 genes in the human DNA, most of the DNA molecule is *non-coding*. The non-coding DNA sequence is the components of the DNA that does not encode protein sequences. Actually, more than 98% of the human DNA mass does not code for any protein. These non-coding areas were earlier often referred to as "junk DNA". However, this term is no longer regarded as valid since we now know that gene's regulation is far more complex than previously thought. Recent research suggests that 80% of the human genome serves some purpose (Pennisi 2012).

2.1.7 Polygenic traits

A polygenic trait is a phenotypic trait that is influenced by more than one gene. Many traits in humans and other species are considered to be controlled by a large number of small effect loci. Moreover, genome-wide association study (GWAS) has shown that many quantitative traits in humans are highly polygenic and recent research suggest that most adaptive events are caused by polygenic adaptation and not by selective sweeps alone (Pritchard and Di Rienzo 2010).

2.1.8 Loci and markers

A *locus* (plural: *loci*) is the specific location of a DNA segment. A genetic *marker* is the specific gene or DNA sequence with a known location that can be used to identify individuals or species. Examples of such genetic markers are *single nucleotide polymorphism* (*SNP*), *microsatellite polymorphism* (*SSR*) and *restriction fragment length polymorphism* (*RFLP*). The nomenclature for a genetic locus is often given as numeric combination of the chromosome number and physical location on the chromosome.

2.1.9 Alleles

An *allele* is one of several forms of a DNA segment at a given locus. Usually, the term is used in conjunction with genes, but can also be use for variants of non-coding areas in the DNA.

The *genotype* is the set of alleles that are carried by an individual at a given locus. If there are **n** alternative alleles there will be **n(n+1)/2** possible genotypes.

2.1.10 Homozygous and Heterozygous

Consider a diploid organism with two alleles A and B at a particular locus. In this case there are three possible genotypes, namely: AA, AB and BB. In the case of AA and BB, we say that the individual is *homozygous* for the allele A and B respectively. However, if the alleles differ (e.g. the genotype is AB) the individual is said to be *heterozygous*.

2.1.11 Mutations

A genetic *mutation* is when there is a change in the DNA sequence of an organism. This happens when a base is replaced (*substitution*), removed (*deletion*) or duplicated (*duplication*). There are two main types of mutations: The ones that happen suddenly in somatic body cells and mutations during meiosis in the gametes. Whereas sudden mutations in somatic cells (replication errors) result in cell death or in worst case cancerous cell growth, mutations in gametes (recombination) are necessary for evolution to take place. Mutations in gametes ensure that new alleles are being created. Mutations do also happen in non-coding areas and even though it does not affect the protein structure directly, the expression of the protein might change.

The term *mutation rate* refers to how often different kinds of mutations (e.g. rate of substitutions) occur in an organism along a time scale.

2.1.12 Single Nucleotide Polymorphism

Small differences in the DNA sequence may have significant impact on the phenotype. Point mutations where only one base is substituted with another may cause faults (diseases or other defects), but in rare occasions also beneficial effects for the organism. If the latter is the case, this new allele may survive and even become dominant if it is more beneficial than others. If a point mutation allele become a variation that is found in at least 1% of the population, it is called a *single nucleotide polymorphism* [1] or *SNP* (pronounced *snip*)(see Figure 2.1) (Jobling et al. 2013). A variant allele that is present in less than one percent of a populations, is named a *variant* or sometime also referred to as a *single nucleotide variant* (SNV). In humans, SNPs occur on average once in 100 to 300 base pairs, which

[1] Variations in DNA sequence are called polymorphism (from Greek for "many forms" Campbell et al. 2008, Chapter 20.4)

means that there are roughly 10 million SNPs in the human genome. SNPs that are found in coding areas in the genome can be divided into two main groups: *synonymous* and *non-synonymous*. A non-synonymous SNP will affect the amino acid sequence produced by the gene, whereas the protein will stay unchanged in the synonymous case. Non-synonymous SNPs can cause differences in the gene expression and thereby give an increased risk of getting a particular decease or affect the response to a certain drug. SNPs found in genes and regulatory regions near genes may be very useful markers that can aid scientists to identify loci associated with different phenotypic traits. By identifying such SNPs it may be possible to provide specialized treatments for a particular condition or disease. For example, *activated protein C resistance* is an inherited condition that causes an increased risk of blood clotting in humans (Stefano and Leone 1995). Among the risk factors for the condition are SNPs in genes coding for blood coagulation factors (e.g. SNP G1691A Factor V Leiden - see Almawi et al. 2005). By knowing which SNPs that are causing the condition it possible to take preventive measures for patients who are known to have inherited these SNPs. For example, patients known to be homozygous for the Factor V Leiden mutation (the G1691A SNP) may be offered anticoagulants or alternative treatments when exposed to associated risk factors (i.e. pregnancy, surgery or oral contraceptives).

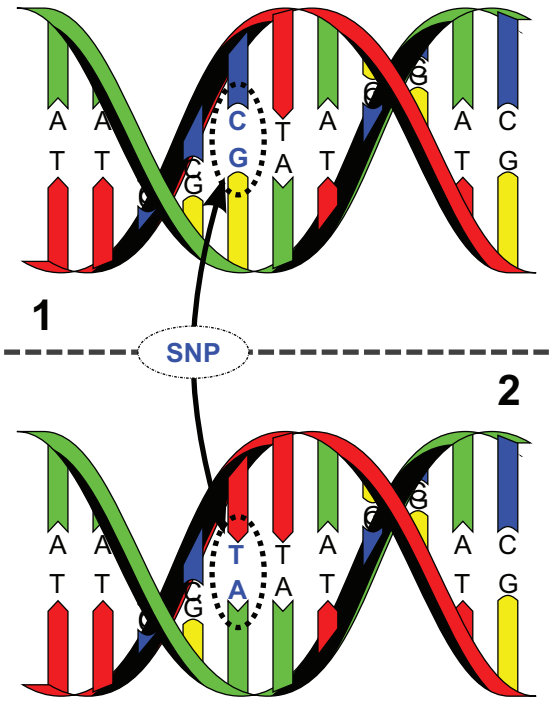

Figure 2.1: Illustration of a SNP. The base pair G/C is substituted by A/T in the DNA sequence.
Source: http://commons.wikimedia.org/wiki/File:Dna-SNP.svg

2.1.13 Dominant and recessive alleles

Assume two alleles, B and b at a locus. If the homozygous genotypes BB and bb lead to different phenotypic traits and the heterozygous genotype Bb cause the same traits as BB, then the allele B is said to be *dominant*. Alleles that only lead to expression of traits in the homozygous case are known as *recessive*. Many genetic diseases are recessive and thus require both copies of the allele to be present in order to be expressed. The individuals that are heterozygous and thus phenotypically normal is considered *carriers* of the disease. When producing only half the gene product is sufficient for normal or near-normal function, the situation is referred to as *haplosufficiency* (i.e. a haploid dose is enough) Jobling et al. 2013, Chapter 3.1.

2.1.14 Natural selection, genetic drift and gene flow

Natural selection is the most fundamental mechanisms of evolution. It is Charles Darwin's most famous theory from *On the Origin of Species* and states that individuals with characteristic traits that are beneficial for their probability of survival will have more opportunities to reproduce, thus their offspring will also benefit from these heritable, advantageous traits. By constantly favouring some allele over others, natural selection can cause *adaptive evolution*, i.e. changes that result in a better match between organism and their environment (Campbell et al. 2008, Chapter 23.3). Depending on the phenotype in the population that is favoured, natural selection can alter the frequency of heritable traits in three different ways (Campbell et al. 2008, Chapter 23.4): 1) *Directional selection*, where the frequency curve is shifting towards one extreme phenotypic trait. 2) *Disruptive selection*, where individuals with extreme phenotypic traits in several directions are favoured over individuals with intermediate phenotypic traits. 3) *Stabilizing selection*, where individuals with intermediate phenotypic traits are favoured over the extreme. In the latter case extreme variants are removed from the population.

Whereas natural selection favours the most beneficial alleles, *genetic drift* adds a random factor to the equation. Due to pure chance, some alleles can happen to survive better than others and thus become dominant or even fixed in the population. The effect of genetic drift tends to be more evident in small populations.

An effect known as the *Bottleneck effect* may occur when there are sudden changes in the environment (like natural disasters, draught, flood, etc.). In this case some alleles will survive passing through a restrictive "bottleneck", just by chance alone. In this way certain alleles may be overrepresented among the survivors while others again might be lost. However, if the population is large, chance events will have less effect on the population allele frequency (Campbell et al. 2008, Chapter 23.3).

The term *Gene flow* refers to migration between subpopulations. Transfer of alleles in and out of populations may change the allele frequencies.

2.1.15 Speciation

For eukaryotic organisms, a species is defined as the largest population group that can interbreed and produce fertile offspring. The term *speciation* refer to the evolutionary process that leads to the emergence of a new biological species. Ecotypes are variations or races within the same species that can interbreed without loss of fertility. Typically, ecotypes exhibits different phenotypic traits caused by spatial heterogeneity and may be the first step towards *Parapatric* speciation.

2.1.16 Hardy-Weinberg Principle

The Hardy-Weinberg Principle is a mathematical theory that describes a hypothetical population that is not evolving, i.e. the gene pool remains the same from one generation to the next. It was developed in 1908 by G. H. Hardy, a British mathematician and Wilhelm Weinberg, German physician and is founded in the Hardy-Weinberg Equation (HWE). The equation states that at a locus with two alleles, the three possible genotypes will be appear in the following proportions:

$$(P_A + P_B)^2 = P_A^2 + 2P_AP_B + P_B^2 = 1$$

The HWE is used to test whether a population is evolving or not by comparing the empirical distribution in the population to the theoretical given by the HWE. However, in order to be valid, HWE assumes that five conditions are met and several of these are often violated in real populations (Campbell et al. 2008, Chapter 23.2). The conditions to be met and associated problems are listed below:

1. **No mutations**.
   Problem: By altering genes, mutations modify the gene pool.

2. **Random mating**.
   Problem: Individuals usually mate preferentially with a subset of the population, thus random mixing of gametes will not occur.

3. **No natural selection**.
   Problem: Difference in survival and reproductive success can alter allele frequencies.

4. **Extremely large population size**.
   Problem: The smaller population, the more likely it is that the allele

frequencies will be altered by chance over generations (genetic drift)

5. **No gene flow**.
   Problem: Gene flow can alter the allele frequencies by moving alleles in and out of the populations.

Even though it is hard to satisfy all the conditions of the HWE, the equation is widely used to do estimations of population frequencies. For example, HWE can be used to estimate the percentage of the populations that carries the allele of an inherited decease. In this case it is assumed that the mutation rate for the gene causing the decease is low, inbreeding is uncommon, selection only occurs against the rare homozygotes and that the population size is large. It is important to remember that HWE only yields approximations to the real percentage of carriers of an allele.

The HWE can be generalized for n distinct alleles in m-ploid as $(P_1 + \cdots + P_n)^m = 1$.

2.1.17 Genotype frequencies and allele frequencies

The *genotype frequency* of a population is the number of individuals with a particular genotype divided by the total number of individuals in the population. Given two alleles, A and B, the genotype frequency for each genotype can be denoted P_{AA}, P_{AB} and P_{BB}. The *allele frequency* of a population is the number of individuals with a given allele divided by the total number of individuals in the population. Again consider the case of two alleles A and B, then the relationship with genotype frequency is as follows:

$$P_A = 2P_{AA} + P_{AB}$$
$$P_B = 2P_{BB} + P_{AB}$$

The empirical allele frequency of a population can be computed using the following strategy:

Let x_i and y_i be the observed counts of respectively allele 1 and allele 2 on individual i, then the empirical allele frequency f in a population k can be determined by the equation:

$$f_k = \frac{\sum\limits_{\forall i \in k} x_i}{\sum\limits_{\forall i \in k} (x_i + y_i)} \tag{2.1}$$

This equation returns the frequency f_k of allele 1 in population k.

Moreover, the sum of the two allele frequencies is 1, thus it can also be seen as a measure of the empirical probability for each allele in the population. Consequently, f_k^c is the empirical frequency of the second allele; which can be calculated by the equation $f_k^c = 1 - f_k$.

2.1.18 F-statistics

F-statistics (or Wright's F_{ST}) is a statistical method for measuring the proportion of genetic diversity due to allele frequency differences among subpopulations. It was developed by Sewald Wright and Gustave Malecot in the 1940s and 1950s and is a widely used statistic in population genetics. The equation states that

$$F_{ST} = \frac{(H_T - H_S)}{H_T}$$

Where H_T is the expected heterozygosity of the entire population and H_S is the mean expected heterozygosity across subpopulations. A guideline for how to interpret the F_{ST} values was given by the authors (see Table 2.1)Jobling et al. 2013.

F_{ST} value	Level of genetic differentiation
less than < 0.05	little
Between 0.05 and 0.15	moderate
Between 0.15 and 0.25	great
Greater than 0.25	very great

Table 2.1: Sewal Wright's qualitative guidelines for interpreting F_{ST}

2.2 Statistics and Mathematics background

2.2.1 About this chapter

Statistical and mathematical modelling plays a vital part in analysing and assessing genetic data. In this chapter we will give a brief introduction to: 1) Bayesian inference, which makes it possible to combine both empirical data and prior knowledge into the analysis, 2) Markov chain Monte Carlo (MCMC) algorithms which can be used to simulate and explore the resulting posterior distribution, 3) An introduction to finite difference, a numeric approximation to analytical derivatives, which can be used to identify change in growth rates for discrete data.

2.2.2 Bayesian inference

The Bayesian approach to statistical testing and method design is emerging as an increasingly effective and practical alternative to its frequentistic counterpart (Carlin and Louis 2011, Chapter 1.1). The computational revolution witnessed the last two decades is an important factor for this success. By using MCMC (see Section 2.3) algorithms and the computational power of programming languages such as *C*, *Python* or *R*, it is possible to explore and estimate probability distributions in higher dimensions.

The main principle in Bayesian analysis is that, in addition to sampling data, a *prior* distribution is required for all unknown quantities in the model. The *prior* and the likelihood of the data to compute the conditional distribution of the unknown quantities given the observed data. The resulting distribution is referred to as the *posterior* distribution. The *posterior* distribution is given by an equation known as the *Bayes' Theorem*:

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int_{\Theta} f(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}} \tag{2.2}$$

where $f(\boldsymbol{y}|\boldsymbol{\theta})$ is the likelihood function of the data $\mathbf{y}$ given the unknown parameters $\boldsymbol{\theta}$ and $\pi(\boldsymbol{\theta})$ is the prior distribution (or simply the *prior*). By observing that the integral in the denominator is just a scaling constant, the equation 2.2 may be expressed in the more convenient shorthand

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \propto f(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) \tag{2.3}$$

Notice that the *posterior* distribution is the *joint distribution* of the

likelihood (of the data) and the *prior*. In order to be a valid probability distribution, the *posterior* distribution is re-normalized so it integrates to 1.

The effect of the *prior* is that the *posterior* distribution will be "pulled" from the likelihood towards this distribution (see Figure 2.2). How much depends on how strong the *prior* is selected to be. The *prior* should reflect the certainty of the prior knowledge the analyst possess prior to the experiment. This prior knowledge is typically an "expert's" opinion or other available relevant information.

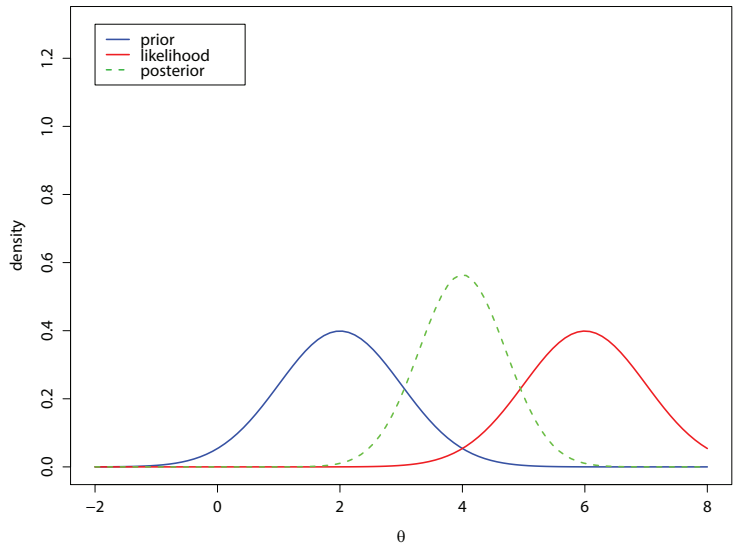

Figure 2.2: Plots of the posterior (green) of a normal prior (blue) and the likelihood (red).

2.2.3 Conjungate priors

The prior distribution $\pi(\theta)$ is called a *conjugate prior* of the likelihood function $f(y|\theta)$ if it leads to a posterior distribution $p(\theta|y)$ belonging to the same probability distribution family as the prior distribution. The prior and the posterior distributions are in such case called *conjugate distributions*. Conjugate priors is often chosen because it is computational convenient (Carlin and Louis 2011).

2.2.4 The Bayesian approach to hypothesis testing

There has been, and still is, an ongoing philosophical dispute between the two different approaches to hypothesis testing: the frequentistic and the Bayesian. The frequentistic accuse the Bayesian methods for being cumbersome and over-reliant of computationally convenient priors.

Whereas the Bayesians are criticising the frequentists for being unable to incorporate relevant prior knowledge into their models.

The traditional frequentistic approach to hypothesis testing is to have a *null* hypothesis, H_0 and an *alternative* hypothesis H_a. After finding a suitable test statistic, the observed significance or *p-value* is computed. The *p-value* is the evidence against H_0, or put another way: the probability of observing a more "extreme" value than the observed data. The *null* hypothesis is rejected if the *p-value* is less than a specified Type I error rate - typically $\alpha < 0.05$.

The frequentistic approach has for a long time been the favoured method for most researchers. However, it has also been the target of substantial criticism: First, the two hypotheses are *nested*, meaning that the *null* hypothesis must be a simplification of the *alternative* hypothesis. Second, the test or the *p-value* only offers evidence *against* the *null* hypothesis. Thus a significant *p-value* does not imply support for the *null* hypothesis, rather is evidence for no support for the *alternative* hypothesis. Third, the *p-value* cannot be directly interpreted as a "weight of evidence" for the two hypotheses. Therefore, it is often misinterpreted as "the probability that H_0 is true" (Carlin and Louis 2011, page 51).

The Bayesian approach to hypothesis testing was developed by Harold Jeffreys in the last century and published in the book *Theory of Probability, 1961*. The intent was not to give a complete new framework for hypothesis testing, but rather an addition to the existing tests. Not only does Bayesian hypothesis testing avoid the difficulties mentioned for the classical approach, but it also offers a simpler and more intuitive way of interpreting the results. Bayesian hypothesis testing has no limit on how many hypotheses that can be simultaneously considered and the hypothesis is not required to be *nested*. Due to this fact the notation is changed from "hypotheses" H_0 and H_a to "models" M_i, $i = 1, \ldots, m$.

Consider two candidate models M_1 and M_2 with data $\boldsymbol{Y}$ and parameter vectors $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$ respectively, then the *marginal distribution* under the prior densities $\pi_i(\boldsymbol{\theta}_i)$, $i = 1, 2$ is found by integrating out the parameters,

$$p(\boldsymbol{y}|M_i) = \int f(\boldsymbol{y}|\boldsymbol{\theta}_i, M_i)\pi_i(\boldsymbol{\theta}_i)d\boldsymbol{\theta}_i, \quad i = 1, 2. \tag{2.4}$$

To obtain the posterior probabilities $P(M_1|\boldsymbol{y})$ and $P(M_2|\boldsymbol{y}) = 1 - P(M_1|\boldsymbol{y})$ *Bayes' Theorem* is used (eq. 2.2). The Bayes factor, which is the ratio of the posterior odds of M_1 to the prior odds of M_1, is given by *Bayes' Theorem* as

$$
\begin{aligned}
BF &= \frac{P\left(M_1|\boldsymbol{y}\right)/P\left(M_2|\boldsymbol{y}\right)}{P\left(M_1\right)/P\left(M_2\right)} \\
&= \frac{\left[\frac{P(\boldsymbol{y}|M_1)P(M_1)}{p(\boldsymbol{y})}\right] / \left[\frac{P(\boldsymbol{y}|M_2)P(M_2)}{p(\boldsymbol{y})}\right]}{P\left(M_1\right)/P\left(M_2\right)} \\
&= \frac{\int f(\boldsymbol{y}|\boldsymbol{\theta}_1, M_1)\pi_i(\boldsymbol{\theta}_1)d\boldsymbol{\theta}_1}{\int f(\boldsymbol{y}|\boldsymbol{\theta}_2, M_2)\pi_i(\boldsymbol{\theta}_2)d\boldsymbol{\theta}_2} \\
&= \frac{p(\boldsymbol{y}|M_1)}{p(\boldsymbol{y}|M_2)}
\end{aligned}
\tag{2.5}
$$

which is the ratio of observed marginal densities for the two models. If the two models have equal prior probability ($P\left(M_1\right) = P\left(M_2\right) = 0.5$), $BF = P\left(M_1|\boldsymbol{y}\right)/P\left(M_2|\boldsymbol{y}\right)$, which is the posterior odds of M_1

If the models share the same parametrization and both hypotheses are simple, the Bayes factor is the same as the likelihood ratio between the two models. In this case the Bayes factor is the odds in favour of M_1 over M_2 given the data alone. Normally the Bayes factor is interpreted as "the evidence given by the data", however *Levine and Schervish (1999)* show that it is more accurate to say that BF captures the change in the odds in favour of M_1 as we move from prior to posterior (Carlin and Louis 2011, page 53).

2.2.5 The interpretation of Bayes factor

The Bayes factor (BF), which can be seen as a summary of the data in favour of one model against another, is given as a number between zero and infinity. Harold Jeffrey's gave in his book *Theory of Probability* an interpretation of the strength of evidence for BF (See Table 2.2). Later there has been several other alternative interpretations. Most notably is the interpretation in (Kass and Raftery 1995, see Table 2.3). However, the main principle is generally the same: A BF below one is a support for the null model, and conversely a BF above one is a support for the alternative model. The support for the alternative model increases as the Bayes factor rises above one.

$\log_{10}\left(BF\right)$	$\left(BF\right)$	Strength of evidence
< 0	< 1	Negative (supports H_0)
0 to $1/2$	1 to 3.2	Not worth more than a bare mention
$1/2$ to 1	3.2 to 10	Substantial
1 to 2	10 to 100	Strong
> 2	> 100	Decisive

Table 2.2: Jeffrey's interpretation of Bayes factors

$2\ln(BF)$	(BF)	Strength of evidence
< 0	< 1	Negative (supports H_0)
0 to 2	1 to 3	Not worth more than a bare mention
2 to 6	3 to 20	Positive
6 to 10	20 to 150	Strong
> 10	> 150	Very strong

Table 2.3: Kass and Raftery's interpretation of Bayes factors

2.3 Markov chain Monte Carlo methods

In probability theory it is often necessary to integrate out densities. However, in Bayesian analysis the probability functions are often too complex to be integrated analytically. In such case, the solution is to use Markov Chain Monte Carlo (MCMC) methods. MCMC is a range of computer simulation techniques where probability distributions in higher dimensions can be explored by sampling from the posterior distribution using "random walk" methods. Today, most Bayesian inference applications, such as STRUCTURE (Falush, Stephens and Pritchard 2003; Pritchard, Stephens and Donnelly 2000) and BAYENV (Coop et al. 2010), depend heavily on MCMC algorithms. In the following subsections we will give a brief introduction to two of the most frequently used MCMC procedures - namely the *Gibbs sampler* and the *Metropolis-Hastings* algorithm. Both algorithms operate by sequentially sampling parameter values from a Markov chain, where the Markov chain's *stationary distribution* is the target posterior distribution.

2.3.1 Introduction to MCMC

Consider a data set $\mathbf{D}$ and a set of parameters $\boldsymbol{\theta}$ that we want to find the most probable values of. Recall that Bayes theorem says that the posterior distribution of $\boldsymbol{\theta}$ given $\mathbf{D}$ is proportional to the joint distribution of the likelihood function and the prior distribution (see Section 2.2.2), thus the posterior can be written as

$$\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{D}) \propto p(\boldsymbol{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{2.6}$$

where $\pi(\boldsymbol{\theta})$ is the unormalized probability of the data that can be normalized by an unknown proportionality constant known as "the Bayes denominator". Markov Chain Monte Carlo (MCMC) algorithms is a way of drawing samples from the distribution $\pi(\boldsymbol{\theta})$ without knowing this normalizing constant. By drawing many samples from this distribution it is possible to compute any quantity of interest (i.e. the mean, standard deviation, confidence regions, etc.).

2.3.2 Metropolis-Hastings algorithm

The Metropolis algorithm was developed in 1953 by *Metropolis et al.* and was generalized by Hastings in 1970. In the year 2000 the Metropolis-Hastings algorithm was named as one of the top ten most influential algorithms of the 20th century by the journal *Computer in Science and Engineering* (*CiSE*). The *Metropolis-Hastings* algorithm works as follows:

Let $\pi(\boldsymbol{\theta})$ be a function that is proportional to the desired probability distribution $P(\boldsymbol{\theta})$ and choose an arbitrary starting value $\boldsymbol{\theta_1}$.

Pick a "proposal distribution" $q(\boldsymbol{\theta_2}|\boldsymbol{\theta_1})$ (e.g. a multivariate normal centred on $\boldsymbol{\theta_1}$). Then q is a transition function that tells us the probability of moving from $\boldsymbol{\theta_1}$ to a location $\boldsymbol{\theta_2}$.

First, generate a candidate point for $\boldsymbol{\theta_2}$ labelled $\boldsymbol{\theta_{2c}}$ by drawing from the proposal distribution around $\boldsymbol{\theta_1}$

Second, calculate an "acceptance" ratio by using the equation

$$\alpha(\boldsymbol{\theta_1}, \boldsymbol{\theta_{2c}}) = \frac{\pi(\boldsymbol{\theta_{2c}})q(\boldsymbol{\theta_1}|\boldsymbol{\theta_{2c}})}{\pi(\boldsymbol{\theta_1})q(\boldsymbol{\theta_{2c}}|\boldsymbol{\theta_1})} \tag{2.7}$$

notice that if q is symmetrical (i.e. multivariate normal) the q's are cancelled out, thus the equation can be simplified as

$$\alpha(\boldsymbol{\theta_1}, \boldsymbol{\theta_{2c}}) = \frac{\pi(\boldsymbol{\theta_{2c}})}{\pi(\boldsymbol{\theta_1})} \tag{2.8}$$

Because $\pi(\boldsymbol{\theta})$ is proportional to the true normalized distribution $P(\boldsymbol{\theta})$, we have that the acceptance ratio $\alpha = \pi(\boldsymbol{\theta_{2c}})/\pi(\boldsymbol{\theta_1}) = P(\boldsymbol{\theta_{2c}})/P(\boldsymbol{\theta_1})$.

In the third step, based on the acceptance ratio α, choose to keep or discard the candidate point $\boldsymbol{\theta_{2c}}$ such that

$$\text{if} \quad \alpha \geq 1, \quad \text{set} \quad \boldsymbol{\theta_2} = \boldsymbol{\theta_{2c}} \tag{2.9}$$

$$\text{if} \quad \alpha < 1, \quad \text{set} \quad \boldsymbol{\theta_2} = \begin{cases} \boldsymbol{\theta_{2c}} & \text{with probability} \quad \alpha \\ \boldsymbol{\theta_1} & \text{with probability} \quad 1-\alpha \end{cases} \tag{2.10}$$

The *Metropolis-Hastings* algorithm can, slightly inaccurate, be expressed in words as: Always accept a proposal that increases the probability, and sometimes accept it if it does not (see e.g. Carlin and Louis 2011, Section 3.4.2 for details about the Metropolis-Hastings algorithm).

2.3.3 Gibbs sampler

The Gibbs sampler is a special case of the Metropolis-Hastings algorithm. By holding all the coordinates except one fixed and let one coordinate vary, the *full conditional distribution* of some distribution $\pi(\boldsymbol{\theta})$ is found. Moreover, the theorem about full conditional distribution states if all the full conditional distributions are known, there exists a unique multivariate distribution that is consistent with them all (Carlin and Louis 2011). Formally the Gibbs sampler works as follows:

Given a model featuring k parameters, $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_k\}$. Under the assumption that samples can be generated from each of the conditional posterior distributions $p(\theta_i|\boldsymbol{\theta}_{j\neq i}, \boldsymbol{y})$, the *Gibbs sampler* is carried out using an arbitrary set of starting values $\left\{\theta_1^{(0)}, \ldots, \theta_k^{(0)}\right\}$ for the parameters. The algorithm proceeds as follows:

For $t = 1 \ldots, T$, repeat:

- **Step 1:** Draw $\theta_1^{(t)}$ from $p\left(\theta_1|\theta_2^{(t-1)}, \theta_3^{(t-1)}, \ldots \theta_k^{(t-1)}, \boldsymbol{y}\right)$
- **Step 2:** Draw $\theta_2^{(t)}$ from $p\left(\theta_2|\theta_1^{(t)}, \theta_3^{(t-1)}, \ldots \theta_k^{(t-1)}, \boldsymbol{y}\right)$

  $\vdots$
- **Step k:** Draw $\theta_k^{(t)}$ from $p\left(\theta_k|\theta_1^{()}, \theta_2^{(t)}, \ldots \theta_k^{(t)}, \boldsymbol{y}\right)$

The Gibbs sampler is *ergodic* (i.e. it theoretically will eventually over time sample all possible values of $\boldsymbol{\theta}$ from distribution we are interested in) and therefore will sample the full joint (posterior) distribution. Figure 2.3 shows how the Gibbs sampler is sampling from a bivariate normal distribution.

2.3.4 The concept of burn-in

It can be shown (Carlin and Louis 2011) that the k-tuple (the model parameters, see Section 2.3.3) obtained after t iterations in the Gibbs sampler converges to sample from the true posterior distribution. Provided that t is large enough, a draw from the sample distribution is therefore a (correlated) sample from the true posterior distribution. Any posterior quantities of interest can be estimated using this sample distribution. To avoid the instability due to inaccurate starting values, it is necessary to discard the first $t < t_0$ iterations of the algorithm. The time $t = 0$ to $t = t_0$ is known as the *burn-in* period. It is important that t_0 is sufficiently large to ensure that the sampling starts after it has reached its *equilibrium space*. Methods for assessing the sampler convergence exists, however, this is a field of ongoing research and a recurring challenge for the users of MCMC algorithms.

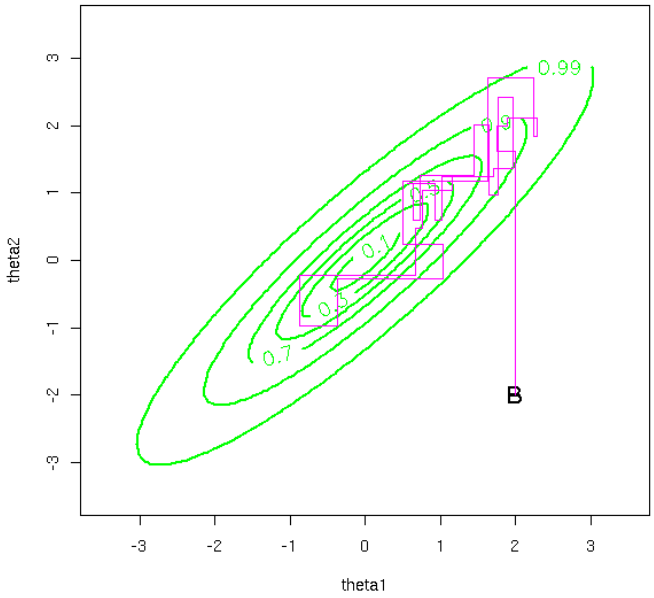

Figure 2.3: Gibbs sampling from a bivariate normal distribution. The purple staircase lines show how the sampling routine samples from one direction at a time starting from B and approaching the center of the distribution. Source: http://zoonek.free.fr/blosxom/R/2006-06-22_useR2006.html

Figure 2.4 shows a visualization of an MCMC simulation of a 3-variate *probability density function* (PDF). The flaming "star" in the top left corner is the peak in the PDF. The different coloured threads show how the algorithm converges from different starting positions (random seeds).

2.3.5 Random seed

The random seed is the starting point for the MCMC algorithm. In theory the starting point should not matter since the algorithm will always reach its target distribution after infinite iterations. However, in a finite world, different starting point can impact the number of iterations needed before the chain converges.

2.3.6 Disadvantages with MCMC

Although there are many advantages with MCMC, there are also disadvantages. One particular challenge is to know when to expect the algorithm to reach equilibrium space. If sampling from the posterior distribution starts before this stage, spurious samples may affect the end result. This is why a proper burn-in is required. However, as described above (section 2.3.4), the time before the posterior distribution reaches equilibrium

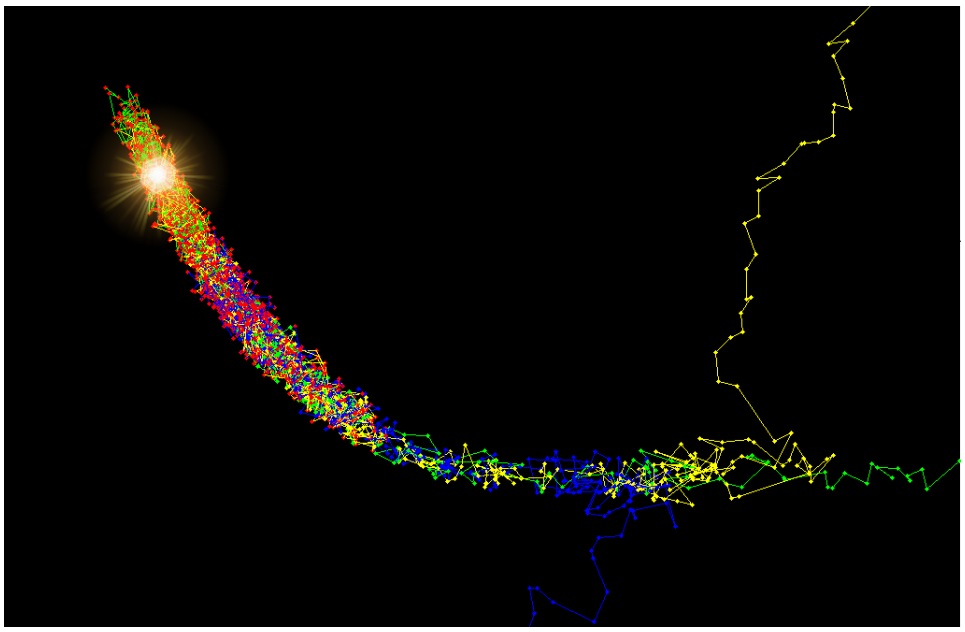

Figure 2.4: Visualization of burn-in and convergence of a MCMC simulation of a 3-variate PDF: the glowing spot in the top left corner shows the peak in the PDF, whereas the thin threads to the right (red, yellow, green and blue) shows how the Markov chain converges from different starting points. We can see that the starting point for the red chain is a more "lucky" one, starting closer to the center of the PDF. Source: http://www.juergenwiki.de/work/wiki/doku.php?id=public:mcmc

space can vary a lot depending on the random seed and the complexity of the model. Due to this problem users of MCMC algorithms often choose a burn-in period that exceeds what really is necessary in order to be absolutely certain that the estimation of the posterior distribution has reached a stable state. Moreover, the number of iterations after burn-in is usually chosen very high to ensure accurate parameter estimates from the posterior distribution. Additionally, due to high variability caused by sampling differences, it is often advisable to run the algorithm several times with different random seeds and compare the results. Altogether, these factors make the use of MCMC algorithms computationally slow compared to other methods.

2.4 Approximation of derivatives using finite difference

From calculus we know that the derivative can be used to detect various properties of a function. The first derivative at a point on a graph can be viewed as the slope of the line tangent at that point. The second derivative measures the *concavity* of a function. If the second derivative is positive, the function is *convex*. If the second derivative is negative, the function is *concave*. However, if the second derivative is zero, the point on the graph is an *inflection point*, meaning the point is a *transition point* where the graph changes curvature (e.g. goes from concave to convex). In computational mathematics, derivatives can be approximated using finite difference equations. We will in this section provide an introduction to a special case of finite difference, namely the second difference and provide an example from one of its applications.

2.4.1 Second difference

There are three different approaches to the second difference: *Forward difference*, *Backward difference* and *Central difference*. The main difference between the three forms is the error term which can be determined using Taylor series. By using Central difference, the error is an average of the two other methods and is therefore often the preferred variant. The approximation of second derivative using central difference can be derived as follows:

Let $u_i = u(x_i)$ and $i = 0, 1, \ldots, N$, then

$$\begin{aligned}\left(\frac{\partial^2 u}{\partial^2 x}\right)_i = \left[\frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\right)\right]_i &= \lim_{\Delta x \to 0} \frac{\left(\frac{\partial}{\partial x}\right)_{i+1/2} - \left(\frac{\partial}{\partial x}\right)_{i-1/2}}{\Delta x} \\ &\approx \frac{\frac{u_{i+1}-u_i}{\Delta x} - \frac{u_i - u_{i-1}}{\Delta x}}{\Delta x} = \frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2} \end{aligned} \tag{2.11}$$

In a discrete setting where the x difference is constant and set to 1 (i.e. $\Delta x = 1$) the second difference approximation can be written as follows:

$$\Delta^2 u_i = u_{i+1} - 2u_i + u_{i-1}, \quad i = 0, 1, \ldots, N \tag{2.12}$$

One particular interesting application for the second difference is the ability to detect the curvature of discrete data. For example, suppose the task is to investigate where a distribution of sorted, increasing numbers goes from a near linear to a more extreme type of growth (i.e. quadratic or

exponential). Calculation of the second difference at each discrete point can potentially identify where the distribution is approximately linear ($\Delta^2 u \approx 0$) and where there are major shifts in the positive direction ($\Delta^2 u > 0$):

Consider a list of length N with sorted and increasing values y. Let y_i be the value at position i, $1 < i < N$, then the corresponding second difference $\Delta_i^2 y$ can be calculated as follows:

$$\Delta^2 y_i = y_{i+1} - 2y_i + y_{i-1}, \quad 1 < i < N, \quad i \in \mathbb{N} \tag{2.13}$$

At points where the distribution is linear, $\Delta^2 y$ will be approximately zero. At points where $\Delta^2 y$ is positive, the distribution will have a convex growth. If $\Delta^2 y$ is positive and increasing, the growth rate is exponential. To identify the first critical point where the distribution changes from linear to a quadratic growth, choose a threshold δ, $\delta > 0$, iterate the data and calculate the second difference for each point. Then let the first value y_k where $\Delta^2 y_k > \delta$ be the first critical point in the distribution of where to expect the data to grow quadratic (convex growth). The use of second difference to determine the growth rate of discrete data is best illustrated by an example:

Example

First we redefine equation 2.13 so that $y_i = L(x)$, $x = 1, 2 \ldots, N$, and the second difference $\Delta^2 y = S(L(x))$. Hence, the second difference for $L(x)$ is defined as

$$S(L(x)) = L(x+1) - 2L(x) + L(x-1) \tag{2.14}$$

Then, consider the following list of sorted numbers:

$$L(x) = [1, 2, 3, 4, 5, 25, 36, 49, 64, 81, 100], \quad x = 1, 2, \ldots, 11$$

The corresponding central second differences can be calculated by using equation 2.14 and yields the values

$$S\left(L(x)\right) = [0, 0, 0, 19, -9, 2, 2, 2, 2], \quad x = 2, 3, \ldots, 10$$

Notice that the first and last number in L does not have a corresponding number in S. For $x = 1, 2 \ldots 5$, L is clearly linear (i.e. $L(x) = x$). The evidence for this can be found in $S(x)$ which is 0 for $x = 2, 3, 4$. However, for $x = 5$, $S\left(L(5)\right) = 25 - 2 \cdot 5 + 4 = 19$ suggesting that there is a substantial change in the growth rate from that point to the next. Furthermore, for $x = 6$, $S(x)$ is negative despite $L(x)$ is increasing indicating a decrease in the inflation. For $x = 7, \ldots 10$, $L(x) = x^2$ and

thus have a positive and constant corresponding value in $S(x) = 2$ (i.e. $L''(x) = 2$) suggesting that the inflation in this region is quadratic. To identify where L is changing from a linear to non-linear growth rate, we can look at the first point in the list where $S > 0$. In this example this corresponds to $S(L(5)) = 19$, indicating an increased growth rate from that point to the next. Figure 2.5 graphically illustrates this example and it is easy to see that there is a change in the growth rate between the data point $L(5)$ and $L(6)$. However, it should be noted that this method only identifies change in growth rate and do not guarantee that the growth continues in the same fashion.

2.4.2 The use of second difference in applications

A widely cited application that employs second difference is a method developed by Evanno, Regnaut and Goudet 2005 (cited over 6000 times as of April 2015 according to Google scholar). The method uses second difference to detect the true number of population clusters identified by the program STRUCTURE (Pritchard, Stephens and Donnelly 2000, Falush, Stephens and Pritchard 2003). STRUCTURE estimates the posterior probability of the data X, given each cluster K ($Pr(X|K)$ using MCMC. For each MCMC step STRUCTURE calculates the log likelihood of the data given a specific K. The output, $LnP(D) = L(K)$, is calculated for each K by averaging all these values and then subtracting half of the variance from the this mean (see Pritchard, Stephens and Donnelly 2000 for details). The true numbers of clusters is often interpreted as the peak of the $L(K)$ distribution. However, The authors of the method found that the true number of clusters K where more likely to be found at the point where this distribution plateaued or continued to increase slightly. This critical point can be determined by looking at the absolute value of second difference $|L(K)''| = |L(K+1) - 2L(K) + L(K-1)|$ and identify the peak in this distribution. To account for uncertainty in the MCMC algorithm in STRUCTURE, $|L(K)''|$ is averaged over multiple runs of STRUCTURE and ΔK is calculated using both this mean and the standard deviation of $L(K)$ (i.e. $\Delta K = mean(|L(K)''|)/sd(L(K))$) (see Evanno, Regnaut and Goudet 2005 for details). Figure 2.6 shows the population structure of African Sorghum (Westengen et al. 2014a) inferred by STRUCTURE evaluated by the Evanno method and visualized by the *Structure Harvester* website (Earl and vonHoldt 2012).

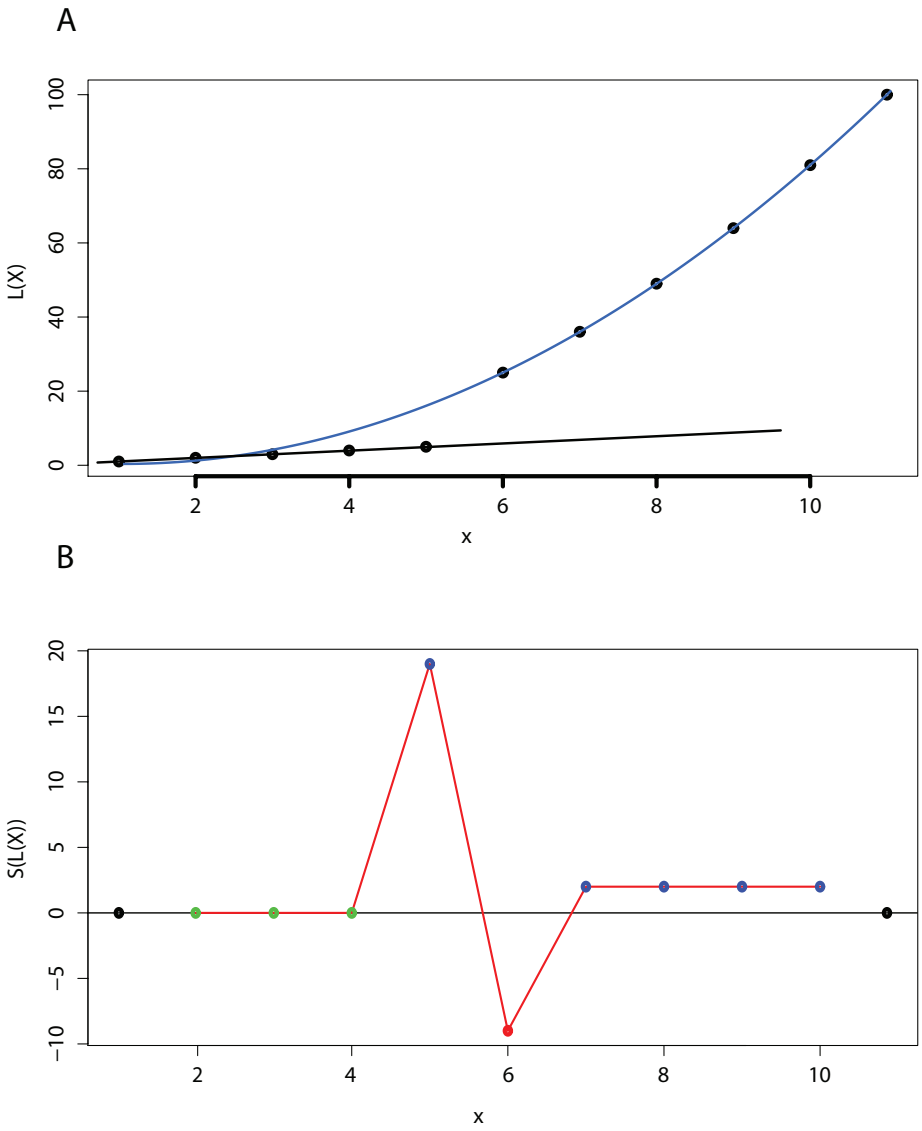

Figure 2.5: Plots visualising the second difference of a discrete function. A) The black and blue lines indicates the linear and quadratic nature of the points $x = 1, \ldots, 5$ and $x = 6, \ldots, 11$ in $L(x)$ respectively. B) The corresponding second difference value $S(L(x))$. The green, blue and red dots corresponds to a linear, concave and convex growth in $L(x)$ respectively.

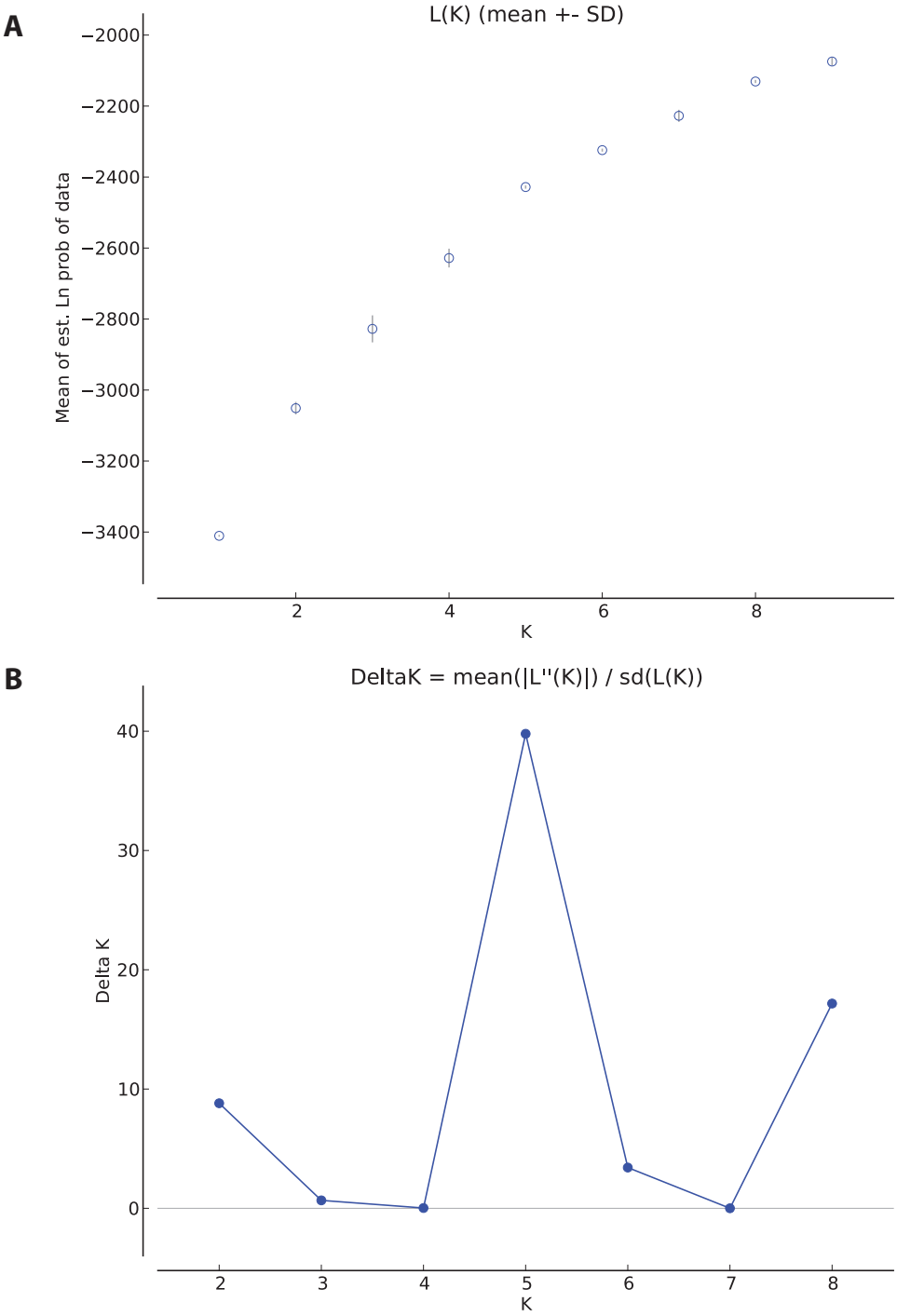

Figure 2.6: Population structure of African Sorghum inferred by multiple runs of STRUCTURE evaluated by the Evanno method and visualized by *Structure Harvester* (Westengen et al. 2014a). A) Mean Ln probability and standard deviation for $K = 1 - 9$ and B) the corresponding ΔK-values showing a peak at $K = 5$, thus five is the most probable number of population clusters. Source: Westengen et al. 2014a.

2.5 Empirical p-value

A statistical method known as "empirical p-value" ($\hat{p}$) is often used as a rank statistics when the results are given in terms of Bayes factors (BFs) (eg. Blair, Granka and Feldman 2014; Hancock et al. 2011b). This method yields a measure of how likely a result is compared to other results in the analysis. The BFs are first ranked according to their value, where the higher BF corresponds to lower ranks. Second, the empirical distribution are normalized to have values between 0 and 1, with 0 and 1 corresponding to the highest and lowest Bayes factor respectively. The most common equation is $\hat{p} = r/n$, where n is the total number of results and r is the number of tests that have produced a test statistic greater than or equal to the result calculated for the actual data. However, it has been shown (North, Curtis and Sham 2002) that this equation is anti-conservative and the correct equation is given as

$$\hat{p} = \frac{r+1}{n+1} \tag{2.15}$$

By using this equation the true p-value associated with $r_k = 5$ and $n = 500$ is .012 compared to .01 using the r/n equation. Thus the latter may yield 20% more significant results at a level of .01. However, it should be noted that in practice the choice of equation is negligible, in particular if r is large (North, Curtis and Sham 2002).

In principle, the empirical p-value is nothing else than a ranking statistics transforming the data to a format similar to p-values (e.g. lower value, higher significance) . For example an $\alpha_{\hat{p}} \leq 0.05$ cutoff is the same as finding the 95 percentile of an empirical distribution of the results from multiple hypothesis testing.

2.6 Determine a cutoff threshold for multiple hypothesis testing

By using traditional p-value cutoffs like α =0.5 or 0.01 when performing tests on thousands of features (e.g. SNPs), there is a chance of having an abundance of false positive results (Storey and Tibshirani 2003). Therefore it is necessary to find a cutoff threshold that best calibrates the balance between the number of true and false positives. Several techniques have been suggested, most of them intended for frequentistic hypothesis testing. One of the simplest methods for controlling the family-wise error rate (FWER; the probability of making one or more false discoveries when performing multiple hypothesis testing) is the Bonferroni correction. This method guarantee that the probability of having one or more false positives

is less than or equal to α by calling all p-values $\leq \alpha/m$ (where m is the number of features tested) significant. However, controlling the FWER using Bonferroni correction is considered to be conservative when performing genome-wide studies (Storey and Tibshirani 2003).

Another widely used and less conservative method is the q-value (Storey and Tibshirani 2003). The q-value is a transform of the p-value and is an extension of the False Discovery Rate (FDR; the proportion of false positives among all positives, see Benjamini and Hochberg 1995). While the p-value is a measure of significance in terms of false positive rate (FPR; the number of false positives divided by the sum of false positives and true negatives), the q-value is a significant measure in terms of FDR. For example, applying a traditional p-value cutoff $\alpha_p \leq 0.05$ means that on average 5% of the truly null features are called significant, whereas a q-value cutoff $\alpha_q \leq 0.05$ means that there is on average 5% truly null features among the features called significant. One particular advantage with the q-value is that it also can be applied to results given on the form of BF. The algorithm calculates the q-value by transforming the BF to the posterior probability of the alternative hypothesis (see Muller, Parmigiani and Rice 2006). The q-value has been successfully applied in several studies employing Bayesian statistics (e.g. Foll and Gaggiotti 2008) and provides a good measure for comparing BF and traditional p-values in the context of multiple hypothesis testing (e.g. Villemereuil et al. 2014).

2.6.1 Interpretation of the q-value and the difference to p-value

Consider an experiment with 1000 tested features. A feature with a p-value of 0.01 implies that there is a 1% risk of that feature being a false positive. Choosing a p-value cutoff threshold of $\alpha_p = 0.01$ implies that there are $1000 \cdot 0.01 = 10$ false positives among features with p-value ≤ 0.01. If this feature is ranked as number 50 in the distribution, this implies that there are on average 10 false positives among the 50 top ranking features. A q-value cutoff threshold of $\alpha_q = 0.01$ on the other hand is much more conservative. If there are 50 features among the 1000 tests that have a q-value≤ 0.01, there is on average $50 \cdot 0.01 = 0.5$ false positives among the 50 top ranking features. If a feature has a q-value of $\hat{\alpha}_q$ it can be interpreted as the proportion of false positives among tested features with a q-value as large or less than $\hat{\alpha}_q$.

It should be noted that the q-value is only defined in the context of multiple hypothesis testing, whereas the p-value (and BF) can be applied to single tests.

2.6.2 BF to q-value conversion algorithm

As mentioned above, the conversion algorithm by Muller, Parmigiani and Rice 2006 uses the posterior probability of the alternative model M_1 when calculating the q-value. By recalling Equation 2.5, we can write the posterior odds of the two models given the data y as:

$$\frac{P(M_1|y)}{P(M_0|y)} = BF\frac{P(M_1)}{P(M_0)} \tag{2.16}$$

By using the fact that the prior probability $P(M_0) = 1 - P(M_1)$ and the posterior probability $P(M_0|y) = 1 - P(M_1|y)$, we can re-inject this into equation 2.16 and have that the posterior probability of M_1 is given as:

$$P(M_1|y) = \frac{1}{1 + \frac{1-P(M_1)}{BF \cdot P(M_1)}} \tag{2.17}$$

By assigning a prior probability for $P(M_1)$, $0 < P(M_1) < 1$, and writing the posterior probability for M_1 as PP, we can use the following pseudo code (algorithm 1) to calculate the q-values for all features (e.g. SNPs) in the dataset:

Algorithm 1 Algorithm for transforming BFs to q-values

```
for each feature i do
  q-value ← 0
  number-of-significant ← 0
  for each feature j do
    if PP(j) is greater than PP(i) then
      q-value ← q-value + (1 − PP(j))
      number-of-significant++
    end if
  end for
  q-value ← q-value/number-of-significant
end for
```

Chapter 3

Review of BAYENV

In this chapter we will give an introduction to the BAYENV model (Coop et al. 2010; Günther and Coop 2013) and how the program works. As already mentioned, there are several challenges with the method. First, interpreting the BAYENV results is challenging because it is provided in terms of BFs which are not as well known as the frequentistic counterpart the p-value. Second, the run-to-run variability of the program is also causing complications for the user (Blair, Granka and Feldman 2014). To illustrate these challenges, we will provide some examples from the literature where BAYENV has been used. Furthermore, in a recent simulation study, the BAYENV showed very good performance when it was compared to seven other methods for detecting selection along environmental gradients (De Mita et al. 2013). However, the performance gain is coming with a considerable cost in terms of computing time (De Mita et al. 2013). Given this, we will discuss the time consumption of BAYENV .

3.1 Introduction to BAYENV

Coop et al. 2010 introduces a Bayesian method that uses environmental correlations to identify loci underlying local adaptation. The model is embodied in the UNIX based software BAYENV which is written in the C language. The binary of the program is distributed freely over the internet. The basic idea is to identify loci where environmental variables of interest have a linear effect on the allele frequencies across populations. By using environmental variables such as water salinity, precipitation or temperature, it may be possible to discover genotypes that are correlated with population's adaptation to divergent conditions. The BAYENV model uses Bayesian inference and MCMC algorithms to infer relationship between allele frequencies and environmental variables.

The method is carried out in a two-step procedure. First, a large set of presumably neutral loci is used to estimate a covariance matrix that

represents the empirical pattern of covariance in allele frequencies between populations due to shared population history and gene flow. MCMC (see Section 2.3 is used to explore the posterior of the covariance matrix and the model parameters. Second, the allele frequencies at each SNP are then tested for correlation to one or more environmental variables, using the covariance matrix from the first step as a null model. A MCMC scheme is use to explore the posteriori of the alternative model and a Bayes factor (BF) is calculated as a measure of support for either of two models. A high Bayes factor indicates a linear correlation between the population allele frequencies and the environmental variable and thus the locus (SNP) may potentially be the target of selection and local adaptation due to diverging environmental conditions. Coop et al. 2010 tested BAYENV and found that the program had higher statistical power than four other methods to detect correlation between environmental variables and allele frequencies. Furthermore, the simulation study performed by De Mita et al. 2013 confirmed this conclusion by showing that the BAYENV method had good statistical power compared to seven other approaches.

In the following sections we describe the statistical models and the implementation of BAYENV.

3.2 The BAYENV model

In the next section we give a brief summary of the BAYENV models and on what basis the Bayes factor (BF) is calculated. For a thorough review of the models, we refer to the article (Coop et al. 2010). The assumption of the model are that the populations are reasonably large and that the loci are in Hardy-Weinberg equilibrium (see Section 2.1.16).

3.2.1 The null model

To account for the shared history and gene flow between populations, a covariance matrix is estimated based on the empirical pattern of covariance in allele frequencies in a large set of putatively neutral control SNPs. The covariance matrix is a measure of how allele frequencies naturally co-vary across populations and is used as the null model in the method. The SNPs that forms the basis for the null model should be a large representative sample from the total population.

In the null model it is assumed that the population frequencies have a multivariate normal distribution. Moreover, the model assumes L unlinked SNPs in K populations. Then let $\mathbf{n}_l = \{n_{1l}, \cdots, n_{Kl}\}$ and $\mathbf{m}_l = \{m_{1l}, \cdots, m_{Kl}\}$ be the observed count of allele 1 and 2 respectively. Furthermore, the allele frequencies for locus l in population k is denoted x_{kl}. The model assumes that the observed counts of each allele (n_{kl}, m_{kl})

are the result of binomial draws from these frequencies. Moreover, it is assumed that the allele frequency in a subpopulation, x_{kl}, is normally distributed around an ancestral allele frequency ϵ $(0 < \epsilon < 1)$. The variance of this normal distribution is given as a factor that is constant across loci multiplied by a locus-specific term, $\epsilon_l(1-\epsilon_l)$. Thus if the allele frequency in the current generation is ϵ, the allele frequency in the next generation is approximated as $\sim \mathcal{N}(\epsilon_l, (1-\epsilon_l)/2N_e)$, where N_e is the effective population size. These assumptions were adopted from a pure drift model suggested by Nicholson et al. 2002.

As opposed to the normal distribution, the population allele frequency x_{kl} is constrained to be between 1 and 0. This problem is solved by having a transform function from a normally distributed surrogate allele frequency θ_{kl} to the population allele frequency x_{kl}. The transform function g is given as:

$$x_{kl} = g(\theta_{kl}) = \begin{cases} 0 & \text{if } \theta_{kl} < 0 \\ \theta_{kl} & \text{if } 0 \leq \theta_{kl} \leq 1 \\ 1 & \text{if } \theta_{kl} > 1 \end{cases} \tag{3.1}$$

The densities $\theta_{kl} \geq 1$ and $\theta_{kl} \leq 0$ represents the probability that the allele has been fixed or lost in the population respectively. For each locus l, θ_{kl} has a marginal distribution $\sim \mathcal{N}(\epsilon_l, \epsilon_l(1-\epsilon_l)C_k)$, where $C_k = \tau/(2N_e)$ is a constant specified after τ generations of genetic drift.

To explicitly estimate the variance-covariance of allele frequencies across populations it is assumed that the surrogate allele frequency θ_l has a multivariate normal distribution:

$$P(\theta_l|\Omega, \epsilon) \sim \text{MVN}(\epsilon_l, \epsilon_l(1-\epsilon_l)\Omega) \tag{3.2}$$

where Ω is variance-covariance matrix of allele frequencies between populations. The joint posterior of the parameters $(\theta_l, \Omega, \epsilon_l)$ is written as

$$P(\theta_l\Omega, \epsilon_l, |\mathbf{n}_l, \mathbf{m}_l) \propto P(\mathbf{n}_l, \mathbf{m}_l|\mathbf{x}_l = g(\theta_l))P(\theta_l|\Omega, \epsilon_l)P(\Omega)P(\epsilon_l) \tag{3.3}$$

As prior for the variance-covariance matrix Ω, $P(\Omega)$, inverse Wishart is chosen due to its property of being the conjugate prior of a multivariate normal distribution. The joint posterior for all loci L is given as

$$\begin{aligned} &P(\Omega, \theta_1, \ldots, \theta_L, \epsilon_l, \ldots \epsilon_L|\mathbf{n}_1, \mathbf{m}_1, \ldots, \mathbf{n}_L, \mathbf{m}_L) \propto \\ &\left\{ \prod_{l=1}^{l=L} P(\mathbf{n}_l, \mathbf{m}_l|\mathbf{x}_l = g(\theta_l))P(\theta_l|\Omega, \epsilon_l)P(\Omega)P(\epsilon_l) \right\} P(\Omega) \end{aligned} \tag{3.4}$$

3.2.2 The alternative model

In the alternative model, a SNP of interest is examined for correlation to a standardized environmental variable Y. In the model, the allele frequency θ_l is allowed to be linearly dependent on Y. Moreover, θ_l has a deviation from ϵ_l that is linearly proportional to Y with coefficient β. The linear relationship between θ and Y can be predicted by the following model:

$$P(\theta_l|\Omega, \epsilon_l\beta) \sim \text{MVN}\left(\epsilon_l + \beta Y, \epsilon_l(1-\epsilon_l)\Omega\right) \tag{3.5}$$

However, due to the boundaries at 0 and 1 the same linear relationship does not necessarily apply between the population frequency x_l and the environmental variable Y.

By defining a prior uniformly distributed $P(\beta)$ on β, the joint posterior can be estimated by

$$P(\theta_l\Omega, \epsilon_l, \beta|\mathbf{n}_l, \mathbf{m}_l) \propto P(\mathbf{n}_l, \mathbf{m}_l|\mathbf{x}_l = g(\theta_l))P(\theta_l|\Omega, \epsilon_l, \beta)P(\Omega)P(\epsilon_l)P(\beta) \tag{3.6}$$

As prior for the covariance matrix for a single locus, the posterior of the covariance matrix estimated from the neutral control SNPs is being used. The prior for the coefficient β is a uniform distribution between β_{min} and β_{max}.

3.2.3 Calculation of Bayes factor

To provide a measure of how likely it is that a SNP is correlated to the specified environmental variable, a Bayes factor is calculated based on the posterior of odds of the two models. If M_1 is the alternative model ($\beta_{min} \leq \beta \leq \beta_{max}$), and M_0 is the null model ($\beta = 0$), then the posterior probabilities $P(M_0|\mathbf{n}_l, \mathbf{m}_l)$ and $P(M_1|\mathbf{n}_l, \mathbf{m}_l)$ can be found by integrating the right-hand side of the equations *1.7* and *1.8* respectively. Thus the Bayes factor for the locus l is given by

$$\frac{P(M_1|\mathbf{n}_l, \mathbf{m}_l)}{P(M_0|\mathbf{n}_l, \mathbf{m}_l)} = \frac{\int P(\mathbf{n}_l, \mathbf{m}_l|\theta_l)P(\theta_l|\beta, \epsilon_l, \Omega)P(\Omega)P(\epsilon_l)P(\beta)d\beta\theta_l d\epsilon_l d\Omega}{\int P(\mathbf{n}_l, \mathbf{m}_l|\theta_l)P(\theta_l|\epsilon_l, \Omega)P(\Omega)P(\epsilon_l)d\theta_l d\epsilon_l d\Omega} \tag{3.7}$$

3.2.4 The use of MCMC

Due to its high complexity, BAYENV uses MCMC to explore the posterior of the two models. On each MCMC iteration the parameters are sequentially updated using a Metropolis updating scheme. A Gibbs sampling scheme is used to update the covariance matrix by sampling from the posterior distribution of the null model. The Bayes factor is calculated using a single run of the MCMC algorithm under the null distribution and by using a technique known as importance sampling. Interested readers are referred to the article appendix A and B (Coop et al. 2010) for further details about the MCMC implementation in BAYENV.

3.3 How the BAYENV analysis is performed

The BAYENV program is run as a two step procedure. First, a covariance matrix is estimated based on a large number of presumed neutral SNPs. The user must specify a random seed for the MCMC algorithm and for how many iterations the program should run. A covariance matrix is drawn from the posterior distribution every 500 MCMC cycle and written to file. One of these covariance matrices serves as a input parameter for the next step in the procedure. Including SNPs in the null model that later should be the subject of testing is unproblematic as long as the number of SNPs used is large (Coop et al. 2010). BAYENV supports a maximum of 10,000 SNPs as basis for the covariance matrix.

Second, in the test phase, each SNP (provided to the program as a single locus frequency file) of interest is tested individually for correlation to one or more environmental variables using the previously estimated covariance matrix as a null model to correct for the neutral allele frequency variation across populations. Similarly to the first step of the procedure, the user must specify the random seed and the number of MCMC iterations for each test SNP. Additionally, the user must provide a file with the environmental variables that are standardised across populations (i.e. subtract the mean and divide by the standard deviation) as an input parameter to the test phase of the program. For each SNP tested, BAYENV writes the file name of the single SNP-file and the resulting BFs calculated for each environmental variable to disk. If more than one SNP is tested, the program appends the results to the already existing result-file (program default). In order to simplify the analysis of the result file it is convenient to name the SNP file in accordance with the name of the SNP.

The number of MCMC iterations needed for the two steps in the procedure is a debated subject (Blair, Granka and Feldman 2014). Coop et al. 2010 found that the MCMC algorithm estimating the covariance matrix converged relative quickly and stated that 5000 iteration was sufficient as a burn-in. However, the authors do not state how many iterations

that was used when testing SNPs in the article, nor do they provide any recommendations on the subject. By looking at articles where BAYENV has been used, the number of MCMC iterations used ranges from a couple of thousand to 1,000,000.

3.3.1 The BAYENV file format

The BAYENV file format is program specific, in accordance with most other bio-analysis programs. The SNPs are organized row-wise, where each SNP occupies two rows; one for each allele. Individuals are grouped in populations and organized column-wise in the matrix.

Formally the BAYENV file format is organized in the following way: By having L SNPs genotyped in K populations, the row vector $\mathbf{n}_l = \{n_{1l} \dots n_{Kl}\}$ is the observed counts for allele 1 at locus l and $\mathbf{m}_l = \{m_{1l} \dots n_{Kl}\}$ is the observed counts for allele 2 at locus l (table 3.1).

$Locus$	Pop_1	Pop_2	$\cdots$	Pop_K
$\mathbf{n}_1$	n_{11}	n_{21}	$\cdots$	n_{K1}
$\mathbf{m}_1$	m_{11}	m_{21}	$\cdots$	m_{K1}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{n}_L$	n_{1L}	n_{2L}	$\cdots$	n_{KL}
$\mathbf{m}_L$	m_{1L}	m_{2L}	$\cdots$	m_{KL}

Table 3.1: Schematic overview the BAYENV file format. The row vectors $\mathbf{n}_j$ and $\mathbf{m}_j$ are the observed counts for allele 1 and 2 respectively in population j. The first row and column are headers and not included in the file.

Table 3.2 shows an example of the BAYENV format taken from the first five SNPs from a dataset containing SNPs from Atlantic cod (*Gadus Morhua*) genotyped by Berg et al. 2015, *in review*.

3.4 Evaluation of the BAYENV method by De Mita et al.

In 2013, De Mita et al. (De Mita et al. 2013) performed a study where the power and robustness of eight different methods to detect selection along environmental gradients were benchmarked. Three of the methods were based on environmental correlation, whereas five methods were differentiation (F_{ST}) based. In this study, 100 populations were simulated along a selective gradient using the software *QUANTINEMO*

SNP	Pop1	Pop2	Pop3	Pop4
SNP_1 $allele_1$	6	0	7	2
SNP_1 $allele_2$	90	112	77	94
SNP_2 $allele_1$	96	112	83	94
SNP_2 $allele_2$	0	0	1	2
SNP_3 $allele_1$	0	1	2	1
SNP_3 $allele_2$	96	111	80	95
SNP_4 $allele_1$	8	7	7	5
SNP_4 $allele_2$	88	105	77	91
SNP_5 $allele_1$	32	56	30	34
SNP_5 $allele_2$	64	56	54	62

Table 3.2: Example of the BAYENV file format taken from a data set containing SNPs from Atlantic cod (Berg et al. 2015, *in review*). The first row and first column is for illustration purpose only and are not included in the actual file.

(Neuenschwander et al. 2008). After simulation the data sets contained 1000 neutral loci and 100 loci under selection. Several different samplings schemes, migration models and rates of self-fertilization were explored (see De Mita et al. 2013 for details).

Among the correlation-based methods was BAYENV by Coop et al. 2010. Several BAYENV analyses were carried out using loci from different sampling schemes. The loci were tested for correlation along three artificial environmental gradients. Only neutral loci were used as a basis for the covariance matrix. The study concluded that correlation based methods were more powerful than differentiation based methods, especially with the lowest level of simulated selection. In particular, the BAYENV method was found to have the best power when using discriminant selection intensities. However, as noted by the authors, the performance gained by BAYENV comes with a significant cost in computing time.

3.5 BAYENV 2.0

In September 2013 BAYENV 2.0 was released in conjunction with the article *Robust identification of Local Adaptation from allele frequencies* (Günther and Coop 2013). As the title indicates, the authors argue that there is a need for a more robust method for analysing allele frequencies and loci underlying local adaptation. The main problem with linear models such as BAYENV is that they are less robust with regard to outlier results. To correct for this, the authors extend BAYENV 2.0 with a method for "standardised allele frequencies" which can be used to conduct additional tests of the user's own choice. To show how the standardized frequencies can be used, a rank-based non-parametric statistics to detect correlation with environmental variables is implemented in the program. As an addition

to the Bayesian test for linear correlation (the BF), BAYENV 2.0 is now able to calculate Pearson's and Spearman's correlation between a standardized allele frequency and an environmental variable (transformed to be in the same frame of reference). These tests are not intended as a replacement for the Bayes factor, but rather as a supplement. Having both the Bayes factor and the Spearman's ρ for a loci and an environmental variable, enables the analysts to be more confident in their conclusions.

The original BAYENV method is unchanged except for more frequent output of the covariance estimate (output every 500 instead of every 5,000 MCMC cycle). We will in the remaining chapters be using BAYENV 2.0 but refer to it as the BAYENV method.

3.6 Challenges using BAYENV

There are several challenges regarding both the use of BAYENV and the interpretation of the results. First of all, the multi-step procedures of the program are not very user friendly and may pose challenges to non-technical researchers. In addition to a particular file format, the program supports testing of only one SNP at the time. Thus, testing of more than a few SNPs requires that the process is automated in some way. This could be challenging if the user does not possess programming skills.

Second, BAYENV employs MCMC simulations both for estimating the null model and for the tests at each locus. The number of iterations needed in order to have convergence is not known in advance. The covariance matrix and the BFs are computed using single draws from the posterior distributions after a certain number of user specified MCMC iterations. This implies that even though the MCMC algorithm has reach its "equilibrium space", every draw using the MCMC sampler is likely to be different. The random seed may also have an impact on the result.

Changing the number of MCMC iteration for both steps in the procedure will most probably lead to different results in terms of BF for each SNP. For example, running a test using 10,000 MCMC iterations and a covariance matrix estimated after 20,000 MCMC iterations will likely provide a different result than a run using 20,000 and 10,000 MCMC iterations for the test and covariance matrix respectively. Changing the random seed will provide yet another result. By different results, we do not necessarily mean qualitatively different results. A high ranking SNP in one run will probably be high ranking in another, however, the signal strength and the order may vary quite a lot depending on the environmental variable. This variability makes it particularly important to check the results with multiple runs of the algorithm (Blair, Granka and Feldman 2014).

3.6.1 Assessment of the results

BAYENV provides the results in terms of BFs (see Section 2.2.4). Several different approaches exist to determine a significance level for BFs, where a percentage cutoff on the empirical distribution or thresholds set by Jefferey's scale of evidence table (see Table 2.2) are among the most popular. However, there are several disadvantages to using these approaches. For example, if there are only three true significant SNPs among 10,000 tested SNPs, a 95% cutoff will lead to 497 false positive results. Another problem may be that a percentage cutoff could potentially include results that actually supports the null model (BF<1) if the overall BF signal is low enough. Using one of the predefined thresholds from one of the significance tables (table 2.2 or 2.3) may also pose challenges. For example deciding on a threshold of BF>100 ("Decisive" according to Jefferey's table) may in some cases lead to a large number of positive results (e.g. if it was used on the results in Coop et al. 2010). For example if 15% of the SNPs shows a BF$>$ 1000, may a "decisive" BF of 100 then be regarded as significant? The problem of using a lookup table is that the choice of cutoff are easily being "customized" to the distribution according to the analyst's prior suppositions. In the subsequent chapters we will refer to cutoffs based on Jeffrey's (or Kass') table (Table 2.2 and Table 2.3) as a static cutoff.

3.6.2 Time consumption

As with most programs that employs MCMC simulation algorithms, BAYENV is quite time consuming. First, a covariance matrix needs to be calculated using at least 10,000 MCMC iteration steps. The time consumption for this, increase with the size of the data (i.e. the number of SNPs and populations). Second, the test for correlation to the environmental variable at each SNP needs to run for at least 10,000 MCMC iterations each. The time consumption for the test phase increases linearly with the number of SNPs tested.

Due to the uncertainty associated with convergence of MCMC algorithms , the BAYENV results should be checked with multiple runs of the program using different random seeds (Coop et al. 2010). To illustrate how time consuming the process can be, consider the following example:

These time estimates are based on the assumption that the BAYENV analysis is carried out sequentially on an up-to-date desktop computer (e.g. Intel i5 CPU). By using 10,000 control SNPs as a basis, a covariance matrix is estimated using 100,000 MCMC iterations. A rough estimate of the time spent creating this matrix is four hours. Next, 50,000 SNPs are tested ten times (using different random seeds) using 100,000 iterations each. If we assume an approximate estimate of five seconds spent at each test, we have the following total time consumption for a full BAYENV analysis:

1 covariance matrix $= 4$ hours

5 seconds $\cdot$ 50000 SNPs $\cdot$ 10 runs $= 2500000$ seconds $= 694.4$ hours

In total: covariance matrix $+$ tests $= 4 + 694.4 = 698.4$ hours $= 29$ days

It is easy to see that the most time consuming part of the analysis is the test at each SNP. Consequently a parallelization of this part could dramatically reduce the total time spent on the analysis. Parallelization and other methods to reduce the time consumption will be discussed in chapter four, five and six.

3.7 Use of BAYENV in research

Since the release in 2010, BAYENV Coop et al. 2010 has become a popular tool among scientists conducting research on population allele frequencies and adaptation to local environments (e.g. Chen et al. 2012; Fang et al. 2012; Fumagalli et al. 2011; Hancock et al. 2010a; Hancock et al. 2011b; Westengen et al. 2014b; Ye et al. 2013, etc). The article is cited 192 times as of April 2015 (according to Google Scholar). However, the guidelines for how BAYENV should be run an how to interpret the results are vague in both the article and the accompanying manual. This fact is reflected in the articles where the method is used.

In Hancock et al. Hancock et al. 2011b, the BAYENV method was applied to genome-wide SNP data containing 61 human populations to search for evidence of adaptation to 11 climatic variables. The study was successful and evidence of adaptation was found to several variables, latitude, summer relative humidity and summer solar radiation being among the most significant.

The BAYENV analysis performed by Hancock et al. 2011b was re-analysed by Blair, Granka and Feldman 2014. The result showed that BAYENV had a high variability between runs (for details see Blair, Granka and Feldman 2014). Among the more significant findings in the original article were three SNPs in the CORIN gene, rs4558846, rs6447571 and rs17601068, reported by Hancock et al. to have an association with minimum winter temperature. They all gave very strong signals, $\log_{10}$BF = 21.9, 28.7 and 20.8 respectively. However, when the same analysis was rerun by Blair et al., the same SNPs showed no significance with $\log_{10}$BF = -0.32, -0.50, and -0.48 respectively. Increasing the number of MCMC iterations from 100,000 to 500,000 did only increase the stability of the method marginally. Based on these findings, Blair, Granka and Feldman 2014 warned against making conclusion based on one single run of the program. The authors also recommend caution in interpreting previous studies that have used only one run.

3.7.1 Examples from the literature

There are three particular aspects with BAYENV that are causing difficulties for the analysts and leads to a host of different approaches. First, the number of iterations that are needed for the procedure in order get a stable and accurate result. Second, the number of independent runs needed to verify the results. Third, how to define a proper cutoff threshold for significance. To illustrate the problem of inconsistent and varying use of the method, we here provide a couple of examples from the literature.

Example 1
In the article *Imprints of Natural Selection Along Environmental Gradients in Phenology-Related Genes of Quercus petraea*, Alberto et al. 2013 use BAYENV to investigate correlation between population allele frequencies in *Quercus petraea* and latitudinal and longitudinal gradients. A total of 175 polymorphic SNPs were tested. Here the authors chose to perform 100 analyses for each variable to account for the run-to-run variability. However, the authors do not state the number of iterations per run nor how the covariance matrix was created. Only SNPs among top 10 in each run was considered each time. For each SNP the mean and variance among runs were calculated. Only SNPs which exhibited stable position in the top 10 BF values were considered candidates. Three SNPs showed a stable average BF above 2 and the highest reported BF was 5.02. For a summary of the BAYENV usage in this example see Table 3.3.

Comments on Example 1
Not stating the number of MCMC iterations used in both BAYENV steps is problematic since this may have an impact on the stability of the results (Blair, Granka and Feldman 2014). The authors make use of a high number independent runs (100). Whether this can compensate for fewer iterations is a subject for further research. Furthermore, the authors do not comment on the choice of cutoff threshold, however, according to Jeffery's interpretation a BF>2 is "Not worth more than a bare mention" (see Table 2.2). One question would be: is a BF>2 high compared to the other SNPs tested? We would argue that the full distribution of test results must be taken into account when deciding a threshold for significance.

Example 2
In the article *Genome-wide single-generation signatures of local selection in the panmictic European eel* (Pujolar et al. 2014), a total of 50,354 SNPs with a minor allele frequency >0.05 from European eel were analysed using BAYENV. The SNPs were tested for correlation to the environmental variables degrees north latitude, degrees east/west longitude and sea-surface temperature. Five independent test runs were carried out to ensure consistency, however, the number of iterations is not mentioned. Furthermore, a BF > 3 was chosen as a cutoff for significant correlation. A total of 87 candidate SNPs representing 74 unique loci were identified. For a summary of the BAYENV usage in this study, see Table 3.3.

Comments on example 2 As mentioned for example 1, the lack of information on the number of MCMC iterations used may be problematic as it may impact the stability of the method (Blair, Granka and Feldman 2014). The authors does not use any descriptive statistics to summarize the results from five runs but only to verify the results from one run. This might be okay, but why not use the average or median BF? The choice of cutoff threshold (BF > 3) is not justified and seems a bit random. Moreover, there is no information on how the BFs are distributed for the entire dataset. We would argue that the choice of cutoff threshold must be in accordance with the full distribution of results and to lesser extent be a "convenient" threshold that may reflect the analyst's prior suppositions.

Example 3
In the article *Complex Patterns of Local Adaptation in Teosinte* (Pyhäjärvi et al. 2013), the authors aim to describe the genetic basis of local adaptation in 21 populations of Teosint (the wild ancestor of Maize) by using total of 36,719 genotyped SNPs. The study used 76 environmental variables where the dimensionality of the data was reduced using PCA. The BAYENV analysis was carried out by first estimating three different covariance matrices based on three different random subsets, each contained 10,000 SNPs. The covariance matrices were estimated using 50,000 MCMC iterations. These were examined and found to have low pairwise difference (less than 10% at most). Five independent BAYENV runs were carried out using 1,000,000 iterations on each SNP to test for association with the environmental variables (the PC's that captured 95% of the variation). As a cutoff for significance, SNPs that showed an average BF across runs in the 99th percentile and were consistently in the 95th percentile of the empirical distribution of Bayes factors from each run, were considered candidates. A total of 1,598 SNPs were identified to be associated with one or more principle components. For a summary of the BAYENV usage in this example see Table 3.3.

Comments on example 3 The use of BAYENV in this article is very well documented. Verifying the convergence by comparing different covariance matrices is a reassuring measure. Perhaps it would have been even better to use an average of the covariance estimates as long as the difference was up to 10%? Using as much as 1,000,000 MCMC iterations provides credibility to the results. However, is this high number of iterations really necessary to ensure convergence? The time consumption of BAYENV increases linearly with the number of MCMC iterations and thus is one of the drawbacks of the BAYENV algorithm (De Mita et al. 2013). The authors make use of a percentage cutoff to decide significance. As mentioned in Section 3.6 this method does not account for the actual BF signals in the distribution of results and may potentially lead to many false positive results. Moreover, the use of average BF as summary statistics for only five runs may be affected by extreme outlier results (i.e. the differences found by Blair, Granka and Feldman 2014, Section 3.7). Maybe the median statistics would have been a more robust way of summarising the results from five runs?

	Example 1	Example 2	Example 3
Number SNPs for the covariance matrix	175	N/A	10,000
Number of iterations for the covariance matrix	N/A	N/A	50,000
Number of SNPs tested	175	50,354	36,719
Number of iterations for the test phase	N/A	N/A	1,000,000
Number of independent BAYENV runs	100	5	5
Statistics for multiple runs	Average BF	N/A	Average BF
Cutoff for significance	BF > 2	BF > 3	99 percentile
Cutoff for variability	Top 10	N/A	95 percentile
Results available online	No	No	No

Table 3.3: Summary statistics for the BAYENV usage in Example 1, 2 and 3. N/A = no information available.

3.7.2 Comments to all examples

The examples in Section 3.7.1 illustrates the prevailing uncertainties about how BAYENV should be run and how to interpret the results. We do not suggest that any of these methods are necessarily wrong, however, the lack of a uniform method for how to carry out the analysis is evident. We know that the run-to-run variability may be high and comparing independent runs is necessary to detect outlier results. Furthermore, the number of MCMC iterations used for both steps in the BAYENV procedure is likely to affect the stability of the end result (Blair, Granka and Feldman 2014). Considering the fact that the MCMC algorithms is quite time consuming it is crucial to find a set of settings for BAYENV that balance the number of MCMC iterations and the number of independent runs with stability and time usage.

The absence of a common strategy for how to define a significant level for the distribution of BFs is also apparent. As pointed out by Coop et al. 2010, an empirical approach (i.e. the "empirical p-value", see section 2.5) has some serious drawbacks, with deciding what cutoff to use being the most obvious as the choice of cutoff often reflects one's prior beliefs of selection. The use of a static cutoff such as Jeffrey's scale of evidence for BF (see Table 2.2) is also unsatisfactory (see Section 3.6.1).

It is crucial to be aware of these facts when using BAYENV to assay the genome-wide pattern of local adaptation. There are three main questions that need further discussion. First, how can we ensure consistent results from a BAYENV analysis? Second, how can we assign a proper significant level for the results? Third, how can we reduce the overall time consumption of BAYENV?

Chapter 4

Methods and materials

In this chapter we first present the datasets and materials used in this thesis. Second, we describe a method named the Second Difference Method (SDM) that uses the properties of second difference to assign a dynamic cutoff level to an empirical distribution of BF results. The method is intended to act as an alternative to conventional cutoff methods such as a percentage or static (table) cutoffs in the context of multiple hypothesis testing. Third, considering the high run-to-run variability observed, we define how the SDM can be used to interpret the results across multiple runs of BAYENV. Fourth, as a time saving measure, we propose a method of reducing the BAYENV test set by excluding SNPs with low maximum allele frequency difference (MAFD) between populations. Fifth, we describe the methods used for testing the stability of BAYENV. Finally, we describe the functional specifications for PYBAYENV, a BAYENV wrapper, which we used to carry out the experiments required to test our hypotheses and to reduce the overall time consumption of BAYENV.

4.1 Materials

In this section we describe the two SNP datasets, populations and the associated environmental variables we used to demonstrate our methods.

4.1.1 The cod dataset

To test our hypotheses and the overall capability of BAYENV we used a dataset consisting of 8809 polymorphic SNPs from the genome of Scandinavian Cod, most of them genotyped by Berg et al. 2015, *in review*. Of the 8809 SNPs, 262 had previously been published (Hemmer-Hansen et al. 2011; Hubert et al. 2010; Moen et al. 2008), 648 were selected from candidate genes, 1554 were non-synonymous coding (see Section 2.1.12),

whereas the remaining 6345 SNPs were randomly distributed throughout 23 linkage groups (a linkage group is a set of two or more loci that have been shown to be physically close but have not yet been assigned to specific chromosome) on the cod genome. The two latter groups are considered to be evolutionarily neutral SNPs. The SNP dataset was made available in GENEPOP format (Raymond and Rousset 1995) where the populations were defined. We will refer to this as the *Cod* dataset.

Populations

The basis for the 8809 SNPs were 194 individuals of adult Atlantic cod collected from 7 locations outside the coast of Scandinavia. The sampling places were: eastern North Sea, Kattegat, Öresund area, the Bornholm basin (2 collections), the Öland area and the Gotland area. Four population clusters were defined based on results from STRUCTURE (Falush, Stephens and Pritchard 2003; Pritchard, Stephens and Donnelly 2000) and used in the subsequent BAYENV analyses. The Baltic population is a merging of the individuals from the four collections from the Baltic Sea.

Environmental variables

Six different environmental variables were available for BAYENV analysis of the Cod dataset: Water salinity (psu), Temperature (°C) and Oxygen level - all respectively on surface (5-10 meters) and at spawning depth. In the case of the Baltic population where individuals from four different sampling sites were grouped, the environmental variables were averaged across these collections sites. We will refer to these variables as follows: salinity at surface as *sal1*, salinity at spawning depth as *sal2*, temperature at surface as *temp1*, temperature at spawning depth as *temp2*, oxygen at surface as *ox1* and oxygen at spawning depth as *ox2*. All environmental variables were obtained from the study by Berg et al. 2015, *in review*.

4.1.2 Maize dataset

In addition to the Cod dataset, we used a relatively small dataset containing 135 polymorphic SNPs from African Maize used in a study by Westengen et al. 2014b. These SNPs are a subset of a panel of African maize containing 43963 SNPs (Westengen et al. 2012), done with the MaizeSNP50 array (Ganal et al. 2011). Out of the 135 chosen SNPs nine were candidate SNPs earlier suggested to be associated with maximum temperature during the growing season, 35 SNPs suggested to be under positive selection based on F_{ST} values and the remaining 100 SNPs were randomly selected from the SNP array. The random SNPs were evenly distributed throughout the ten maize chromosomes. Furthermore, 109 of the SNPs are located in known

or putative genes identified by marker search databases (www.panzea.org and gramene.org). The reader is referred to the article for further details. The dataset was available in GENEPOP format. We refer to these SNPs as the *Maize* dataset.

Populations

Three "populations" were defined based on samples from three different stages in the local seeds systems of Mangae, a village in the Morogoro district in Tanzania. The three populations were: 1) A formal seed population consisting of seeds from the formal sector. 2) Seeds reportedly used for one generation (one year). 3) Seeds recycled over 10 years. Seeds from all populations were of, or were originating from the improved maize variety *Staha*.

Environmental variable

An ordinal environmental variable based on the stages in the local seed system was defined in a similar manner as Heerwaarden, Hufford and Ross-Ibarra 2012 used breeding era. The three stages were "translated" into quantitative variables such that "original"=1, "used one year"=2, "used 10 years"=3.

4.2 The Second Difference Method (SDM)

In this Section we construct the SDM whose objective is to define a set of significant SNPs from a distribution of BF results obtained from a BAYENV analysis. We also define how it can be used on multiple runs of the program to ensure more reliable results.

4.2.1 The definition of SDM

We here propose a method, named the SDM, which aims to define a dynamic significance threshold for the BF results gained from a BAYENV analysis. Our hypothesis is that the true significant results can be separated from the non-significant results by looking at the sorted empirical distribution of BFs as a gradually increasing function where the true significant results are found after a critical break in the slope. Such a critical point can be found where the second derivative of the function has a sudden and substantial jump in the positive direction. Further, we assume that the insignificant results are found in regions where the same function has a linear growth and the true significant results are found in regions where the

function has a convex growth (see Section 2.4). Thus, by setting the cutoff threshold where the function has an accelerating growth rate, we expect to be able to access the significant results without including non-significant results.

Since we are dealing with discrete data, the second derivative must be approximated using second difference. By sequentially calculating the second difference for each BF value, we can detect where the distribution has an approximately linear growth (i.e. $\Delta^2 y \approx 0$, see Section 2.4.1) and where the distribution enters a more extreme growth rate (e.g. $\Delta^2 y > 1$).

Consider a list of increasingly sorted BF results $\mathbf{y}_i$, $i = 1, 2, \ldots N$. Let y_n be the BF value at position n, $1 < n < N$, $n \in \mathbb{N}$ then the corresponding second difference is calculated as follows

$$\Delta^2 y_n = y_{n+1} - 2y_n + y_{n-1}, \quad y \in \mathbb{R}, \quad n \in \mathbb{N} \tag{4.1}$$

To identify the regions with convex growth, choose a threshold $\delta > 0$ and let the first value y_k where $\Delta^2 y_k > \delta$ be the starting point of where to expect the distribution of BF values to grow quadratically or exponentially (convex growth). We are interested in the values after the initial change in the growth rate has been made, thus the set of significant SNP's with associated BF value $\mathbf{y}$ can be written $\omega = \{y_{k+1}, y_{k+2}, \ldots, y_N\}$. We will in the subsequent chapters refer to ω as the significance set from the BAYENV analysis.

4.2.2 Selecting the threshold δ

In regions where the distribution of BFs is approximately linear, Δ^2 will be fluctuating around zero. Therefore, the key factor for successfully detecting the critical point is to choose the correct δ. A simple solution would be to choose a constant δ (e.g. 0.5 or 2). However, we want δ to be more in agreement with the shape of the empirical distribution. In other words, if there are many SNPs with a very high BF (e.g. BF > 150), we can allow δ be less sensitive by allowing a greater deviation from zero. In an opposite case where the general BF level is low, we can choose a δ that is more sensitive (closer to zero).

There are several ways we can scale δ such that the cutoff is better adjusted to the shape of a specific distribution. We here suggest one variant where δ is scaled according to two important measures from the distribution: 1) The maximum BF value and 2) The number of BF values above a certain level (e.g. a level from *Jeffrey's* table). We define the scaled and dynamic cutoff threshold for the second difference Δ^2 as follows.

Let α be the lower limit for a "strong" support from *Jeffrey's* "scale of evidence" table (i.e. log(BF)=2, see Table 2.2). Then let N_α be the number

of elements in the empirical distribution that has a BF $> \alpha$. Finally let A be the maximum BF value in the same distribution. Thus, a dynamic cutoff level $\hat{\delta}$ for $\Delta^2 y$ can be obtained using the following equation:

$$\hat{\delta} = \begin{cases} \epsilon + \log_{10} N_\alpha \log_{10} A & \text{if} \quad N_\alpha \geq 1 \\ \epsilon & \text{if} \quad N_\alpha = 0 \end{cases}, \quad 0 < \epsilon \leq 1, \quad \epsilon, \alpha, A \in \mathbb{R}, \quad N_\alpha \in \mathbb{N} \tag{4.2}$$

Where ϵ is a small real valued constant, $0 < \epsilon \leq 1$. The main purpose of ϵ is to ensure that the algorithm works even if there are no SNPs with BF $> \alpha$ ($N_\alpha = 0$). Additionally, ϵ can be used to "fine tune" the sensitivity of the algorithm. However, for the subsequent analyses a default value of $\epsilon = 0.5$ is used unless otherwise stated.

The pseudo code in algorithm 2 summarises how the SDM pipeline is carried out:

Algorithm 2 The Second Difference Method Algorithm

```
y ← sort(BF results, increasing=True)
N ← length(y)
δ̂ ← ε + log10(N_α(y)) × log10(A(y))
k ← 0
for each BF value y_i, i = 2, ..., N − 1 do
    Δy_i ← y_{i+1} − 2y_i + y_{n−1}
    if Δy_i is greater than δ̂ then
        k ← i
        break for loop
    end if
end for
ω ← [y_{k+1}, y_{k+2}, ..., y_N]
```

4.2.3 Defining the set of significant SNPs across multiple runs of BAYENV

The proposed SDM provides a dynamic cutoff threshold for the distribution from one single run of BAYENV. However, to address the problem of run-to-run variability (Blair, Granka and Feldman 2014) we need to verify the stability of the results across multiple runs of BAYENV (i.e. all SNPs in the dataset are tested for association to the environmental variable once in each run). We can ensure that the results are more reliable by: 1) computing the union of the significance sets (ω) calculated using the SDM algorithm (see Algorithm 2) for each run. 2) Count the number of times each SNP in the union set has been defined as significant. A final set of significant SNPs can be defined as the SNPs that appear more frequently than a specified cutoff threshold. We first define the union set as follows.

Consider that T runs has been carried out using BAYENV. Let ω_i be the identified set of significant results from test run τ_i, $i = 1, 2, \ldots, \mathrm{T}$. Then the complete set of significant results Ω is defined as:

$$\Omega = \bigcup_{i=1}^{\mathrm{T}} \omega_i \tag{4.3}$$

We expect that each ω_i will contain different number of SNPs and consequently not always the same SNPs. In order to define a final set of significant results we count the number of times each SNP appears in Ω and use a cutoff for maximum variability to remove outlier results (i.e. the least consistent SNPs).

Let S be the total number of unique SNPs y in Ω, k_i the number of times the specific SNP y_i has been defined in Ω and κ be a predefined cutoff value, then the final set of significant SNPs $\hat{\Omega}$ can be written as follows:

$$\hat{\Omega} = \bigcup_{i=1}^{S} \begin{cases} y_i & \text{if} \quad \frac{k_i}{\mathrm{T}} \geq \kappa \\ \emptyset & \text{if} \quad \frac{k_i}{\mathrm{T}} < \kappa \end{cases}, \quad i \in \mathbb{N} \tag{4.4}$$

We suggest that $\kappa \geq 0.7$ in order to retain only the SNPs that are consistently included in more than 70 percent of the ω's. We will refer to Ω and $\hat{\Omega}$ as the "union set" and "total significance set" (TSS) of a BAYENV analysis respectively, carried out using T independent runs.

For the subsequent experiments, we will be using the same strategy for defining union sets and TSS for other cutoff methods such as percentage or static cutoffs. The only difference is that instead of the SDM we use one of the other cutoff methods when we define the significance sets (ω).

4.2.4 Stability score

To measure the between-run variability as well as the variability between complete BAYENV analyses we introduce a stability score s. The idea is to have a standardised measure for the discrepancy within and between significant sets.

Let $\bar{x}$ be the mean number of times SNPs have been identified in a union set (Ω, see Equation 4.3). Then we normalize this mean such that 0 reflects that no SNPs have been identified more than once (i.e. $s(\bar{x} = 1)$) and 1 if all SNPs have been identified in all runs. For T independent runs of BAYENV, we write the stability score s as

$$s(\bar{x}) = \frac{\bar{x} - 1}{T - 1}, \quad T > 1 \tag{4.5}$$

By using this score we have a robust and easy measure of the run-to-run variability between union sets.

4.3 Reducing the time consumption by reducing the test set

Running BAYENV tests on many loci is a very time consuming task due to its extensive use of MCMC algorithms (see Section 3.6.2). A dataset may potentially contain several million SNPs and many of these may not be worthwhile testing in this context because the difference in allele frequencies are marginal between populations. For example, if all populations have an allele frequency of 0.5 at a particular locus, this must be considered to be a neutral for selection (in this context) and may therefore be excluded from testing. Considering that the time consumption in the test phase of BAYENV increases linearly with the number of SNPs tested, it can be well worth trying to reduce the number of test SNPs to a minimum.

First, loci that are monomorphic (loci where all populations exhibit only one and the same allele) across all populations can obviously be removed from the tests set as these contain little information in this context. In order to further reduce the amount of test SNPs, we suggest using the measure maximum allele frequency difference (MAFD) across populations as a cutoff. The MAFD is an indicator of the maximum deviance in allele frequencies between the populations being assayed. The MAFD can be computed in the following way:

Let N be the number of loci in the test set $L = l_1 \dots l_N$, K be the number of populations and x_k^l be the allele frequency for population k in locus l, then the maximum allele frequency difference Δ_l for locus l can be defined as

$$\Delta_l = \max_{1 \le i \le K-1, i < j \le K} \left| x_i^l - x_j^l \right| \quad i, j \in \mathbb{N} \tag{4.6}$$

We can use this equation to extract SNPs with high maximum allele frequency difference, Δ_l, by applying an appropriate cutoff threshold κ.

Let N be the number of loci in the original data set and Δ_l be the maximum allele frequency difference for locus l as described in equation 4.6. Furthermore, let κ, $0 < \kappa < 1$, be an appropriate cutoff value for Δ. Thus the reduced test set Ψ is defined as

$$\Psi = \bigcup_{i=1}^{N} \begin{cases} l_i & \text{if} \quad \Delta_i \geq \kappa \\ \emptyset & \text{if} \quad \Delta_i < \kappa \end{cases}, \quad i \in \mathbb{N}, \quad \Delta, \kappa \in \mathbb{R} \tag{4.7}$$

Note that this method also will exclude loci that are monomorphic across all populations as these will have $\Delta = 0$.

4.4 General methods for the tests performed

In this Section we describe the methods and experiments used to explore and test the previously derived hypotheses and questions.

Significance sets

We compared the significance sets obtained using the SDM (see Section 4.2) to corresponding sets obtained by using percentage and static BF cutoff thresholds on the empirical distribution of BF results. As a rank statistics for the BF results, we calculated the empirical p-value (see Section 2.5) by employing equation 2.15. Hence, we used the cutoff $\alpha_{\hat{p}} \leq 0.01$ and $\alpha_{\hat{p}} \leq 0.05$ to refer to the top 1% and 5% SNPs in the empirical distribution of BFs respectively. We refer to these cutoff thresholds as *alpha1* and *alpha5* respectively. Second, we employed two static cutoff thresholds based on Jefferey's scale of evidence for BF (table 2.2): 1) "Substantial" (BF $\geq$ 3.2). 2) "Strong" (BF $\geq$ 10). We refer to these cutoff thresholds as *jeff3.2* and *jeff10* respectively.

Conversion of BFs to q-values

As a measure to control for FDR in the significance sets, we followed the algorithm by Muller, Parmigiani and Rice 2006 as employed in Villemereuil et al. 2014 (see Section 2.6.2) to convert the BFs to q-values by using algorithm 1 (page 32). As the prior probability for the alternative model needs to be specified, we followed Villemereuil et al. 2014 and defined $P(M_1) = 0.01$ (the prior odds of alternative model, see equation 2.16). From the q-values we computed the expected number of FPs (as elaborated in Section 2.6.1) in each significance set and compared these values.

Naming conventions

We use the term "BAYENV run" (or just "run") when we perform the test for environmental correlation on all or a subset of the SNPs in the dataset. We will also use this term when a predefined subset of a full dataset is used.

For a set of multiple runs of BAYENV we use the term "BAYENV analysis" or a "full analysis".

General BAYENV MCMC parameters

For the covariance matrix, we used an average of all covariance matrices estimated by BAYENV after the same number of MCMC iterations as were specified for the corresponding tests (unless otherwise stated). For example if the tests in a run were carried out using 10,000 iterations, the covariance matrix was calculated as an average from 10,000/500=20 estimations performed by BAYENV. The starting points (random seeds) for the MCMC algorithm were drawn randomly both for the calculation of the covariance matrix and each independent test run.

4.4.1 Testing for correlation to environmental variables

For all subsequent BAYENV analyses on the *Cod* data set (see Section 4.1.1), tests for correlation to all available environmental variables (*sal1*, *sal2*, *temp1*, *temp2*, *ox1* and *ox2*; see Section 4.1.1) were carried out for each SNP. Likewise, for the *Maize* data set, tests for correlation to the ordinal "environmental" variable (see Section 4.1.2) were carried out for each SNP.

4.4.2 Manhattan plots

To visualize how the BF results are distributed over the entire genome, we use Manhattan plots. A Manhattan plot is a type of scatter plot often used to display data when there are many data points (i.e. genome wide tests). On the x-axis we displayed the SNPs according to its position in each Linkage group/chromosome. The associated $\log_{10}$(BF)/BF are displayed on the y-axis. The linkage groups/chromosomes are plotted with different colours for easier inspection of the data.

4.4.3 Plotting of the union sets

As a way to visualize the variability within each union set, we sort the SNPs in Ω (see Section 4.2.3) according to the number of times, k, it has been identified as significant. This vector is then plotted as a histogram with the value of k on the y-axis. The cutoff value κ is plotted as a horizontal line. Thus, SNPs that are included in $\hat{\Omega}$ (TSS) are the SNPs above this line. We will be referring to this as a significance-plot.

4.5 Methods for testing the convergence of the covariance matrix

To investigate the consistency and convergence of the covariance matrix used by BAYENV as the null model (see Section 3.2), we ran the first step (the estimation of the covariance matrix) of the algorithm using all (8809) SNPs in the *Cod* dataset as basis. Three test runs were performed using 10,000, 100,000 and 500,000 MCMC iterations respectively. To serve as a reference for the tests, we ran a second covariance estimation using 500,000 MCMC iterations and calculated an average matrix based on all estimates outputted from BAYENV (single draws from the posterior is output every 500 MCMC iteration). The random seed was chosen randomly for all runs. As test matrices, we used the last covariance estimate (a single draw from the posterior) from each test run. Additionally, for all test runs we calculated an average matrix based on all covariance matrix estimates output by BAYENV. As a measure of deviation between the different covariance estimates, we subtract each test matrix from the reference matrix and plot the absolute value of the difference as heatmaps. The heatmaps are plotted using MATLAB (MATLAB 2012).

4.6 Methods for testing the SDM

4.6.1 Testing the SDM on simulated BF values

To verify that the SDM works as intended and to demonstrate the capabilities of the method, we simulated three sets of artificial results designed to represent three different outcome scenarios from a BAYENV analysis. We used *R* to draw random samples from the uniform distribution (*runif*) in specific intervals according to what scenario we wanted to simulate. The sample intervals were divided into three different groups representing neutral SNPs supporting the null model (*neSNPs*), "non-significant" SNPs supporting the alternative hypothesis (*noSNPs*) and "significant" SNPs (*noSNPs*) respectively. Between the *isSNPS* and the *siSNPs* groups we deliberately made a gap between the sampling intervals to act as a trigger point for the SDM algorithm (i.e. the SDM should be able to separate the distribution at this point).

In the first set we simulated a scenario where the BF signal was low in general with only a few high ranking "SNPs". In this set we drew 9,900 samples in the interval $[0.01, 1]$ (*neSNPs*), 95 samples in the interval $[1, 10]$ (*noSNPs*) and five samples in the interval $[20, 1000]$ (*siSNPs*). In total the set consisted of 10,000 "SNPs". We will refer to this set as *Sim-weak*.

In the second set, we simulated a scenario where there were in general a strong BF signal with relatively many high ranking "SNPs". In this set

we drew 8,500 samples in the interval $[0.01, 1]$ (*neSNPs*), 1350 samples in the interval $[1, 300]$ (*noSNPs*) and 150 samples in the interval $[500, 10000]$ (*siSNPs*) - in total 10,000 "SNPs". We will refer to this set as *Sim-strong*.

In the third set, we increased the total number of simulated values to 100,000 and added a second interval of *noSNPs*. The rationale for this was to test how the SDM algorithm reacted on a large dataset as well as a different distribution of the simulated BF values (different sampling intervals). Out of the 100,000 samples, 85,000 were drawn in the interval $[0.01, 1]$ (*neSNPs*), 13,000 in the interval $[1, 10]$ (*noSNPs*), 1,980 in the interval $[10, 1000]$ (also *noSNPs*) and 20 in the interval $[10000, 10000000]$ (*siSNPs*). We refer to this set as *Sim-large*.

We used standard settings for the SDM algorithm (see Section 4.2.2), i.e. $\alpha = 10$ and $\epsilon = 0.5$.

To visualise how the second difference (Δ^2, see equation 4.1) is distributed according to the simulated and sorted BF values, we plot $\log_{10}$ transformed value of Δ^2 as scatter plots in *R*. To improve the readability of the plots, all values $\Delta^2 < 10^{-5}$ are set to 10^{-5}. For comparison the we plot the $\log_{10}$ transformed sorted simulated BFs.

4.6.2 Testing the SDM on a single BAYENV run on the *Cod* dataset

As an initial test on the real BAYENV results, we tested the SDM on a single run of BAYENV on the *Cod* dataset and compared it to three other cutoff methods. The tests for correlation to the environmental variables were carried out using 500,000 MCMC iterations for each SNP. In this experiment we compared the significance sets (one set for each variable) from SDM to the corresponding significance sets obtained using the cutoff thresholds *jeff3.2*, *alpha1* and *alpha5*. The BFs for each SNP were converted to q-values as a measure of FDR (see Section 4.4). The lowest q-value in each set was used to obtain information on the expected proportion of FPs and compute the expected number of FPs in each significance set.

To visualise how the second difference (Δ^2) is distributed according to the sorted BF results (see equation 4.1) and where the cutoffs is made, we plot the $\log_{10}$ transformed second difference value (Δ^2) as scatter plots in *R*. To improve the readability of the plots, all values $\Delta^2 < 10^{-5}$ are set to 10^{-5}. Two scatter plots of the corresponding $\log_{10}$ transformed BF values are made to be compared to the second difference distribution.

Plots of the FDR

To visualise the FDR within the significance sets found using different cutoff methods, we plot the expected number of FPs and TPs for each cutoff

method and environmental variable in *R* as a stacked barplot. We use the significance sets obtained in the experiment described in Section 4.6.2. The number of FP is obtained by multiplying the number significant SNPs with the maximum q-value in each set (see Section 2.6.1). The number of TPs is found by subtracting the number of FPs from the total number of SNPs in the same set. We use a logarithmic scale ($\log_{10}$) to improve the readability of the plot. Due to the logarithmic scale, all values (number of FP/TP) below one are plotted as zero. The stacked barplot is plotted in *R* using the package *ggplot2*.

4.6.3 Testing the SDM on multiple BAYENV runs on the *Cod* dataset

To test how the variability of BAYENV affect the different cutoff methods, 32 independent runs of BAYENV on the *Cod* dataset were carried out using 500,000 MCMC iterations for each SNP. For this experiment we used five different cutoff methods: SDM, *alpha1*, *alpha5*, *jeff3.2* and *jeff10*. For each run we found the significance sets and calculated the union set across all runs for each cutoff method individually. For all union sets we employed a threshold, $\kappa = 0.7$, for maximum variability across runs (equation 4.4). We refer to these final sets of significant SNPs as total significance sets (TSS).

For each environmental variable and cutoff method we calculate a maximum q-value for the TSS's based on the median BF obtained from the 32 runs. We also use the median statistics to summarize the maximum and minimum BF for each TSS.

Venn diagrams

We plotted Venn diagrams to visualise how the significant SNPs are distributed among the different union sets and TSS's. A Venn diagram is used in set theory to visualise the union, intersection and difference between two or more sets. By using Venn diagram, we have a tool to show how SDM performed compared with the cutoff methods *alpha1*, *alpha5* and *jeff10*. The Venn diagrams are plotted using *R* and the package *VennDiagram*. As input we use the union sets and TSS's obtained from the experiment described in Section 4.6.3.

4.6.4 Testing the SDM on the *Maize* dataset

To test our second difference method (SDM) on a different dataset, we ran a full BAYENV analysis on the *Maize* data (see Section 4.1.2). Since this dataset is small, a high percentage of SNPs under positive selection will likely affect the neutralness of the covariance matrix. Therefore, we

used only the random SNPs as a basis for the null model (100 SNPs). The covariance matrix used was an average of 200 draws from the posterior distribution obtained after 100,000 MCMC iterations. We ran 32 replicate runs of BAYENV where we tested all SNPs in the dataset (135 SNPs) for correlation to the stage variable. The test runs were carried out using 100,000 MCMC iterations for each SNP. The BFs were averaged over the 32 runs and are plotted as Manhattan plot in *R* (see Section 4.4.2).

To assign a cutoff threshold for significance, we applied SDM to all runs individually and calculated the union set. The TSS was obtained after applying a cutoff threshold of $\kappa = 0.7$ for maximum variability between runs. Because we expected a generally low BF signal in this test, the sensitivity constant ϵ was adjusted from 0.5 to 0.2 in the $\hat{\delta}$-equation (see equation 4.2). The resulting union set is plotted as a significance-plot (see Section 4.4.3).

As a measure of FDR, we converted the average BF from the 32 runs to q-values (section 4.4) in the same manner as for the analysis on the *Cod* dataset (section 4.6.3).

4.7 Methods for testing the stability of the BAYENV method

4.7.1 Testing the stability of BAYENV by comparing analyses carried out using different number of MCMC iterations

To investigate whether a high run-to-run variability is a function of the number of iterations for the BAYENV algorithm, we carried out several analyses on the *Cod* dataset where we gradually increased the number of iterations. Thirty-two runs using different random seeds were carried out in eight independent BAYENV analyses using 10,000, 25,000, 50,000, 75,000, 100,000, 500,000 and 1,000,000 MCMC iterations respectively. A second analysis using 500,000 iterations was performed to serve as a reference set. For each analysis we calculated a union set based on cutoffs using the SDM and *alpha1*. For all union sets we calculated the stability score (Equation 4.5 on page 53) as a measure of the run-to-run variability.

4.7.2 Testing the relationship between run-to-run variability and the number of independent BAYENV runs

To explore how many independent runs that are needed in order to get a stable result, we used the results from two different BAYENV analyses each consisted of 32 independent runs carried out using 500,000 MCMC iterations each. The first analysis was used as a test set and the second

served as a reference set. We divided the first test set into six different subsets where we doubled the number of runs for each subset starting with one run and ending with 32 (i.e. same as the reference set). For the subsets containing 2 to 32 runs and the reference set, we calculated the median BF for all six variables. For the six environmental variables we calculated the set difference between the test sets and the reference set for an *alpha1* cutoff. The differences were plotted as the percentage of equal SNPs.

4.8 Testing the method of reducing the test set by excluding SNPs with low maximum allele frequency difference

To test our method of excluding SNPs with small allele frequency difference between populations (see Section 4.3), we carried out four BAYENV analyses on the *Cod* dataset. For each analysis we included only SNPs that had a maximum allele frequency difference (MAFD) in the top 90, 95, 97.5 and 0.99 percentile of the empirical distribution of the MAFD respectively. As null model for these BAYENV analyses, a covariance matrix from a previously performed analysis carried out using 500,000 MCMC iterations was used (all 8809 SNPs were used as basis). The four analyses consisted of 32 replicate runs of BAYENV using 100,000 iterations and different random seeds for the MCMC algorithm. The median statistics were calculated for all six environmental variables. As a reference we used the median statistics from the experiment in Section 4.6.3. For the six environmental variables we calculated the set difference between SNPs ranking among top 88 SNPs (*alpha1* cutoff) in the test and in the reference analysis respectively. Additionally, we compared the 20 top ranking SNPs of the test sets and the reference set. The differences are plotted as the percentage of equal SNPs. We also compute the union set and TSS using the SDM for each analysis. As a reference for the SDM results, we use the TSS's obtained from the experiment in Section 4.6.3.

To calculate the time savings of reducing the test set, we employed the time estimates provided by PYBAYENV (see Section 4.9.8 and 5.1.4). The time estimates from testing the reduced sets are compared to the time estimates for the full dataset.

4.8.1 Plots of the correlation between allele frequency difference and BAYENV results

To visualise the correlation between the maximum allele frequency between populations and the BAYENV results (BFs), we plot these two units as scatter plot in *R*. The BFs are logarithmic transformed for better readability. A spline regression is performed to show the trend in the data and to

pinpoint the significance. We use the R method *smooth.spline()* to visualize the spline and the R library *mgcv* to determine the significance. Significant SNPs found using the SDM method on the full dataset (from experiment in Section 4.6.3) are plotted in red to show where these are located in the distribution. Four vertical lines are plotted indicating the 90, 95, 97.5 and 0.99 percentile cutoffs for the maximum allele frequency difference.

4.9 Functional specifications for PyBayenv

To address the challenges described in chapter 3.6 we constructed a program, PyBayenv. There were three main purposes for developing PyBayenv: 1) Simplify the Bayenv analysis by implementing a file format converter and by streamlining of the process in general (i.e. estimating the covariance matrix and carry out all tests in one single run). 2) Reduce the time usage when performing multiple Bayenv analyses by parallelizing the process. 3) Implement our suggested methods (section 4.2 and 4.3). In the subsequent sections we outline the functional specifications for PyBayenv.

4.9.1 The main purpose of PyBayenv

PyBayenv should provide a user friendly way to perform multiple Bayenv analyses on a large set of SNP data. In a one-step procedure PyBayenv should be able to convert data from a common file format, estimate the covariance matrix, run multiple tests on selected SNPs, exclude SNPs from testing based on the maximum allele frequency difference (MAFD) between populations (section 4.3) and interpret the results using SDM (section 4.2).

4.9.2 The conversion between formats

PyBayenv should be able to read in SNP data from the Genepop format (Raymond and Rousset 1995) and output the data in the Bayenv file format (see Table 3.1 and 3.2). The Genepop format was chosen because it contains population information and thus is particularly suitable in this context. Additionally, the Genepop format is widely used. The reader is referred to the manual for details about the Genepop format (http://kimura.univ-montp2.fr/~rousset/Genepop.pdf).

4.9.3 Standardising environmental variables

PyBayenv should provide a function for standardisation of the environmental variables (Bayenv require that the environmental variables are

standardised to have mean zero and standard deviation of one). The standardisation should be performed by subtracting the mean and dividing by the standard deviation.

4.9.4 Estimation of the covariance matrix

PyBayenv should estimate the covariance matrix based on a set of user selected SNPs. Furthermore, the covariance matrix used in the test phase should be an average of all covariance matrices output by Bayenv (output every 500 iterations).

4.9.5 The test for environmental correlation

PyBayenv should be able to sequentially test all SNPs of interest. (recall that Bayenv only supports testing one SNP without restarting the program - see Section 3.3)).

4.9.6 Random seed

The random seed for Bayenv should be chosen randomly by PyBayenv for each run of the algorithm. This applies both to the estimation of the covariance matrix and test phase.

4.9.7 Parallelization

The user should be able to run multiple Bayenv analyses using a single run of PyBayenv, thus taking advantage of today's multi core CPUs.

4.9.8 Timekeeping and time estimates

PyBayenv should provide the user with information about the time consumption of the tests at each SNP. Moreover, PyBayenv should continuously update the user with an estimate of how long it will take for the analysis to complete.

4.9.9 Documentation of the Bayenv analyses

PyBayenv should document all steps in the procedure in accordance with Sandve et al. 2013. This includes random seeds used and the number of MCMC iterations for each run.

4.9.10 Defining a set of significant SNPs based on SDM

PyBayenv should implement the second difference algorithm described in Section 4.2 to assign a significance threshold for the results from for each run. Furthermore, if multiple runs are carried out, the program should calculate the set of significant SNPs using equation 4.3 (page 52) and write the results to file.

4.9.11 Reducing the test set based on maximum allele frequency difference

PyBayenv should implement the method for reducing the set of test SNPs based on a maximum allele frequency difference between populations (MAFD) (described in Section 4.3).

4.9.12 The user interface

PyBayenv should have a command line interface that is available from the UNIX platform.

4.10 Testing the time consumption using PyBayenv in parallel mode

As previously discussed, the variability of Bayenv requires that the results are verified by comparing two or more replicate runs of the program (see Blair, Granka and Feldman 2014). As potential time saving measure, a multi-processing feature was implemented in PyBayenv (see Section 4.9.8 and 5.1.4). To test the time saved using this feature, we first ran a Bayenv analysis on the cod dataset using PyBayenv in single-processing mode as a reference for the time consumption. Next, we ran three Bayenv analyses where respectively 8, 16 and 32 replicate runs were carried out in parallel. The estimated time consumption for the reference run was compared to the estimated time consumption for the parallel runs. All analyses were carried out using 100,000 MCMC iterations for each SNP and was executed on a desktop computer running Redhat Linux with an Intel i7 CPU having 8 cores.

Chapter 5

Results

5.1 Implementation of PyBayenv

To address the challenges described in chapter 3.6 we constructed a program, PyBayenv, whose main purpose was to facilitate the use of Bayenv and extend it with functions for testing our hypotheses. PyBayenv was constructed in accordance with the specifications described in Section 4.9. As the name indicates, the program was written in the *Python* programming language (http://python.org/). In the subsequent sections we summarize how PyBayenv was constructed. The architecture of PyBayenv package and classes is visualized in the UML diagrams in Figure 5.2 and 5.3 respectively. A snapshot of the PyBayenv man page (help menu), describing the parameter options for the program, is shown in Figure 5.6. The PyBayenv package is available online at http://folk.uio.no/kristori/thesis/pybayenv/ (username: sdm, password: PyBayenv). Example files are supplied, please see the README file for instructions.

5.1.1 The conversion between formats

We followed the requirements in Section 4.9.2 and implemented a format converter in the PyBayenv application. Hence, the default input for PyBayenv is the Genepop format. When converted, the Bayenv format is written to file in two versions: one in the standard Bayenv format (see Table 3.1 and 3.2) intended for the estimation of the covariance matrix, and one for the test phase containing additional SNP information. Figure 5.4 shows a snapshot of the Genepop format before conversion and the two Bayenv formats when processing the *Cod* dataset.

NAME

PyBayenv - A wrapper for BAYENV

SYNOPSIS

PyBayenv [OPTION]...

DESCRIPTION

PyBayenv is a program that enables the user to run a full BAYENV 2.0 analysis of a SNP data set. The input file must be in GENEPOP format. The user can take advantage of multi-core CPUs by carrying out several independent runs of BAYENV 2.0 in parallel. Additionally, the user may reduce the test set by excluding SNPs based on the maximum allele frequency difference (MAFD) between populations.
PyBayenv also calculates a significance level for the results based on the Second Difference Method (SDM).

OPTIONS

--covsize the number of loci in the null model
-c same as --covsize
--debug run program in debug mode.
-d same as --debug
--envfile environment variables to be used
-e same as --envfile
--file Input file
-f same as --file
--help display this help and exit
-h same as --help
--iterations number of iterations for the null model
-i same as --iterations
--reduce reduce a percentage of the SNPs based on MAFD
-r same as --reduce
--numpop number of populations in the dataset
-n same as --numpop
--skipcov skip building covariance matrices
-s same as --skipcov
--testsize number of loci to test
-t same as --testsize
--numenv number of environment variables to test
-z same as --num_env
--nullfile file for the null model
-l same as --nullfile
--numtests number of tests to perform
-p same as --numtests
--skiptest skip the test part
-b same as --skiptest
--epsilon the sensitivity of SDM
-E same as --epsilon

Figure 5.1: The man page for PyBayenv describing the user options for the program.

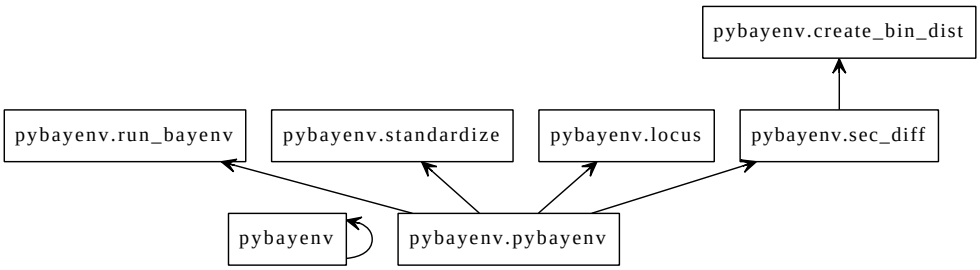

Figure 5.2: UML of the modules in the PyBayenv package

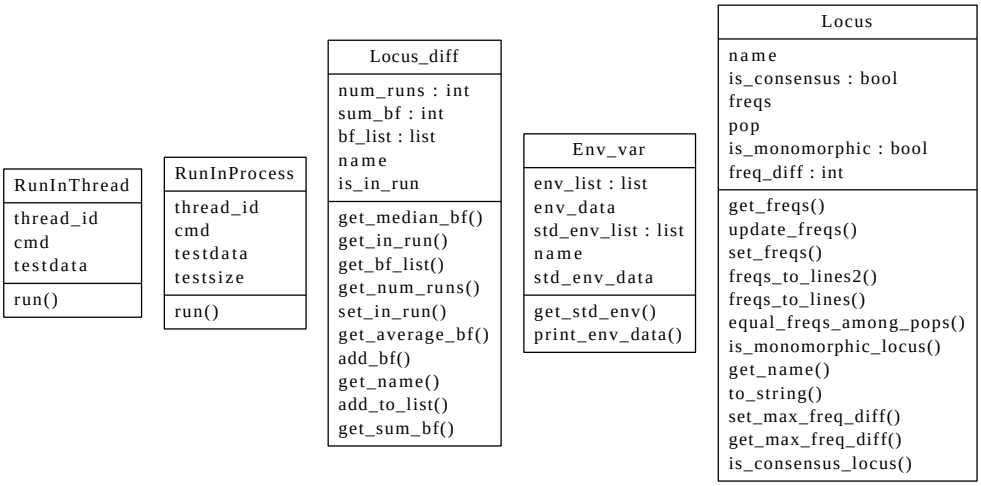

Figure 5.3: UML of classes in PyBayenv

5.1.2 Standardizing environmental variables

In accordance with the specifications in Section 4.9.3 a function for standardising the environmental variable(s) was implemented in PyBayenv. Thus the input variable(s) for PyBayenv are unstandardised and are standardised using the *pybayenv.standardize* module. PyBayenv standardises the variable(s) to have mean zero and a standard deviation of one. The standardised environmental variable(s) are written to file. Figure 5.5 shows the unstandardised input variables and the standardised output variables (see Section 4.1.1) used by PyBayenv to analyse the *Cod* dataset.

5.1.3 Estimation of the covariance matrix

We followed the requirements in Section 4.9.4 and implemented a wrapper function for running the first step of a Bayenv analysis - the estimation of the covariance matrix - within PyBayenv . The function *commands.getstatusoutput()* is used to run Bayenv from *Python*. The file containing the original Bayenv format (see Section 5.1.1) serves as input for Bayenv in this step. A random seed in the interval 1-99999 is drawn

A

```
#Header information
CAN-rs119055103,CAN-rs119055173,CAN-rs119055284,CAN-rs119055307,CAN-rs119055454,CAN-rs119055470,CAN
Pop
ST_1001, 0202 0101 0202 0202 0202 0102 0101 0101 0102 0202 0202 0202 0102 0101 0102 0202 0102 0202
ST_1002, 0102 0101 0202 0102 0102 0101 0102 0101 0101 0202 0202 0101 0101 0101 0102 0202 0101 0202
ST_1003, 0202 0101 0202 0202 0102 0101 0101 0101 0101 0202 0202 0202 0101 0101 0102 0202 0101 0202
ST_1004, 0202 0101 0202 0202 0101 0101 0101 0303 0101 0202 0202 0102 0101 0101 0202 0202 0101 0202
ST_1006, 0202 0101 0202 0202 0102 0102 0101 0101 0101 0202 0202 0202 0101 0101 0202 0202 0101 0202
ST_1009, 0202 0101 0202 0202 0202 0101 0101 0103 0101 0202 0202 0202 0101 0101 0202 0102 0102 0202
ST_1010, 0102 0101 0202 0202 0102 0101 0101 0101 0101 0202 0202 0101 0101 0101 0102 0202 0101 0202
ST_1012, 0102 0101 0202 0202 0202 0102 0101 0101 0101 0202 0202 0102 0102 0101 0102 0102 0102 0202
ST_1014, 0102 0101 0202 0102 0102 0101 0101 0103 0101 0102 0202 0102 0101 0101 0102 0202 0102 0202
```

B

```
6    0    7    2
90   112  77   94
96   112  83   94
0    0    1    2
0    1    2    1
96   111  80   95
8    7    7    5
88   105  77   91
32   56   30   34
64   56   54   62
86   105  71   81
10   7    13   15
84   111  76   85
12   1    8    11
67   72   60   65
29   40   24   31
69   92   69   67
27   20   15   29
3    8    1    3
93   104  83   93
```

C

```
CAN-rs119055103  6    0    7    2
90   112  77   94
CAN-rs119055173  96   112  83   94
0    0    1    2
CAN-rs119055284  0    1    2    1
96   111  80   95
CAN-rs119055307  8    7    7    5
88   105  77   91
CAN-rs119055454  32   56   30   34
64   56   54   62
CAN-rs119055470  86   105  71   81
10   7    13   15
CAN-rs119055496  84   111  76   85
12   1    8    11
CAN-rs119055497  67   72   60   65
29   40   24   31
CAN-rs119055505  69   92   69   67
27   20   15   29
CAN-rs119055519  3    8    1    3
93   104  83   93
```

Figure 5.4: Snapshots of the input and output formats of PyBayenv when analysing the cod dataset. A) The Genepop format where the genotype information is organized as one line per individual with one column for each SNP. B) The original Bayenv format (see Table 3.1) and C) the alternative Bayenv format intended for the test phase displaying allele counts for the four populations (see Section 4.1.1) on the first 10 SNPs in the dataset.

A

"Salinity surface"	24,2	7,4	34,7	16,4
"Salinity deep"	34,1	13,6	34,8	30,9
"Temperature surface"	3,5	13,6	6,0	3,4
"Temperature deep"	6,3	5,7	6,1	6,4
"Oxygen surface"	8,0	7,4	6,8	8,5
"Oxygen deep"	6,5	4,5	6,7	6,2

B

0.35091242269518724	-1.3215212514265584	1.3961834690212787	-0.4255746402899091
0.66536265418746388	-1.706799852046103	0.74636332513202419	0.29507387272661406
-0.75128278357813705	1.6768631729464019	-0.1502565567156274	-0.77532383265263738
0.65275336576821896	-1.5852581740085327	-0.0932504808240327	1.0257552890643464
0.50951017108525365	-0.43112399091829051	-1.3717581529218361	1.2933719727548743
0.60345824189498243	-1.69543029865733	0.83334709595021383	0.25862496081213576

Figure 5.5: Snapshot of the environment variables before and after the standardisation performed by PyBayenv . A) The environmental variable input format (unstandardised). B) The environmental output format (after standardisation). The variables are for the *Cod* dataset (see Section 4.1.1). The output has been slightly modified with extra tabs for better visualisation.

randomly by PyBayenv according specifications in 4.9.6 using the *Python* function *random.uniform()*. The time consumption for creating the covariance matrix is traced using the *time.time()* function in *Python* and the time usage is written to file in accordance with the requirements in Section 4.9.8 and 4.9.9. The output from Bayenv , which is a file containing a total of (number of iterations/200) covariance matrix estimates, is read by PyBayenv , converted to a *NumPy* array and averaged using the *NumPy* function *average()* (Jones, Oliphant, Peterson et al. 2001–).

5.1.4 The test for environmental correlation

A wrapper for the test phase of Bayenv was implemented in PyBayenv in accordance with the specifications in Section 4.9.5. As for the first step (estimating the covariance matrix), the function *commands.getstatusoutput()* is used to run Bayenv within *Python*. The alternative Bayenv format file (see Section 5.1.1 and Figure 5.4) is used to sequentially create individual files for each SNP, with file names corresponding to the SNP name specified in the file. The individual SNP file is written to disc before it is passed on as an argument to Bayenv through the *Python* method *commands.getstatusoutput()*. After the Bayenv test is completed, the individual SNP file is deleted. This process is repeated for all SNPs given as input to PyBayenv . The resulting BF for each SNP is appended

to a result file. The *Python* function *time.time()* is used to trace the time spent on testing each SNP in accordance with the specification in Section 4.9.8. This information is used by PyBayenv to estimate a finish time for the complete analysis. We note that tests showed that the process of writing and deleting files on disc had a negligible effect on the time spent on testing each SNP (result not shown).

5.1.5 Parallelization

As a time saving measure, we implemented support for carrying out multiple runs of Bayenv in parallel using PyBayenv (in accordance with the specification in Section 4.9.7). PyBayenv uses the library *Python Multiprocess* and the function *RunInProcess()* to run a user specified number of parallel instances of the test function (described in Section 5.1.4). For each process one individual SNP file is written to disc with the process id in the file name to avoid problems with concurrent access. For each process, the process id, the SNP name, the SNP number, the time consumption of testing a SNP and a time estimate for the complete analysis are continuously written to *stdout* (the terminal) when tests are being performed by PyBayenv (see Figure 5.6). The Bayenv test command for each process is written to file in order to document the analysis in accordance with specification in Section 4.9.9 (see Figure 5.7).

```
PyBayenv: process 2 is processing MOEN-rs119054541-2 (7042)... done. 10.771186 sec to complete. Estimated time remaining: 317.211424 minutes
PyBayenv: process 24 is processing MOEN-rs119054544-24 (7043)... done. 10.784945 sec to complete. Estimated time remaining: 317.616631 minutes
PyBayenv: process 15 is processing MOEN-rs119054544-15 (7043)... done. 10.787612 sec to complete. Estimated time remaining: 317.695172 minutes
PyBayenv: process 0 is processing MOEN-rs119054544-0 (7043)... done. 10.775060 sec to complete. Estimated time remaining: 317.325515 minutes
PyBayenv: process 10 is processing MOEN-rs119054544-10 (7043)... done. 10.781394 sec to complete. Estimated time remaining: 317.512053 minutes
PyBayenv: process 8 is processing MOEN-rs119054544-8 (7043)... done. 10.808893 sec to complete. Estimated time remaining: 318.321898 minutes
PyBayenv: process 4 is processing MOEN-rs119054544-4 (7043)... done. 10.754733 sec to complete. Estimated time remaining: 316.726889 minutes
PyBayenv: process 27 is processing MOEN-rs119054544-27 (7043)... done. 10.803982 sec to complete. Estimated time remaining: 318.177263 minutes
PyBayenv: process 22 is processing MOEN-rs119054544-22 (7043)... done. 10.777980 sec to complete. Estimated time remaining: 317.411507 minutes
PyBayenv: process 25 is processing MOEN-rs119054544-25 (7043)... done. 10.771366 sec to complete. Estimated time remaining: 317.216725 minutes
PyBayenv: process 21 is processing MOEN-rs119054544-21 (7043)... done. 10.785242 sec to complete. Estimated time remaining: 317.625372 minutes
PyBayenv: process 19 is processing MOEN-rs119054544-19 (7043)... done. 10.741812 sec to complete. Estimated time remaining: 316.346363 minutes
PyBayenv: process 23 is processing MOEN-rs119054544-23 (7043)... done. 10.776150 sec to complete. Estimated time remaining: 317.357617 minutes
PyBayenv: process 3 is processing MOEN-rs119054544-3 (7043)... done. 10.762810 sec to complete. Estimated time remaining: 316.964754 minutes
PyBayenv: process 18 is processing MOEN-rs119054544-18 (7043)... done. 10.767147 sec to complete. Estimated time remaining: 317.092481 minutes
PyBayenv: process 12 is processing MOEN-rs119054544-12 (7043)... done. 10.778491 sec to complete. Estimated time remaining: 317.426561 minutes
```

Figure 5.6: A terminal snapshot from the progress of PyBayenv when carrying out 32 Bayenv analyses in parallel on the *Cod* dataset. Every line contains information about the process number, the SNP being processed, the SNP number, the time it took to test the SNP and a time estimate for how long it would take to test the remaining SNPs.

5.1.6 Reducing the test set based on maximum allele frequency difference

As an additional measure for further reducing the time spent in the test phase of the Bayenv analysis, we implemented the method described in Section 4.3 in PyBayenv (see Section 4.3). The hypothesis was that we could exclude SNPs with low maximum allele frequency difference

```
Test 1:
bayenv2 -i %s -m mean_covar.txt -e std_environ_cod2_4group.env -p 4 -k 500000 -n 6 -t -r 98201 -o %s
Test 2:
bayenv2 -i %s -m mean_covar.txt -e std_environ_cod2_4group.env -p 4 -k 500000 -n 6 -t -r 45320 -o %s
Test 3:
bayenv2 -i %s -m mean_covar.txt -e std_environ_cod2_4group.env -p 4 -k 500000 -n 6 -t -r 39267 -o %s
Test 4:
bayenv2 -i %s -m mean_covar.txt -e std_environ_cod2_4group.env -p 4 -k 500000 -n 6 -t -r 95820 -o %s
Test 5:
bayenv2 -i %s -m mean_covar.txt -e std_environ_cod2_4group.env -p 4 -k 500000 -n 6 -t -r 17573 -o %s
Test 6:
bayenv2 -i %s -m mean_covar.txt -e std_environ_cod2_4group.env -p 4 -k 500000 -n 6 -t -r 85471 -o %s
Test 7:
bayenv2 -i %s -m mean_covar.txt -e std_environ_cod2_4group.env -p 4 -k 500000 -n 6 -t -r 97294 -o %s
Test 8:
bayenv2 -i %s -m mean_covar.txt -e std_environ_cod2_4group.env -p 4 -k 500000 -n 6 -t -r 31567 -o %s
Test 9:
bayenv2 -i %s -m mean_covar.txt -e std_environ_cod2_4group.env -p 4 -k 500000 -n 6 -t -r 2079 -o %s
Test 10:
bayenv2 -i %s -m mean_covar.txt -e std_environ_cod2_4group.env -p 4 -k 500000 -n 6 -t -r 96237 -o %s
Test 11:
bayenv2 -i %s -m mean_covar.txt -e std_environ_cod2_4group.env -p 4 -k 500000 -n 6 -t -r 84656 -o %s
Test 12:
bayenv2 -i %s -m mean_covar.txt -e std_environ_cod2_4group.env -p 4 -k 500000 -n 6 -t -r 39234 -o %s
```

Figure 5.7: Documentation of the BAYENV commands used in the test phase for the first 12 processes in PYBAYENV . The arguments "-i %s" and "-o %s" is replaced by the SNP name and the result (output) file respectively. For the other arguments, the user is referred to the BAYENV manual http://www.eve.ucdavis.edu/gmcoop/Software/Bayenv/bayenv_manual.pdf

(MAFD) across population from testing. To accomplish this, PYBAYENV calculates the MAFD for each SNP using equation 4.6 (page 53). PYBAYENV uses the *Python* package *stats* from the *SciPy* library (Jones, Oliphant, Peterson et al. 2001–) to compute the user specified empirical tails of SNPs with largest MAFD.

5.1.7 Defining a set of significant SNPs based on SDM

According to the specifications in Section 4.9.10 the SDM algorithm for interpreting the BAYENV results (see Section 4.2) was implemented in PYBAYENV. After the test phase is finished (see Section 5.1.4), the result files are read in by PYBAYENV and parsed according to equation 4.1 (page 50). The function *diff()* in the *NumPy* library (Jones, Oliphant, Peterson et al. 2001–) is used to determine the second order central difference (see Section 2.4.1) for the distribution of BFs. Moreover, the union sets are obtained by employing equation 4.3 (page 52). PYBAYENV writes the results from SDM to the terminal window (see Figure 5.8). Additionally, PYBAYENV writes the following statistics to disc as separate files for further analysis: 1) SNP name and BF for all significance sets determined by SDM for all runs and variables. 2) SNP name and average/median BF across runs for the significance sets determined by SDM for all variables. 3) The second

difference for all runs and variables. 4) The union sets for each variable containing SNP name, a binary indicator for which run the SNP has been identified and the total number of runs the SNP has been identified in.

5.2 Results from testing the time consumption using PyBayenv in parallel mode

To investigate whether we could save time by parallelizing the Bayenv process, we ran four analyses on the *Cod* dataset using PyBayenv as described in Section 4.10. The first analysis was carried out using a single run where its time consumption served as a benchmark. The three other analyses consisted of 8, 16, 32 replicate runs of Bayenv carried out in parallel by PyBayenv. Figure 5.9 show the total estimated time consumption for the four analyses and indicate the trend for the same number of runs carried out in serial.

Testing one SNP in the single run case took on average 3.1 second to complete using 100,000 MCMC iterations. An estimate of the total time consumption testing all 8809 SNPs in the dataset is therefore 455.1 minutes.

By testing each SNP eight times in parallel, the total time consumption was on average 3.7 seconds. An estimate for the completed dataset tested eight times is 543.2 minutes. Thus, the overhead of running eight processes instead of one is approximately 19.3%. Increasing the number of parallel runs to 16, lead to a doubling of the time consumption (estimated 1067.4 minutes in total) compared to eight runs. By increasing the parallel runs to 32 we saw another doubling (2099.5 minutes in total) compared to 16 parallel runs. The doubling of the time consumption seen going from eight to 16 and 16 to 32 parallel runs is a consequence of limited number of cores in the CPU (the CPU had eight cores).

A

*** var 1 test 1 ***
alpha = 10
epsilon = 0.5
N_alpha: 24
A: 359.64
delta_hat: 4.02763773074
Second difference = 6.109
Lowest sign BF: 43.136
Highest not sign BF: 35.181
Excluded SNPs = 8803
Total significant snps for var 1 test 1 is 6

B

*** var 2 test 1 ***
alpha = 10
epsilon = 0.5
N_alpha: 445
A: 100400.0
delta_hat: 13.7463915506
Second difference = 17.22
Lowest sign BF: 311.69
Highest not sign BF: 288.99
Excluded SNPs = 8726
Total significant snps for var 2 test 1 is 83

C

*** var 3 test 1 ***
alpha = 10
epsilon = 0.5
N_alpha: 166
A: 57285.0
delta_hat: 11.063365124
Second difference = 12.63
Lowest sign BF: 182.44
Highest not sign BF: 164.76
Excluded SNPs = 8798
Total significant snps for var 3 test 1 is 11

D

*** var 4 test 1 ***
alpha = 10
epsilon = 0.5
N_alpha: 55
A: 4148.6
delta_hat: 6.79646089453
Second difference = 7.389
Lowest sign BF: 41.203
Highest not sign BF: 33.431
Excluded SNPs = 8804
Total significant snps for var 4 test 1 is 5

E

*** var 5 test 1 ***
alpha = 10
epsilon = 0.5
N_alpha: 4
A: 153.51
delta_hat: 1.8161854256
Second difference = 3.4141
Lowest sign BF: 12.565
Highest not sign BF: 8.8762
Excluded SNPs = 8805
Total significant snps for var 5 test 1 is 4

F

*** var 6 test 1 ***
alpha = 10
epsilon = 0.5
N_alpha: 470
A: 93918.0
delta_hat: 13.7876715117
Second difference = 15.32
Lowest sign BF: 273.4
Highest not sign BF: 252.49
Excluded SNPs = 8720
Total significant snps for var 6 test 1 is 89

G

Total significant SNPs for var 1 is 6
Total significant SNPs for var 2 is 83
Total significant SNPs for var 3 is 11
Total significant SNPs for var 4 is 5
Total significant SNPs for var 5 is 4
Total significant SNPs for var 6 is 89

Figure 5.8: PyBayenv output after parsing the results from a single run of Bayenv using SDM on the *Cod* dataset. For each environmental variable PyBayenv prints the constants and variables involved in equation 4.2 (page 51), the second difference used as cutoff, lowest significant and highest non-significant BF value, number of non-significant SNPs and the number of significant SNPs. Finally, PyBayenv writes the number of SNPs in the union sets for each variable. Summary statistics for A) *sal1*, B) *sal2*, C) *temp1*, D) *temp2*, E) *ox1*, F) *ox2*. G) The number of SNPs in the union sets for all tested variables. Notice that the statistics is from a single run, the union sets are equal to the significance sets.

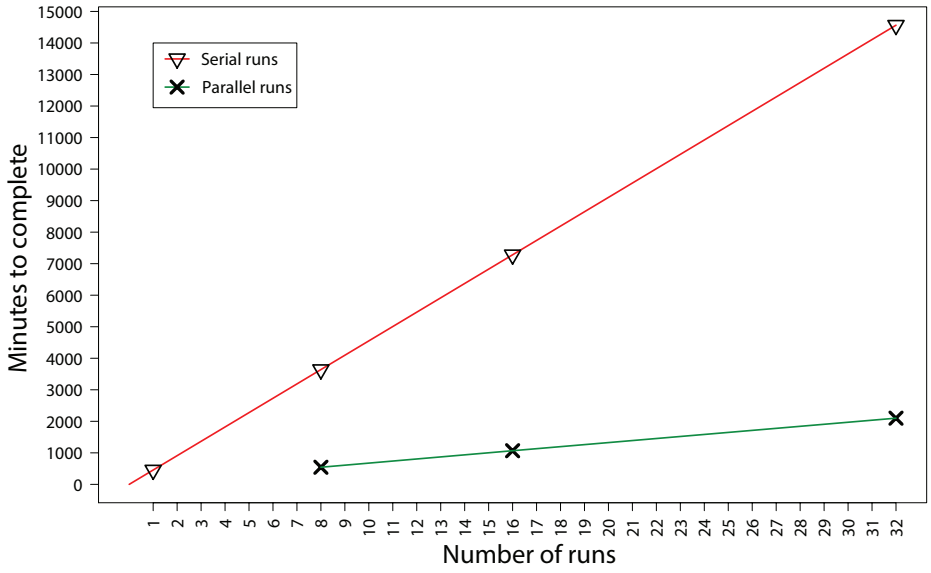

Figure 5.9: Total estimated time consumption of PyBayenv testing 8809 SNPs for environmental correlation. The red line indicates the time consumption used when each run is carried out in serial. The green line indicates the time consumption when the runs are carried out in parallel using a computer with eight cores.

5.3 Results from testing the convergence of the covariance matrix

To investigate whether a higher number of MCMC iterations and calculation of an average matrix would lead to improved consistency of the covariance matrix, we calculated the absolute value of the difference between six test matrices and a reference matrix according to the method in Section 4.5.

The heatmaps in Figure 5.10 shows that the difference is highest for the covariance matrices obtained after 10,000 MCMC iterations than for the other estimates. The average difference for these were 2.1×10^{-4} and 3.5×10^{-4} for the single draw and the average matrix respectively. The matrix estimates based on 100,000 MCMC iterations showed an average difference of 1.1×10^{-4} and 3.8×10^{-5} for the single draw and the average matrix respectively. The smallest difference was found between the average matrix obtained after 500,000 MCMC iterations and the reference (1.3×10^{-5}). The single draw found after 500,000 MCMC iterations did, however, have an higher average difference (9.4×10^{-5}) than the average matrix obtained after 100,000 MCMC iterations (3.8×10^{-5}).

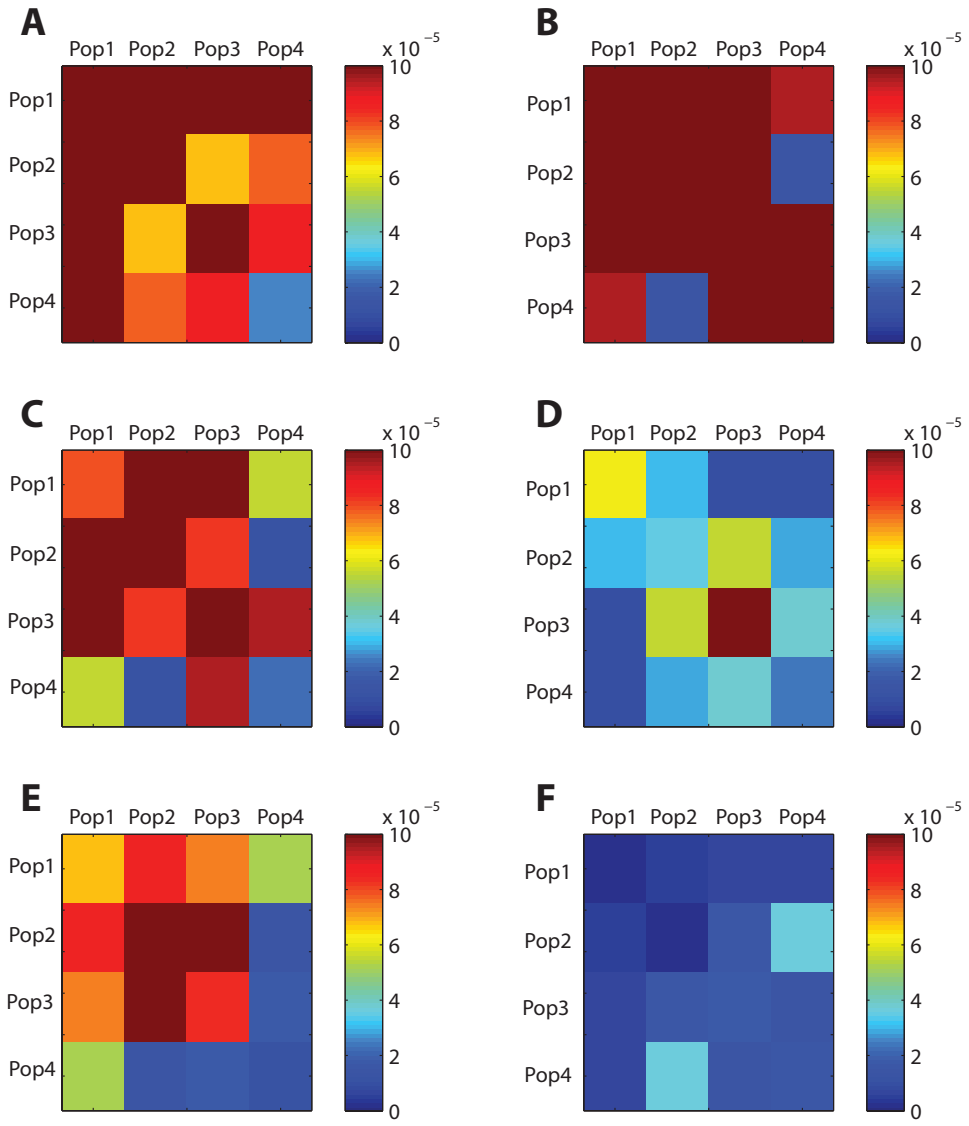

Figure 5.10: Heatmaps of the absolute value of the difference between test covariance matrix estimates and the reference (average matrix calculated after 500,000 MCMC iterations). A) Single estimate after 10,000 MCMC iterations. B) Average matrix after 10,000 MCMC iterations. C) Single estimate after 100,000 MCMC iterations. D) Average matrix after 100,000 MCMC iterations. E) Single estimate after 500,000 MCMC iterations. F) Average matrix after 500,000 MCMC iterations.

5.4 Results from the tests of the SDM

When using BAYENV to perform a genome-wide scan for correlation between population allele frequencies and environmental variables, there is a need to define a significance threshold for the results (BFs). As we have seen in Section 3.6, conventional cutoff thresholds such as a static (*Jefferey's table*, see Table 2.2) or percentage cutoff does not take the distribution of the results into account and hence may potentially lead to many FPs. To address this problem, we proposed a method named the second difference method (SDM). The idea behind the SDM was to use second difference to automatically define a significance threshold base on the shape of the BF distribution. The method use the second difference of a sorted distribution of BAYENV results (BFs) to determine the cutoff (see Section 4.2.1). In the subsequent sections we present the results from the experiments on the SDM described in Section 4.4.

5.4.1 Example on how the SDM algorithm works

The key factor for the SDM to work as intended, is to define a suitable cutoff for the second difference (see Section 4.2). We suggested an equation where the cutoff, $\hat{\delta}$, was determined by two important measures from the distribution: the largest BF (A) value and the number of SNPs displaying a BF above 10 (N_α). Additionally, the equation consisted of a small constant factor ϵ ($0 < \epsilon \leq 1$), which purpose was to make sure that $\hat{\delta} < 0$ and to "fine tune" the SDM algorithm (see Section 4.2.2). To clarify how the SDM works, we here provide an hypothetical example:

Consider that 10,000 SNPs are tested, 100 SNPs achieve a BF>10 (N_α) and the largest BF in the distribution is 1,000 (A). The constant factor ϵ is defined to be 0.5 (default value). Then $\hat{\delta}$ is calculated as follows:

$$\begin{aligned}\hat{\delta} &= \epsilon + \log_{10} N_\alpha \log_{10} A \\ &= 0.5 + \log_{10}(100) \log_{10}(1,000) \\ &= 6.5\end{aligned}$$

Thus the cutoff on the second difference distribution is 6.5. Next, consider that the last 29 to 20 values in the increasingly sorted distribution of BFs ($y_{9972}, \ldots, y_{9981}$, see Section 4.2.1) are as follows: $[44, 45, 46, 47, 48, 49, 50, 65, 68, 74]$. Then the corresponding second difference (central difference, see Section 2.4.1) is: $[0, 0, 0, 0, 0, 14, -12, 3]$. The three consecutive BF values $[y_{9977}, y_{9978}, y_{9979}] = [49, 50, 65]$ have a central difference $\Delta y_{9978} = 14$ which is greater than $\hat{\delta} = 6.5$. Hence, this is used by the SDM as the cutoff. It follows from the definition in Section 4.2.1 that the SNPs with a BF value greater or equal to 65 are included in the significance

set (ω). Thus a total of 22 SNPs is declared significant for this distribution of results. Note that in this example it is assumed that the 9971 SNPs with a BF < 44 have an associated second difference $\Delta y \leq 6.5$ and hence will not trigger an earlier cutoff.

5.4.2 Results from testing the SDM on simulated BF values

In order to demonstrate and verify the SDM algorithm, we simulated three sets of artificial BF results simulating three different and possible outcome scenarios from a BAYENV analysis. The sets, *Sim-weak*, *Sim-strong* and *Sim-large*, were simulated according to the method in Section 4.6.1. The SDM part of PYBAYENV (see Section 4.9.10 and 5.1.7) was used to apply the algorithm on the three simulated sets. Figure 5.11 visualize how the simulated sets and the corresponding second difference are distributed. The PYBAYENV output in Figure 5.12, summarises the results after using the SDM function in PYBAYENV on the three sets.

On the basis of the data in *Sim-weak*, SDM defined $\hat{\delta}$ to be 2.3. The second difference in the data that triggered the cutoff was 48.7. The lowest BF value declared as significant was 58.4 and the highest non-significant BF value was 9.8. The total number of significant "SNPs" was five. Moreover, all the "SNPs" in the *siSNPs* group were declared significant, whereas none were declared significant from the *neSNPs* or the *noSNPs* group (see Figure 5.12 A for details).

On the basis of the data in *Sim-strong*, SDM defined $\hat{\delta}$ to be 13.1. The second difference in the data that triggered the cutoff was 365.0 The lowest BF value declared as significant was 665.5 and the highest non-significant BF value was 299.9. The total number of significant "SNPs" was 150. Moreover, all the "SNPs" in the *siSNPs* group were declared significant, whereas none were declared significant from the *neSNPs* or the *noSNPs* group (see Figure 5.12 B for details).

On the basis of the data in *Sim-large*, SDM defined $\hat{\delta}$ to be 23.6. The second difference in the data that triggered the cutoff was 190,955.6 The lowest BF value declared as significant was 191,955.2 and the highest non-significant BF value was 999.5. The total number of significant "SNPs" was 20. Moreover, all the "SNPs" in the *siSNPs* group were declared significant, whereas none were declared significant from the *neSNPs* or the *noSNPs* group (see Figure 5.12 C for details).

5.4.3 Results from the tests on the *Cod* dataset

In the two subsequent sections we present the results from using the SDM to interpret the BAYENV analyses on the *Cod* dataset. Manhattan plots (see Section 4.4.2) of the $\log_{10}$ (median BF after 32 independent runs of BAYENV)

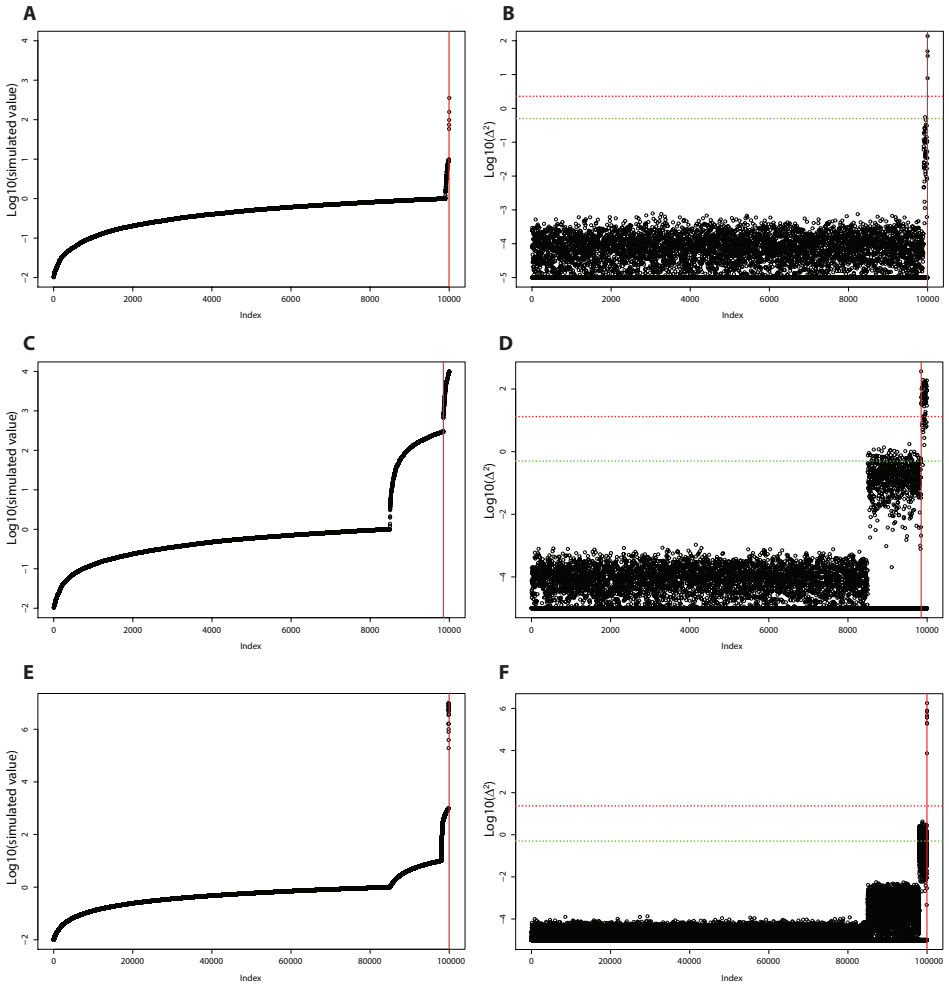

Figure 5.11: BF values and corresponding second difference (Δ^2) for the three simulated datasets. A) and B) Logarithmic transformed BF and second differences values for the *Sim-weak* dataset C) and D) Logarithmic transformed BF and second differences values for the *Sim-strong* dataset. E) and F) Logarithmic transformed BF and second differences values for the *Sim-large* dataset. The red vertical lines indicate where the cutoff is being made by SDM. The red dotted horizontal lines (B and D) indicate the second difference threshold $\hat{\delta}$. The green dotted horizontal lines (B and D) indicate a second difference of 0.5 (the default value of ϵ).

A

*** var 1 test 1 ***
alpha = 10
epsilon = 0.5
N_alpha: 5
A: 354.144290551
delta_hat: 2.2818005266
Second difference = 48.5690498815
Lowest sign BF: 58.3991160337
Highest not sign BF: 9.81214514654
Excluded SNPs = 9995
Total significant snps for var 1 test 1 is 5
_____
Total significant SNPs for var 1 is 5

B

*** var 1 test 1 ***
alpha = 10
epsilon = 0.5
N_alpha: 1456
A: 9977.04333067
delta_hat: 13.1494882169
Second difference = 365.088171532
Lowest sign BF: 665.476977942
Highest not sign BF: 299.898631754
Excluded SNPs = 9850
Total significant snps for var 1 test 1 is 150
_____
Total significant SNPs for var 1 is 150

C

*** var 1 test 1 ***
alpha = 10
epsilon = 0.5
N_alpha: 2000
A: 9718892.45935
delta_hat: 23.5663325779
Second difference = 190955.644349
Lowest sign BF: 191955.240183
Highest not sign BF: 999.486349975
Excluded SNPs = 99980
Total significant snps for var 1 test 1 is 20
_____
Total significant SNPs for var 1 is 20

Figure 5.12: PyBayenv output summarising the results from and the factors used by the SDM algorithm. A) The results on the *Sim-weak* dataset. B) The results on the *Sim-strong* dataset. C) The results on the *Sim-large* dataset.

were plotted to illustrate how the results are distributed across the Cod genome (see Figure 5.13). Among the SNPs reported by BAYENV to have a $\log_{10}$ BF > 2 were SNPs found within or closely located to genes associated with osmoregulation, as well as genes known to play important roles in the hydration and development of oocytes (see Berg et al. 2015, *in review* for a complete analysis of these BAYENV results from a biological perspective).

5.4.4 Testing the SDM on a single BAYENV run on the *Cod* dataset

To explore how the SDM performed on real data, we applied the method to the results from a single run of BAYENV on the *Cod* dataset (see Methods Section 4.6.2). We used PYBAYENV to carry out the BAYENV analysis and the SDM implementation in PYBAYENV was used to parse the results and compute the significance sets (figure 5.8 in Section 5.1.7 shows the PYBAYENV output from this particular BAYENV analysis). q-values were calculated for the distribution of BFs for all environmental variables (*ox1*, *ox2 sal1*, *sal2*, *temp1* and *temp2*) in accordance with the method in Section 4.4. The proportion of FPs in each SDM set was compared to the significance sets found using the cutoff thresholds *alpha1*, *alpha5* and *jeff3.2*. Summary statistics for all environmental variables and cutoff methods are given as in table 5.1. Figure 5.14 illustrates the number of FPs and TPs in each set graphically.

In accordance with the method in Section 4.6.2, we plotted the second differences (Δ^2, see equation 4.1) for the ordered list of BFs. Figure 5.15 visualize the distribution of Δ^2 and corresponding logarithmic transformed BF values for the environmental variables *sal1* and *sal2*.

For *sal1*, SDM defined the cutoff threshold, $\hat{\delta}$, to be 4.0. The second difference in the data that triggered the cutoff was 6.1. The lowest BF value declared as significant was 43.1 and the highest non-significant BF value was 35.2. The total number in the significance set (ω, see Section 4.2.1) were six (see Figure 5.8 A for details).

For *sal2*, SDM defined $\hat{\delta}$ to be 13.7. The second difference in the data that triggered the cutoff was 17.2. The lowest BF value declared as significant was 311.7 and the highest non-significant BF value was 289.0. The total number in the significance set were 83 (see Figure 5.8 B for details).

For corresponding statistics the other variables (*temp1 temp2*, *ox1* and *ox2*), please see Figure 5.8 C, D, E, F.

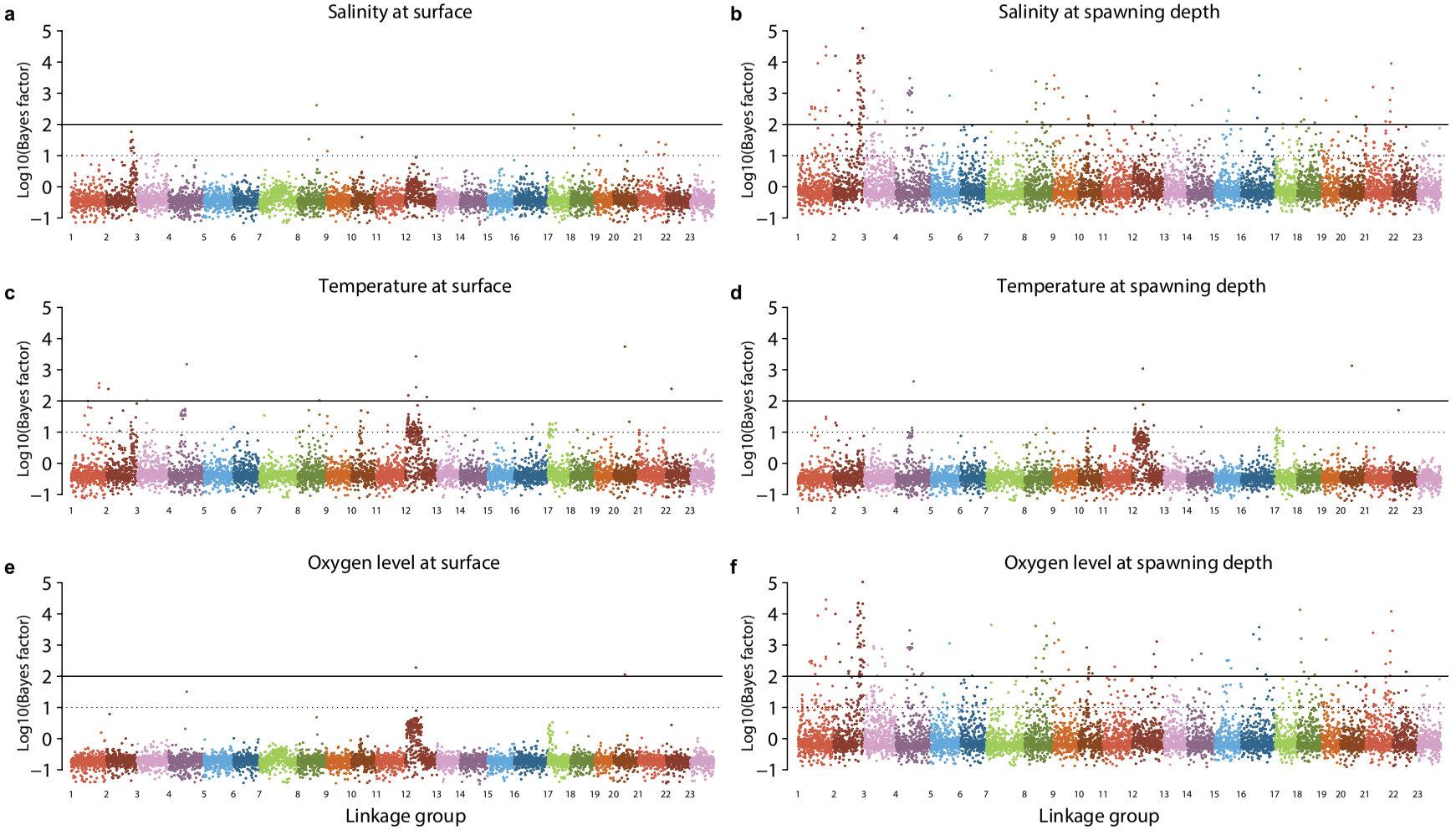

Figure 5.13: Manhattan plot of SNP association with the environmental variables water, temperature, oxygen - all at surface and spawning depth. The plots are based on the median $\log_{10}$ Bayes factor calculated from 32 independent runs of BAYENV. The SNPs are ordered by linkage group and sorted according to the position in each linkage group. The solid line at 2 and dotted line at 1 ("decisive" and "strong" association from Jefferey's table [2.2] respectively) illustrates how the distribution of Bayes factor differs between environmental variables.

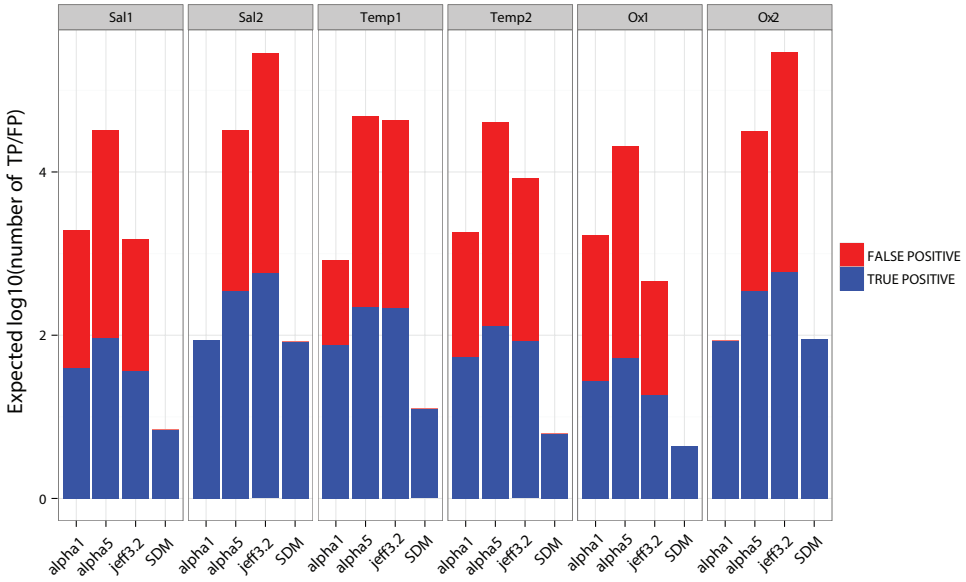

Figure 5.14: Expected number of TP/FP in the significance sets defined by the cutoffs *alpha1*, *alpha5*, *jeff3.2* and SDM. Expected proportions of FPs below one percent are plotted as zero.

The significance sets obtained using the four different cutoff methods for the variable *sal1* shows that SDM appears to be more conservative than its cutoff level counterparts (see Table 5.1 A). Six SNPs are included in the SDM significance set, whereas for *alpha1*, *alpha5* and *jeff3.2* the corresponding numbers are 88, 440 and 78, respectively (note that *alpha1* and *alpha2* will always contain 88 and 440 SNPs respectively). The maximum q-value for SDM is 0.089, implying that we can expect 0.53 FPs in this set on average. The maximum q-value for *alpha1*, *alpha5* and *jeff3.2* is in the interval 0.52-0.79 indicating a higher proportion of FPs in these sets. The minimum BF value in the SDM set is 43.14, whereas the same numbers for the other sets are between 1.10 and 3.25.

The pattern seen for the variable *sal1* is repeated for the variables *temp1*, *temp2* and *ox2*: There are few SNPs in the SDM set compared to the sets obtained from the other methods. A common feature for these variables is that the overall signal strength in terms of BF is relatively low. The maximum BF in the distribution might be high, but there are relatively few SNPs showing a BF value above 10.

For the variables *sal2* and *ox2*, SDM shows highly similar results to that of *alpha1* by defining 83 and 89 significant SNPs respectively. The expected proportion of FPs in these sets is considerably lower than what we have seen for the other variables. For example, for the variable *sal2* the maximum q-value for SDM is 0.009. With 83 SNPs in this set, we can expect 0.76 FPs among these results. As opposed to *sal1*, the overall signal strength given for *sal2* is much stronger. The minimum BF value for the 83 SNPs is as high as 311.69 indicating that many SNPs in the dataset are strongly correlated to this variable. A similar pattern is seen for the variable

ox2 using SDM: a low maximum q-value of 0.011 and a high minimum BF of 273.4 in the significance set.

All q-values for this experiment is available in SI table A.

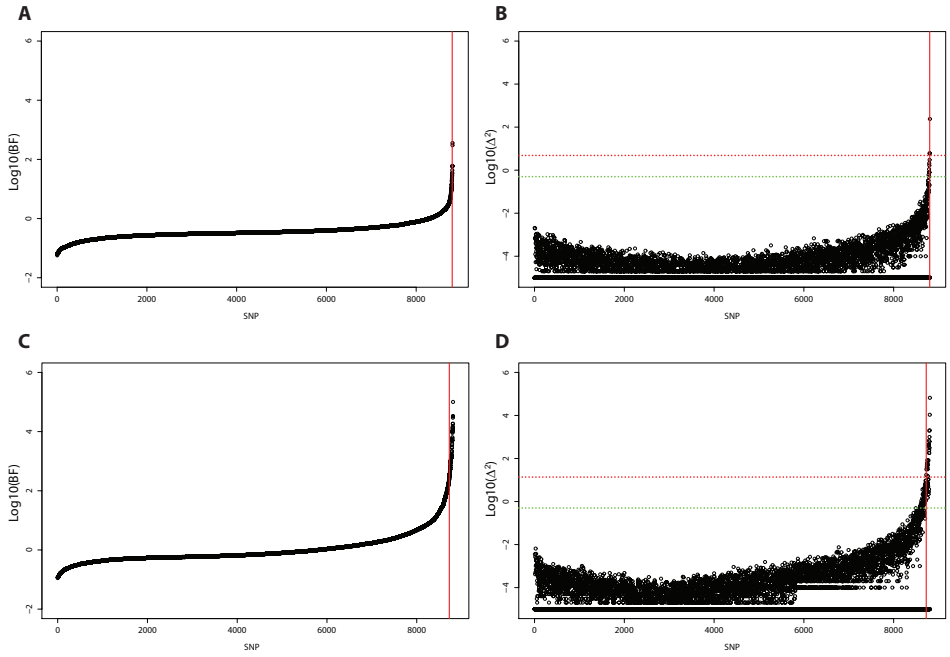

Figure 5.15: BF values and corresponding second difference (Δ^2) for the variables *sal1* and *sal2* on the *Cod* dataset. A) and B) Logarithmic transformed BF and second differences values for *sal1*. C) and D) Logarithmic transformed BF and second differences values for *sal2*. The vertical red line shows where the cutoff is being made in the distribution. The red vertical lines indicates where the cutoff is being made by SDM. The red dotted horizontal lines (B and D) indicates the second difference threshold $\hat{\delta}$. The green dotted horizontal lines (B and D) indicates a second difference of 0.5 (the default value of ϵ).

5.4.5 Testing the SDM on multiple BAYENV runs on the *Cod* dataset

One particular issue with BAYENV is the potential run-to-run variability (Blair, Granka and Feldman 2014). To explore how the SDM handles this problem, we examined the results from 32 independent runs of the *Cod* dataset carried out in parallel by PYBAYENV. Correlation between allele frequency and the environmental variables, salinity (*sal1* and *sal2*), temperature (*temp1* and *temp2*), and oxygen (*ox1* and *ox2*), all at surface and spawning depth were tested. The SDM function in PYBAYENV was used to interpret and summarise the results across the independent runs. The Manhattan plot in Figure 5.13 visualise how the results (median

A **Statistics for Salinity surface (sal1)**

Method	Max BF	Min BF	Max q-value	Sign. SNPs	Expected # FP
SDM	359.64	43.136	0.089	6	0.534
alpha1	359.64	2.942	0.548	88	48.246
alpha5	359.64	1.096	0.786	440	345.936
jeff3.2	359.64	3.248	0.523	78	40.967

B **Statistics for Salinity deep (sal2)**

Method	Max BF	Min BF	Max q-value	Sign. SNPs	Expected # FP
SDM	100400	311.690	0.009	83	0.764
alpha1	100400	248.920	0.010	88	0.905
alpha5	100400	10.209	0.208	440	91.539
jeff3.2	100400	3.203	0.456	1078	491.832

C **Statistics for Temperature surface (temp1)**

Method	Max BF	Min BF	Max q-value	Sign. SNPs	Expected # FP
SDM	57285	182.44	0.023	11	0.254
alpha1	57285	24.460	0.124	88	10,890
alpha5	57285	3.009	0,488	440	214.799
jeff3.2	57285	3.200	0.475	418	198.695

D **Statistics for Temperature deep (temp2)**

Method	Max BF	Min BF	Max q-value	Sign. SNPs	Expected # FP
SDM	4148.6	41.203	0.057	5	0.285
alpha1	4148.6	7.038	0.384	88	33.826
alpha5	4148.6	1.164	0.706	440	310.769
jeff3.2	4148.6	3.205	0.529	183	96.835

E **Statistics for Oxygen surface (ox1)**

Method	Max BF	Min BF	Max q-value	Sign. SNPs	Expected # FP
SDM	153.51	12.565	0.147	4	0.586
alpha1	153.51	1.879	0.681	88	59.887
alpha5	153.51	0.399	0,878	440	386.396
jeff3.2	153.51	3.300	0.570	43	24.523

F **Statistics for Oxygen deep (ox2)**

Method	Max BF	Min BF	Max q-value	Sign. SNPs	Expected # FP
SDM	93918	273.400	0.0107	89	0.952
alpha1	93918	274.820	0.0104	88	0.899
alpha5	93918	10.981	0.204	440	89.575
jeff3.2	93918	3.202	0.454	1088	493.902

Table 5.1: Statistics from using the cutoff methods SDM, *alpha1*, *alpha5* and *jeff3.2* on the results from a single BAYENV run carried out on the *Cod* dataset. For each environmental variable and cutoff method, the following statistics is provided: The maximum BF (**Max BF**). The minimum BF using the specified cutoff method (**Min BF**). The maximum q-value in the significance set (**Max q-value**). The number of SNPs in the specified significance set (**Sign. SNPs**). The expected number of FPs in the specified significance set (**Expected #** FP).

$\log_{10}$BF) from this experiment are distributed across the Cod genome. In this experiment we compared the SDM to the cutoff methods *alpha1*, *alpha5*, *jeff3.2* and *jeff10* (see Section 4.4). We tested the methods on all environmental variables and counted how many times each SNP was identified as significant. In order to be included in the Total Significance Set *TSS* (see Section 4.2.3), an SNP needed to be present in at least 70% of the runs ($\kappa = 0.7$; see equation 4.4). Result statistics for the six variables are summarized in Table 5.2. To visualise how the significant SNPs are distributed among the different cutoff methods, we plotted Venn diagrams of the union sets and TSS's for the variables *sal1* and *sal2* (see Figure 5.16). The Venn diagrams were produced according to the methods in Section 4.6.3.

The results for *sal1* show that there is a high variability between runs for this variable using all cutoff methods. Out of a total of 84 SNPs identified using SDM, only 4.8% were identified as significant in 22 (70%) or more runs. This variability is also evident using the other cutoff methods. For example *jeff10* resulted in a total of 773 SNPs identified in one or more of the 32 runs, but only 2.2% of the SNPs were identified in 22 or more runs. The highest percentage of consistent SNPs (TSS SNPs) was obtained using *alpha5* with 11.1% of 2072 SNPs identified. However, we note that almost one fourth of all (8809) SNPs were identified as significant once or more using this cutoff method.

In terms of FDR for *sal1*, SDM shows the lowest maximum q-value. The q-value of 0.056 indicates that there is on average 0.22 FPs among the five significant SNPs identified by SDM. On the other end of the scale we find *alpha5* with a FDR of 0.747, indicating that 171.7 out of 230 SNPs are false discoveries.

Like in the previous experiment where we examined one single run of BAYENV, the pattern seen for *sal1* is reflected in the variables *temp1*, *temp2* and *ox2*. All these variables exhibits relatively few SNPs with a high BF signal as indicated by the minimum median BF statistics (figure 5.2). The overall run-to-run variability for these variables was also high: The percentage SNPs identified in more than 70% of the runs was in the interval 0.95% (*jeff10* on *ox1*) and 23.8% (*alpha5* on *temp1*).

The five SNPs identified for *sal1* using SDM were all included in the corresponding set of eight SNPs identified in previous experiment (figure 5.1). The same statistics for *temp1*, *temp2* and *ox2* did however not show the same consistency: here the similarities was 5 out of 13, 2 out of 7 and 2 out of 6, respectively.

The variables *sal2* and *ox2* show more consistency in general. The percentage of SNPs that are identified in more than 70% of the runs are in the interval 55.1% (*jeff10* on *sal2*) and 76.8% (*alpha1* on *ox2*). The number of SNPs identified by SDM is also significantly higher for these variables. A total of 86 SNPs were identified in 22 or more runs for *sal2*, whereas the

same number for *ox2* was 77. As for the previous experiment, the results from the SDM on these variables are highly similar to that obtained using *alpha1*.

In terms of FDR, the maximum q-value for *sal2* and *ox2* (both approximately 0.01) indicates that there is a relatively low number of false discoveries among the SNPs that are deemed significant using SDM.

The TSS from SDM on *sal2* and *ox2* is highly similar to the corresponding significance set obtained using SDM on a single run of BAYENV. For *sal2* the intersect of the two sets were 83, whereas the two sets had 77 SNPs in common for *ox2*.

Results for Salinity surface (sal1)

Method	Max med. BF	Min med. BF	Max q-value TSS	SNPs in TSS	Exp. # FP	Union SNPs
SDM	413.96	57.48	0.056	4	0.224	84
Alpha1	413.96	4.90	0.431	46	19.809	557
Alpha5	413.96	1.39	0.747	230	171.708	2072
Jeff10	413.96	17.69	0.212	17	3.601	773
Jeff3.2	413.96	4.23	0.459	54	24.772	2226

Results for Salinity spawning depth (sal2)

Method	Max med. BF	Min med. BF	Max q-value TSS	SNPs in TSS	Exp. # FP	Union SNPs
SDM	121480	262.58	0.011	86	0.937	127
Alpha1	121480	268.47	0.011	85	0.907	121
Alpha5	121480	10.49	0.199	399	79.545	647
Jeff10	121480	10.6	0.197	405	79.696	734
jeff3.2	121480	3.34	0.448	983	440.752	1626

Results for Temperature surface (temp1)

Method	Max med. BF	Min med. BF	Max q-value TSS	SNPs in TSS	Exp. # FP	Union SNPs
SDM	5535.35	149.52	0.021	9	0.192	234
Alpha1	5535.35	22.62	0.151	43	6.488	381
Alpha5	5535.35	2.98	0.508	339	172.095	1427
Jeff10	5535.35	12.28	0.264	86	22.672	1478
Jeff3.2	5535.35	3.65	0.474	319	151.161	2937

Results for Temperature spawning depth (temp2)

Method	Max med. BF	Min med. BF	Max q-value TSS	SNPs in TSS	Exp. # FP	Union SNPs
SDM	1336.4	422.31	0.007	3	0.022	160
Alpha1	1336.4	9.38	0.354	33	11.683	382
Alpha5	1336.4	1.39	0.676	305	206.210	1892
Jeff10	1336.4	13.19	0.292	17	4.968	764
Jeff3.2	1336.4	3.88	0.538	163	87.753	1862

Results for Oxygen surface (ox1)

Method	Max med. BF	Min med. BF	Max q-value TSS	SNPs in TSS	Exp. # FP	Union SNPs
SDM	190.07	31.81	0.059	3	0.178	106
Alpha1	190.07	2.39	0.715	37	26.439	401
Alpha5	190.07	0.5	0.847	230	194.699	2617
Jeff10	190.07	31.81	0.059	3	0.178	315
Jeff3.2	190.07	4.52	0.396	7	2.771	955

Results for Oxygen spawning depth (ox2)

Method	Max med. BF	Min med. BF	Max q-value TSS	SNPs in TSS	Exp. # FP	Union SNPs
SDM	105945	321.35	0.008	77	0.645	123
Alpha1	105945	260.63	0.011	86	0.959	112
Alpha5	105945	10.77	0.209	415	86.786	584
Jeff10	105945	10.35	0.217	444	96.563	629
Jeff3.2	105945	3.3	0.443	1002	443.588	1401

Table 5.2: Summary statistics from using the cutoff methods SDM, *alpha1*, *alpha5 jeff10* and *jeff3.2* on the results from 32 BAYENV runs on the *Cod* dataset. A cutoff $\kappa = 0.7$ is used to obtain the TSS's. For each environmental variable and cutoff method, we provide the following statistics: The maximum median BF (**Max med. BF**). The minimum median BF (**Min med. BF**). The maximum q-value in the specific TSS (**Max q-value TSS**). The number of SNPs in the specific TSS (**SNPs in TSS**). The expected number of FPs in the specified TSS (**Expected #** FP). The number of SNPs found in the specified union sets (**Union SNPs**)

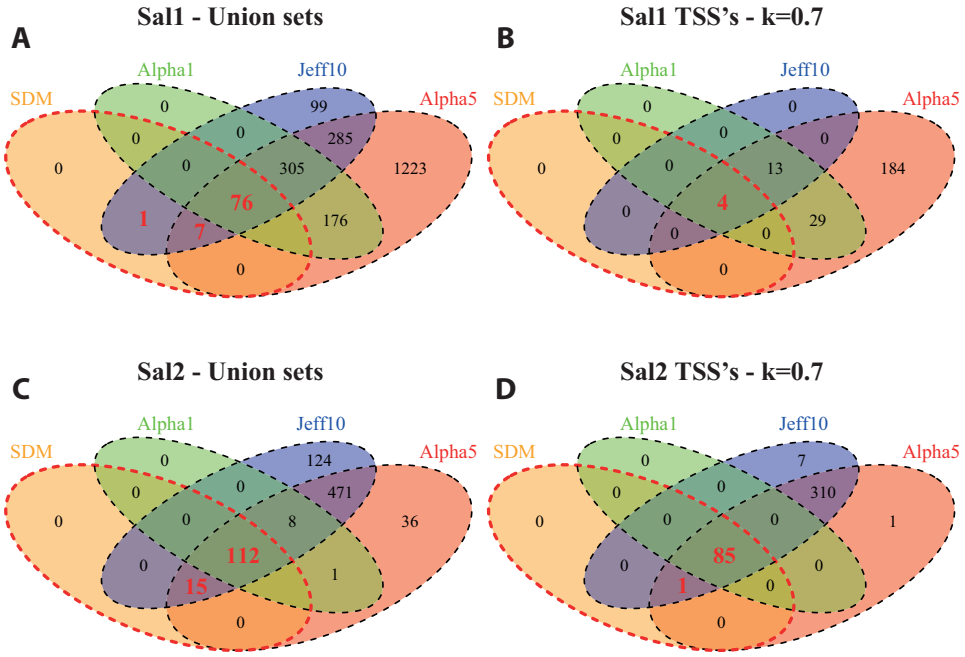

Figure 5.16: Venn diagrams of the distribution of significant SNPs using four different cutoff methods: SDM, *alpha1*, *alpha5* and *jeff10*. A) All SNPs identified as significantly associated with the variable *sal1* in any of the thirty-two runs. B) SNPs significantly associated with *sal1* in at least 70 percent of the runs ($\kappa = 0.7$). C) All SNPs identified as significantly associated with the variable *sal2* in any of the thirty-two runs. D) SNPs significantly associated with *sal2* in at least 70 percent of the runs ($\kappa = 0.7$). The SDM subset values are drawn in red and bold

5.4.6 Testing the SDM on the *Maize* dataset

We explored the abilities of the SDM by applying it to the results from a BAYENV analysis on the *Maize* dataset (see Section 4.1.2). In this experiment, 135 SNPs were tested for correlation to an ordinal environmental variable - the three stages in the local seed system of Staha (see Section 4.1.2). The 32 independent runs of BAYENV were carried out in parallel by PYBAYENV. Only presumed neutral SNPs were used for the null model (see Methods Section 4.6.4). The resulting average BF for each SNP was plotted as a Manhattan plot (see Figure 5.17 A)

Due to the low signal strength in terms of BF received from this dataset, we adjusted the ϵ component in the $\hat{\delta}$ equation (see equation 4.2) from 0.5 (default) to 0.2. In 28 out of the 32 runs, the cutoff threshold $\hat{\delta}$ was computed to be equal to ϵ (i.e. $N_\alpha \leq 1$). In the four other cases $\hat{\delta}$ was in the interval 0.85 and 1.36. For runs using $\hat{\delta} = \epsilon$ as cutoff threshold, the average number of SNPs in the significance set was 9.3, whereas for the runs where $\hat{\delta}$ were adjusted upwards the corresponding number was 5.0.

A total of 36 SNPs were declared as significant in one ore more runs using the SDM. The SNPs included in the union set were plotted in a significance-plot (see Figure 5.17B). Only three SNPs were consistent in more than 70% of the runs. However, these three showed the highest average BF and were all among the candidates for positive selection (based on F_{ST} values)

We converted the average BF for each SNP to q-values as a measure of FDR. The maximum q-value for the SNPs declared as significant by SDM was 0.56 indicating a high proportion of FPs among these results.

Applying an *alpha1*, *alpha5* or *jeff3.2* cutoff threshold on the average BF distribution yields one, seven and three significant SNPs respectively.

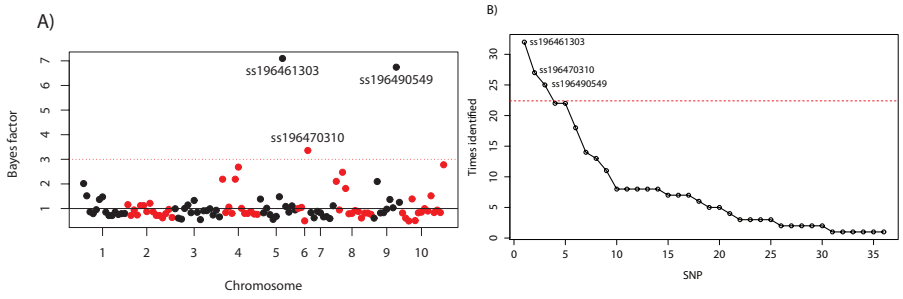

Figure 5.17: **A)** Manhattan plot of SNP association with seed system stage based on the average BF. The SNPs are plotted according to chromosome and position at chromosome along the X-axis. Chromosomes alter between black and red color. The red dotted line at 3 indicates a positive Bayes factor according to Kass and Raftery 1995 (table 2.3). Source: Westengen et al. 2014b **B)** Significance plot of the union set obtained by applying the SDM to the results from 32 independent runs of BAYENV. The red dotted line indicates significance with κ=0.7 (eq. 4.4)

5.5 Results from the tests on the stability of the BAYENV method

In this section we present the results from the tests on the stability of the BAYENV method. All test in this section are carried out using the *Cod* dataset.

5.5.1 The impact of increasing the number of MCMC iterations in the test phase of BAYENV

To assess whether the number of MCMC affected the run-to-run variability of BAYENV, we examined the results from eight different analyses of the *Cod* data set. Each analysis encompassed 32 independent runs of BAYENV using different number of MCMC iterations (see Section 4.7.1). We

calculated the stability score (see Section 4.2.4) for the union sets based on a *alpha1* cutoff and the SDM (see Figure 5.18).

The variables *sal2* and *ox2*, shows a clear upward trend as the number of MCMC iterations are increased for both cutoff methods. However, for the remaining four variables, the trend is negative meaning the variability increases with the number of MCMC iterations. Whereas the SDM shows a consistent upward trend for the variable *ox2*, the same variable seems to be peaking at 500,000 iterations by using a *alpha1* cutoff. The variable *sal2* shows a peak at 500,000 iterations in both methods. The second analysis of 500,000 iterations (500kb in Figure 5.18) shows very similar stability score to its 500,000 run counterpart and slightly more similar using the SDM. The discrepancy between the variables is more evident when the SDM is used.

5.5.2 Testing the relationship between run-to-run variability and the number of independent BAYENV runs

To investigate whether multiple independent runs of BAYENV would lead to more certainty about the result, we evaluated five subsets (1, 2, 4, 8, 16 independent runs) and the full set from a 32 run analysis and compared them to the median statistics from a second control analysis consisting of 32 independent runs (see Methods Section 4.7.2). The percentage of equal SNPs in the 99 percentile tail (88 SNPs) for each variable and subset is plotted in Figure 5.19.

Two of the variables, *sal2* and *ox2*, showed very high consistency by having almost identical empirical 99 percentile tails (*alpha1* cutoff) to the reference data set in all test sets. The four other variables (*sal1*, *temp1*, *temp2* and *ox1*) however, showed increasing agreement with the reference set as the number of runs increased. The variable *temp1* had the most significant gain by showing an increase in similarity to the reference set from 52.8 percent after one test run to 86.5 percent after 16 runs (same after 32 runs).

Not all variables displayed a steady increase in similarity to the reference set. For example, the variable *ox2* dropped almost 7 percent after increasing the number of runs from 8 to 16, and *sal1* dropped a couple of percent after 4 runs. The trend was however clear: all variables show the best or equal to best similarity to the reference set after 32 independent runs.

5.6 Reducing the test set based on the maximum allele frequency difference between populations

Testing many SNPs using BAYENV is a very time consuming task (see Section 3.6.2). In order to reduce the total time spent on the test phase of BAYENV, we proposed a method where only a subset of the available

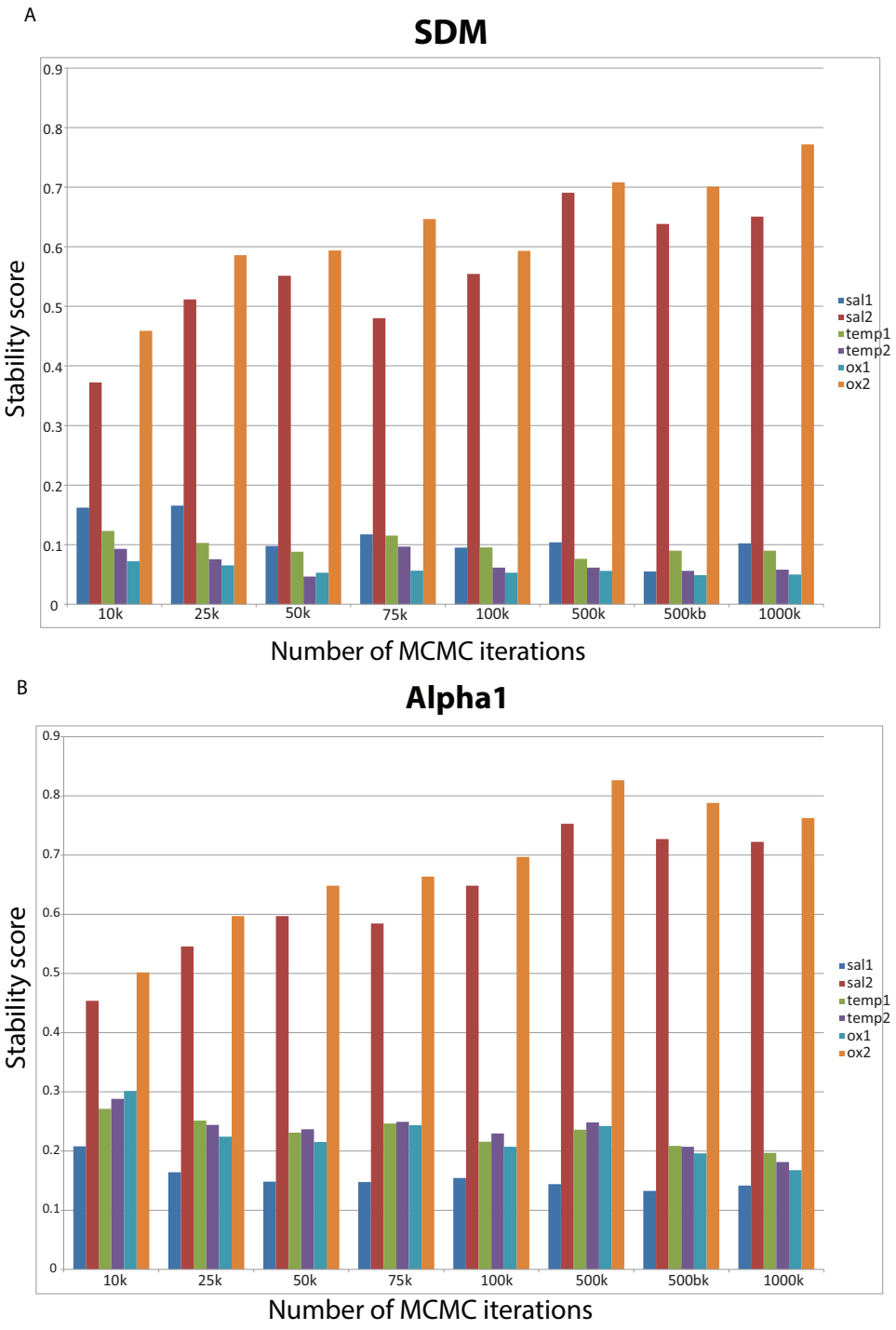

Figure 5.18: Barplots of the stability scores obtained from BAYENV analyses carried out using different number of MCMC iterations. The results were calculated based on 32 replicate runs for each test. A) Stability scores for the SDM cutoff. B) Stability scores for the *alpha1* cutoff. The run named 500kb is the second analysis carried out using 500,000 MCMC iterations

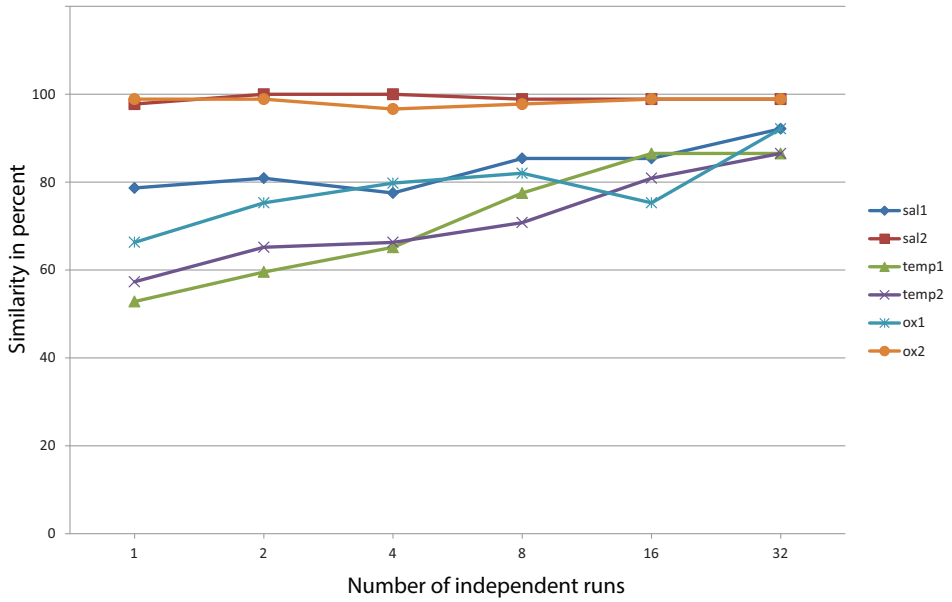

Figure 5.19: Plot of set difference between significance set based on a 99 percentile cutoff. The plot shows the similarity in percent between the median BF from the test sets and the reference set.

SNPs are tested (see Section 4.3). The method uses the maximum allele frequency difference (MAFD) between population to exclude SNPs that are less "interesting" from the test set. For this experiment we followed the method described in Section 4.8. We used PYBAYENV to calculate the MAFD and to carry out the tests on the reduced sets (see Section 5.1.6).

To make salient the relationship between the results ($\log_{10}$(BF)) and the MAFD between populations, these quantities were plotted for each environmental variable (see Figure 5.20). The significance for the spline regressions were $p < 0.001$ for all environmental variables. All SNPs except one (for *sal2*) that were declared significant using the SDM on the full dataset (see Section 5.4.5) were found among SNPs with MAFD among the top 90 percentile. We found that the four cutoffs, 90, 95, 97.5 and 99, resulted in MAFD cutoffs of 0.32, 0.39, 0.44 and 0.55 respectively. Figure (see Figure 5.21) shows the similarity in percent between the tests and reference sets.

The comparison of the TSS's obtained from SDM shows that the variables *temp2* and *ox1* has a 100% similarity to the reference on all cutoff levels. The variables *sal2* and *ox2* decreases from ca. 90% to ca. 60% as the cutoff becomes stricter, whereas the variables *temp1* and *sal1* shows an opposite trend going from ca. 50% to 75%. The comparison of the top 88 ranked SNPs (*alpha1* cutoff) shows that by excluding SNPs with MAFD outside the 90 percentile, almost 100% from the reference set is retained in the test set for the variables *sal2* and *ox2*. All variables show a similarity of 76.4% or more to the reference results using a 90 percentile cutoff. Whereas the

variables *sal2* and *ox2* shows a steady but modest decline moving from 90 percentile to 99 percentile cutoff, the drop is more dramatic for the other variables. The variable *ox1* show the most extreme drop moving from 79.7% similarity using a 90 percentile cutoff to 8.9% and 4.4% using a 97.5 and 99 percentile cutoff respectively.

By comparing the top 20 ranked SNPs, the result is very similar to what we found comparing the top 88 SNPs. The variables *sal2*, *ox2* and *sal1* show even more consistency with the reference by only failing to detect 10% (two SNPs) after excluding 99% of the SNPs from testing. Except for the 90 percentile cutoff, the other variable does also show a higher similarity to the reference when comparing the top 20 ranked SNPs.

Time saving

The main goal of reducing the test set was to save time on the test phase of the BAYENV analysis. Since the time consumption is linear to the number of SNPs tested, we were able to reduce the time spent on the test phase by 90% by excluding 90% of SNPs. A full BAYENV analysis (32 runs and 100,000 iterations using PYBAYENV in parallel mode) on all SNPs (8809) in the data set took approximately 543.2 minutes. By excluding 90% of the SNPs, we were able to reduce this to 54.3 minutes. By further reducing the test set to 95, 97.5 and 99, we were able to reduce the time consumption to 27.2, 13.6 and 5.4 minutes respectively.

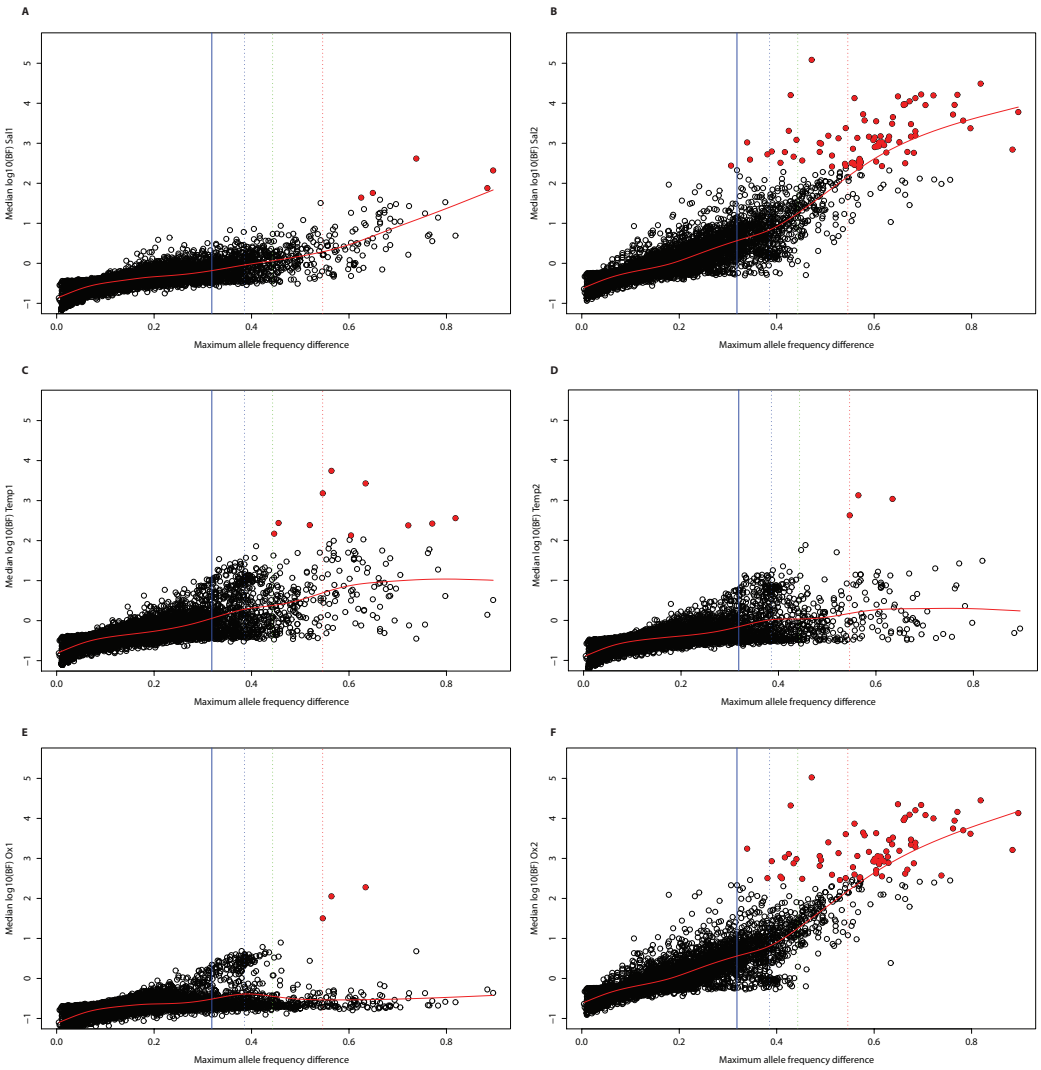

Figure 5.20: Plots showing the correlation between median $\log_{10}(\mathrm{BF})$ and the maximum allele frequency difference (MAFD) for the variables: A) Salinity surface (*sal1*) B) Salinity depth (*sal2*) C) Temperature surface (*temp1*) D) Temperature depth (*temp2*) E) Oxygen surface (*ox2*) and F) Oxygen depth (*ox2*). SNPs identified as significant using SDM are coloured in red. The vertical lines indicate the MAFD cutoffs, where blue, blue dotted, green dotted and red dotted represents a 90, 95, 97.5 and 99 percentile cutoff respectively. The red smoothed spline indicates the trend in the data.

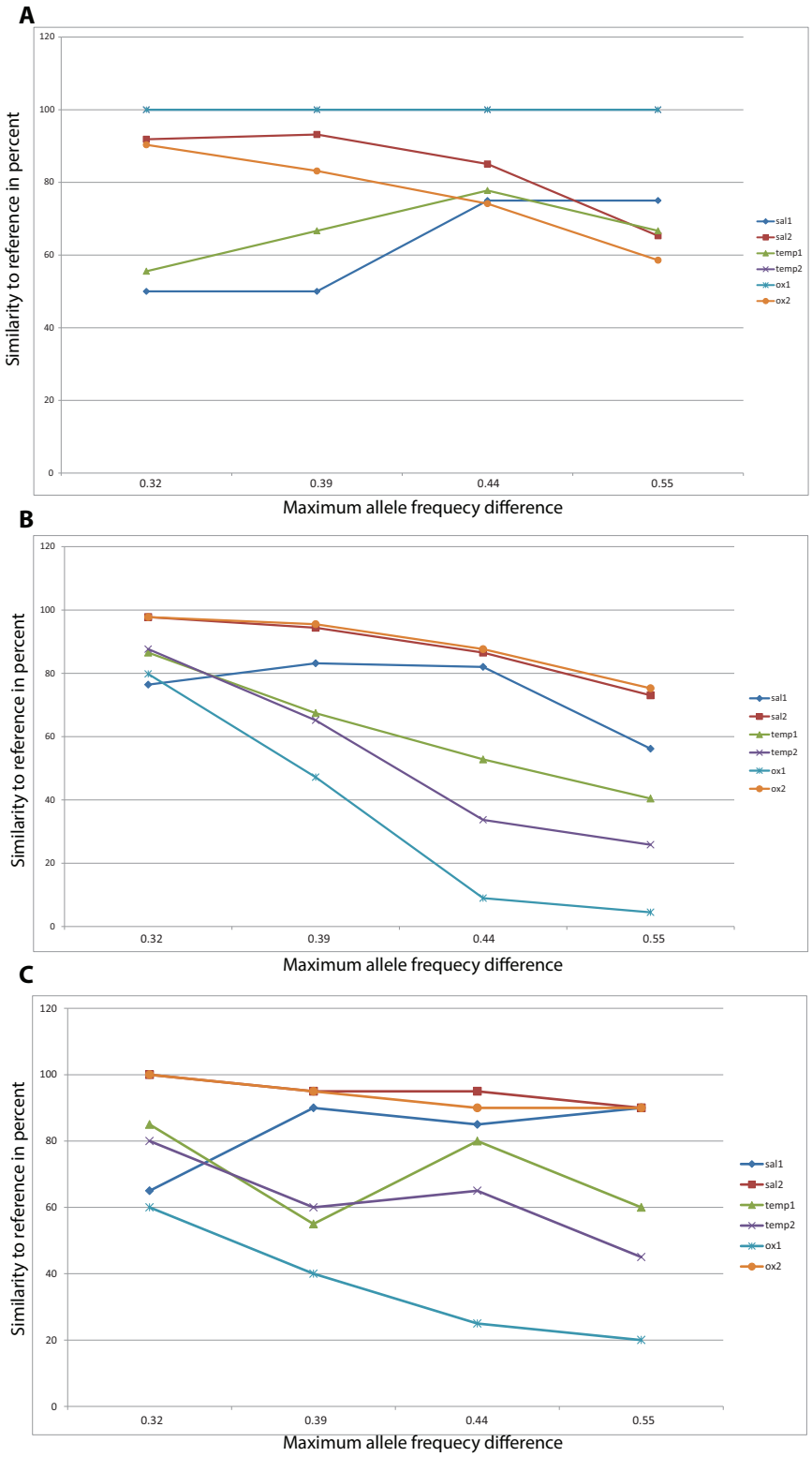

Figure 5.21: Percentage of similar SNPs between BAYENV test and reference results analysing the *Cod* dataset. A) SDM on reduced sets vs. SDM on the reference set (temp2 and ox1 have identical results). B) Top 88 SNPs in reduced test set vs. top 88 (99 percentile) in the reference set. C) Top 20 SNPs in reduced test set vs. top 20 in the reference set.

Chapter 6

Discussion

In this chapter, we discuss and evaluate the methods used and results obtained in this thesis.

6.1 Evaluation of PyBayenv

We developed a wrapper program for Bayenv named PyBayenv as a tool to help us carry out the tests and experiments in this thesis (see Section 4.9 and 5.1). PyBayenv had two main objectives: facilitating and streamlining the Bayenv analysis and providing functions for testing our hypotheses.

The first challenge was to convert the SNP data to the distinctive Bayenv format (see Section 5.1.1 and Figure 5.4). We chose the Genepop file format (Raymond and Rousset 1995) as input format for PyBayenv because this contains population information. In the future, we hope to add support for other file formats such as the Structure format (Falush, Stephens and Pritchard 2003; Pritchard, Stephens and Donnelly 2000). Meanwhile, the user is referred to a third party conversion tool such as PGDSpider (Lischer and Excoffier 2012) in order to convert the SNP data to the Genepop format.

In PyBayenv we implemented a feature for standardising the environmental variables (see Section 5.1.2). The user provides the environmental variables in conventional form and PyBayenv serves to standardise these before they are given to Bayenv as input arguments. By adding this feature we minimize the possibility of user made errors by ensuring that the environmental variables are indeed standardised while simultaneously streamlining the Bayenv analysis.

The Bayenv analysis is rather complicated and cumbersome due to its multi-step procedure (see Section 3.6). To simplify the Bayenv analysis, we implemented a wrapper function in PyBayenv (see Section 5.1.3 and

5.1.4). The most challenging part of this function was to find a good solution for the test phase of BAYENV. In this phase BAYENV needs to be started and stopped for every SNP in the dataset. Additionally, the BAYENV requires that a single SNP file, already on the disk, is provided as input. We solved this by sequentially writing each SNP to disk before calling BAYENV as an external program from PYBAYENV. This is a cumbersome way of carrying out this procedure, however, the program was only available as a compiled binary file, hence we had no access to modify the program directly. Moreover, modifying BAYENV would also have required permission and involvement from the authors of the program (Coop et al. 2010; Günther and Coop 2013). However, that said, we found that the time consumption of writing to disk on every test cycle was marginal compared to the MCMC algorithm in BAYENV.

The wrapper functionality of the test phase of BAYENV is by far the most time saving measure in PYBAYENV. Testing more than a few SNPs "by hand" is unrealistic. A skilled UNIX user might be able to write a *bash* script that can automate the process in a similar way, however, this would require a lot of work and preparations (i.e. the single SNP files in a large dataset). By developing PYBAYENV we think we have provided an easy and time saving solution for all researchers that want to perform the somewhat complicated BAYENV analysis.

The PYBAYENV functionalities for testing our hypothesis will be discussed and evaluated in the subsequent sections.

6.1.1 Parallelization

In order to save time, we implemented a feature in PYBAYENV where several BAYENV runs could be carried out in parallel (see Section 5.1.5 and 5.2). The time estimates from the test runs suggest that there is a time overhead of only 19.3% by running eight runs in parallel compared to one single run. This overhead is negligible compared to the time consumption of eight runs carried out in serial. An estimate of the time consumption for eight sequential runs would be 3560.8 minutes (59.3 hours). Compared to the 543.2 minutes used by running the eight analyses in parallel implies a time saving of 84.7%.

The results from running 16 or 32 analyses in parallel show that there were no further time savings due to the limited number of cores on the lab computer (8 cores). However, running more processes than cores did not slow down the overall time consumption either. Doubling the number of processes compared to cores only lead to a doubling of the time consumption. So, in conclusion: Even if the number of runs exceeds the number of cores, PYBAYENV provides a time saving feature running the processes in parallel mode.

6.2 Evaluation of the convergence of the covariance matrix

An accurate and consistent covariance matrix estimate is important for the test phase of BAYENV where this is used as the null model. Hence, we examined the convergence of both single draws and average covariance matrices calculated from BAYENV runs using different MCMC chain lengths (see Section 5.3). We plotted heatmaps (figure 5.10) to visualize the difference (absolute values) between the test sets and the reference set. By examining the single draw estimates (figure 5.10: A, C and E), it is evident that the covariance estimate is converging towards some kind of consensus: The difference decreases as the MCMC iterations are increased. The covariance matrices based on the average of all estimates output by BAYENV (figure 5.10: B, D and F), show the same trend: better match with the reference as the number of MCMC are increased. Most interestingly, the single draw after 10,000 MCMC iterations perform better than the matrix averaged over the same number of iterations. This is probably due to the fact that a large proportion of the average matrix consists of very early outputs from BAYENV where the estimates are highly unstable. However, the matrix averaged over 100,000 MCMC iterations did outperform the single estimates from both 100,000 and 500,000 MCMC iterations. The consistency of this matrix was only exceeded by the average matrix produced after 500,000 MCMC iterations.

If only few iterations (typically 10,000) is used, it might be better to use the last single draw from the posterior than an average that includes a large proportion of unstable estimates. However, this experiment shows that an average of the estimates produced after 100,000 or 500,000 MCMC iterations is more consistent and thus recommendable. An even better strategy would be to run BAYENV using 500,000 MCMC iterations and declare the output from the first 100,000 MCMC iterations as burn-in and exclude them when calculating the average matrix and hence get rid of the unstable estimates from the beginning of the chain.

6.3 Evaluation of the SDM

As pointed out earlier, the problem with conventional methods such as a percentage or static (i.e. Jefferey's table; see Table 2.2) cutoff is that it does not sufficiently take into account the overall distribution of BFs. For example an *alpha5* cutoff (top 95 percentile) on the results for association to the variable *ox1* (oxygen at surface) yields 283 (out of 440) "significant" results that actually supports the null model by having a BF < 1. Or, using a static cutoff such as *jeff3.2* (BF > 3.2) on the results for association to the variable *ox2* (oxygen at spawning depth) yield as many as 1088 significant SNPs (12% of all SNPs in the *Cod* dataset). By converting the BFs to q-

values, we were able to show that these cutoffs yield an abundance of false positive results (see Table 5.1 and SI table A).

One of the primary goals of this thesis was to provide an alternative method for defining a significance threshold for an empirical distribution of BFs obtained from a BAYENV analysis. We proposed a method we called the Second Difference Method (SDM) whose purpose was to determine where an increasingly sorted distribution of BF values had an approximate linear growth and where it was convex (or exponential). The rationale behind the SDM was to calculate a significance threshold based on the shape of the sorted distribution of BFs for every run and environmental variable individually instead of making a cutoff based on a percentage or static cutoff that does not take the actual distribution of BF values into account. The growth rate was approximated using second difference (central difference, see Section 2.4.1). The hypothesis was that the non-significant results follows a linear trend, whereas the truly significant results stand out by having non-linear growth. The break in the slope where the second difference distribution has a sudden and substantial jump in the positive direction, was made an indicator of a change in the growth rate and used to separate putatively non-significant from putatively significant results (see Method Section 4.2).

The results from testing the SDM on simulated as well as real data, show that the method is able to make an intelligent cutoff that adjusts to the shape of the distribution of the results: when the overall BF signal is low, SDM defines few significant SNPs, whereas in the case of a stronger signal, more SNPs are defined as significant. In terms of FDR, the SDM was found to be more conservative and yielded fewer FPs than conventional methods based on a percentage or static cutoff.

6.3.1 Evaluation of testing the SDM on the simulated data

As an initial experiment before running our proposed SDM on real datasets, we simulated three sets of artificial BAYENV results (BFs) and used the SDM implementation in PYBAYENV to parse the data (see Section 5.4.2). The results show without exception, that the SDM was able to separate all "SNPs" defined in the significant group (*siSNPs*) from "SNPs" in the groups containing neutral (*neSNPs*) and non-significant (*noSNPs*) "SNPs" (see Methods Section 4.6.1 for details about the simulated BF values). In other words, the SDM works as intended on the simulated datasets. It may be argued that the gaps defined between the *noSNPs* and the *siSNPs* "SNPs" were unrealistically large, however, these simulated sets were only intended to demonstrate how SDM uses a sudden and substantial change in the second difference distribution to define the cutoff. Additionally, if we look at the distribution of the second difference for the simulated sets (see Figure 5.11) and the second difference value that triggered the cutoff, it is apparent that the gap could have been considerably less and SDM still

would have been able to separate the two groups. For example, the $\hat{\delta}$ for the *Sim-weak* dataset was 2.3, whereas the cutoff was made where the second difference was 48.7. This was 21.2 times higher than necessary to trigger this cutoff. Actually, only one "SNP" outside the *siSNPs* group had a second difference greater than 0.5. This was found in the *noSNPs* group and had a second difference of 0.55 and corresponded to the 62 largest BF value. Hence, a cutoff threshold based purely on $\epsilon = 0.5$ would have resulted in inclusion of 57 additional "SNPs" from the *noSNPs* group in the significance set.

The difference between $\hat{\delta}$ and the value that triggered the cutoff was even higher for *Sim-strong* and *Sim-large* (27.8 and 8,091.3 times as large as $\hat{\delta}$ respectively). A cutoff threshold $\hat{\delta} = 0.5$ on these datasets would have resulted in inclusion of almost all "SNPs" in *noSNPs* group, however, none of the "SNPs" from the *neSNPs* group would have been included (see Figure 5.11).

6.3.2 Evaluation of the SDM on a single BAYENV run

As an initial test on real data, we carried out a single run of BAYENV on the Cod dataset (Berg et al. 2015, *in review*, see materials Section 4.1.1) using PYBAYENV and interpreted the results using the built in SDM module (see Section 5.4.4). By looking at the distributions of BFs it is evident that some variables exhibit a stronger signal than others. For example, the number of SNPs with a BF greater than 10 (N_α, see equation 4.2) for *sal1* (salinity at surface) was 24, whereas the corresponding number for *sal2* (salinity at spawning depth) was 445. The maximum BF (A) for *sal1* and *sal2* was 359.6 and 100,400.0 respectively. This difference in signal strength was reflected in cutoff threshold $\hat{\delta}$ defined by the SDM. For *sal1* and *sal2* the second difference cutoff $\hat{\delta}$ was 4.0 and 17.2 respectively. Despite the fact that *sal1* had a less stringent cutoff, the number of SNPs in the significance set was considerably less than for *sal2* (six versus 83 SNPs).

We compared the significance sets obtained using SDM to significance sets obtained using the cutoff thresholds *alpha1*, *alpha5* (top 99 and 95 percentile) and *jeff3.2* (BF>3.2). FDR in each set was evaluated by transforming the BF values to q-values. Most interestingly, SDM appears generally to be much more conservative than its counterparts (see Figure 5.1 and 5.14). For the variables *sal1*, *temp1*, *temp2* and *ox1*, the number of significant results is approximately one tenth of the results obtained from the other cutoff methods. However, if we look at the corresponding q-values, the expected proportion of FPs using SDM is in a more "reasonable" range than what is obtained using the other cutoff methods. For example, for the variable *sal1*, all alternative cutoff methods yield at least 52.3% expected false discoveries, whereas the same number is only 8.9% for SDM. In terms of the variables *sal2* and *ox2*, the significance sets obtained from SDM is highly similar to that of *alpha1*. While this is a coincidence it

demonstrates that SDM is able to dynamically adjust to the distribution of BF values. By looking at the maximum q-values for these variables, the expected proportion of FPs in the SDM sets is ≈ 0.01, which we believe to be a conservative q-value cutoff.

This initial experiment on real data indicates that the dynamic cutoff provided by SDM may offer a more flexible way of interpreting the empirical distribution of BFs obtained from a BAYENV analysis than methods based on either a static (*jeff3.2*) or percentage (*alpha1* and *alpha5*) cutoffs. For variables where there are relatively few high ranking SNPs (e.g. *sal1*), SDM is more conservative than the other methods. For variables, such as *sal2* and *ox2* where the overall BF signal is higher, SDM show flexibility by allowing more SNPs to be included in the set of significant results.

6.3.3 Evaluation of the SDM on multiple BAYENV runs

We investigated how different cutoff methods dealt with the run-to-run variability of BAYENV by carrying out 32 independent runs of the program on the *Cod* dataset. We compared the Total Significance Sets (TSS: SNPs defined as significant in more than 70% of the runs) (see method Section 4.6.3) obtained by SDM to corresponding sets using the cutoff methods *alpha1*, *alpha5 jeff3.2* and *jeff10*. As for the experiment on one single BAYENV run, SDM appears to be much more conservative than the other cutoff methods. For the variables *sal1*, *temp1*, *temp2* and *ox2*, the TSS only contained between 3 and 9 SNPs (see Table 5.2). The maximum q-values for these sets however, show that despite few and top ranking SNPs, the FDR is not very low. The FDR for the other cutoff methods is much higher (with the exception of *jeff10* for *ox1*). This fact may indicate that the true number of significant SNPs is small (or even zero) for these variables and that SDM provides a more acceptable cutoff threshold in terms of FDR. The high run-to-run variability seen reflects the uncertainty in the results and underpins this thought.

In contrast to *sal1*, *temp1*, *temp2* and *ox2*, the variables *sal2* and *ox2* showed in general more consistency. As for the previous experiment on a single run of BAYENV (see Results Section 5.4.5), the final set of significant SNPs (TSS), contained approximately the same number of SNPs as the corresponding set using *alpha1* cutoff. The high number of significant SNPs and the low FDR in combination with low run-to-run variability, indicates that the variables *sal2* and *ox2* have a strong linear effect on the population allele frequencies and thus more SNPs are likely to be under selection.

We plotted Venn diagrams of the union sets and the TSS's obtained from the cutoff methods described above to visualise how these sets relates to each other (see Figure 5.16). By examining the Venn diagrams of the union sets from *sal1* and *sal2* (Figure 5.16 A and C) and comparing these to the corresponding TSS's (Figure 5.16 B and D) it is evident that *sal1* has a

much higher run-to-run variability than *sal2* for all cutoff methods. For example, only 2% of the SNPs identified as significant in one or more runs (union set) is consistent in more than 70% of the runs (TSS) using *jeff3.2* and *jeff10* on the results from *sal1*, whereas the corresponding percentage for the variable *sal2* is 60% and 55% respectively. Moreover, the Venn diagrams also help to clarify the efficiency of the SDM and to make salient the disadvantages of the other methods. For example, the Venn diagram for the TSS's of *sal1* (Figure 5.16 B) shows that *jeff10* yields relatively few significant SNPs compared to *alpha1* (17 versus 46). Consequently, the FDR is more "satisfactory" for *jeff10* than for *alpha1* (0.2 versus 0.4) for this variable. However, if we look at the corresponding sets for *sal2* (Figure 5.16 C), the scenario is turned around: now *jeff10* yields almost five times as many significant SNPs (405 versus 85) as an *alpha1* cutoff. The maximum q-values indicates that *jeff10* yields 20% FPs versus 1% FPs using *alpha1*. This tells us that in terms of FDR, the percentage and static cutoffs produce somewhat arbitrarily TSS's that does not adjust to the actual distribution of BFs. The SDM, on the other hand, shows that it is capable of adjusting to the distribution of BFs by defining few SNPs when the overall BF signal is low and the variability is high (*sal1*) and defining more SNPs under the opposite scenario. The q-values calculated for the SDM sets support this supposition (see Table 5.2).

6.3.4 Evaluation of the SDM applied to the Maize dataset

To explore how the SDM would work with smaller datasets, we applied it to the results from a BAYENV analysis on 135 SNPs from African maize (Westengen et al. 2014b). We used PYBAYENV to carry out a full BAYENV analysis (32 runs) and to interpret the results (see Results Section 5.4.6). Due to the low expected signal strength in terms of BF from the data, the constant ϵ was adjusted from 0.5 (default) to 0.2. In 87.5% of the runs (28 of 32 runs), $\hat{\delta}$ was computed to be equal to ϵ. A $\hat{\delta} = \epsilon$ implies that there is at maximum one SNP in the distribution that has a BF value greater than 10 (N_α, see equation 4.2). We saw that there were on average more SNPs defined to be significant when $\hat{\delta}$ was defined to be equal to ϵ. This may indicate that by adjusting down the ϵ value we made $\hat{\delta}$ more tolerant - maybe too tolerant. However, if we look at the median instead of the average statistics, the difference in significance set size is not as obvious (eight versus five), indicating that the average statistics is somewhat skewed due to some outlier results.

When the results from this analysis were published, SNPs showing an average BF$>$ 3 were regarded as significant. As the basis for this decision, Kass' table (see Table 2.3) for interpretation of BF was used. This cutoff resulted in three significant SNPs, all being identified as candidates for positive selection using the software LOSITAN (Antao et al. 2008). Moreover, two of these SNPs were located in known putative protein coding genes with known orthologs in rice and sorghum (Westengen et

al. 2014b). Most interestingly the SDM method detected the same three SNPs (see Figure 5.17. The fact that SDM supports the same cutoff, makes the choice of cutoff even more convincing as it is automated and thus is completely objective.

We saw that the results varied considerably between runs. Only 8.3% (three out of 36) SNPs were consistently ($\kappa = 0.7$) among the "significant" SNPs using the SDM method (see Figure 5.17). This shows the necessity of checking the BAYENV results with multiple runs of the algorithm.

By converting the BFs to q-values, we saw that there may be a high proportion of FPs among the top ranking SNPs indicating some uncertainty of the results. This is probably a consequence of few SNPs forming the basis for the covariance matrix and a low BF signal in general. However, in this experiment we have showed that the SDM could be successfully applied to smaller datasets and our strategy of running multiple runs of the BAYENV algorithm is essential to ensure a stable result.

6.3.5 Evaluation of the $\hat{\delta}$ equation

The equation that SDM uses for defining the cutoff threshold for the second difference ($\hat{\delta}$, see Section 4.2.2), was designed to reflect the shape and overall distribution of BFs from a BAYENV analysis. In addition to a small constant factor (ϵ), the equation consisted of two important measures from the distribution: the largest BF (A) value and the number of SNPs displaying a BF above 10 (N_α). By applying the SDM to simulated as well as real data, we have shown that $\hat{\delta}$ makes smart cutoffs on the second difference distribution. However, we do see that there is room for improvement. For example, when using SDM on the *Maize* dataset, we saw that $\hat{\delta}$ was more than four times as large if the runs had a $N_\alpha > 1$. This fact was reflected in the number of significant SNPs in the corresponding significance sets (see Discussion Section 6.3.4). Perhaps this could have been avoided if ϵ had been determined by statistics from the BF distribution and not by the user. When analysing the *Cod* dataset we used the same ϵ value for all environmental variables (the default value 0.5). Considering that the variables showed very different BF signal strength, perhaps the ϵ should have been differentiated accordingly. An automation of the choice of ϵ would also have made the SDM more user friendly. Moreover, the N_α component may be criticised for affecting the $\hat{\delta}$ equation in the "wrong" direction if adjusted (i.e. a lower α value would lead to a more stringent $\hat{\delta}$). We find $\hat{\delta}$ to be working adequately on the datasets used in this thesis, however, in order to be more confident on the choice of δ (see Section 4.2), we encourage more research on this topic.

6.3.6 Conclusions on SDM

We have in this thesis applied the SDM to the BAYENV results from analysing two different datasets as well as simulated results and shown that it could provide a versatile alternative to other cutoff methods. As we have seen (figure 5.1 and 5.2), a percentage and static cutoff often result in an abundance of false discoveries. Furthermore, by using one of these methods, the choice of cutoff could very easily be misled by the analyst's prior beliefs of selection (Coop et al. 2010). One of the real advantages of the SDM is that it is automated and hence provides an unbiased measure of significance. Moreover, using second difference to detect a change in the growth rate of an empirical distribution of BFs has proven to provide a successful cutoff in terms of FDR. By examining the q-values, we see a trend that using a static cutoff might be better when the BF signal is low (i.e. *ox1* in Figure 5.2), whereas a percentage cutoff may be better when the signal is high (i.e. *ox2* in Figure 5.2). The SDM however, provides cutoff that yields an acceptable proportion of FPs both when the signal from the variable is strong and weak.

Our strategy of running multiple replicate runs of BAYENV seems to work well in partnership with the SDM. The total number of significant SNPs is approximately the same as applying the method to a single run (Figure 5.1) and multiple runs (Figure 5.2). The proportion of FPs in the significance sets also stays approximately constant. The high variability however, indicates that multiple runs are necessary to ensure a stable set of results.

By converting the BFs to q-values we had a measure of FDR in the TSS. We used this to show that the SDM provides a cutoff threshold that was more stringent and thus yielded fewer expected FPs. However, we want to point out that a more stringent cutoff also leads to more FNs. Hence, the choice of cutoff threshold must reflect the proportion FPs that can be tolerated and there is no right or wrong answer to this question. However, a significance level must be defined and we think that the SDM provides an elegant and intuitive way of handling the problem.

6.4 Evaluation of the stability of BAYENV

In this section, we discuss the results from testing the stability of the BAYENV method (see Section 5.5).

6.4.1 The impact of increasing the number of MCMC iterations in the test phase of BAYENV

The variability of MCMC algorithms in general and the BAYENV method in particular is already known (Blair, Granka and Feldman 2014; Coop et al. 2010). To assess how the MCMC chain length affects the stability of the results, we calculated a stability score (see Methods Section 4.2.4) for eight BAYENV analyses carried out using different number of MCMC iterations (see Methods Section 4.7.1). In this experiment we compared the SDM to the *alpha1* cutoff threshold.

The results show that the stability differs dramatically between environmental variables. Whereas two variables exhibit a steady increase in stability (*sal2* and *ox2*) this trend was negative for the four other variables (*sal1*, *temp1*, *temp2* and *ox1*) (see Figure 5.18).

The development of SDM and *alpha1* on *sal2* and *ox2* (see Figure 5.18) is strikingly similar and most probably due to the fact that these methods produce significance sets with approximately the same number of SNPs (ca. 88). The stability gained going from 10,000 to 500,000 MCMC iteration is substantial (≈66% increase). The stability gained going from 100,000 to 500,000 MCMC iterations is less notable (≈15%), however it supports the findings of Blair, Granka and Feldman 2014 (see Section 3.7) suggesting that 500,000 iterations produce slightly more stable results than 100,000 iterations. The increasing stability shown may indicate that there is a strong correlation between the allele frequencies and these variables for a substantial number of SNPs in the dataset.

The low stability score obtained for the variables *sal1*, *temp1*, *temp2* and *ox1*, indicates that these have a high run-to-run variability. The reason for why the stability is decreasing as the number of MCMC iterations is increased is not clear. However, we know from the previous experiments (see Section 5.4.4 and 5.4.5) that only very few SNPs show a persistent high BF for these variables (see Tables 5.1 and 5.2). This may indicate that only a few SNPs truly correlate to these variables.

From Figure 5.18 it is apparent that the stability score is lower in general when using the SDM cutoff. This is even more evident for the variables showing the lowest stability score (i.e. *sal1* and *ox1*). This is probably due to the fact that the significance sets produced by SDM can vary in size and the variation is more noticeable for the most unstable variable.

This experiment shows that a higher number of iterations would lead to more certainty about the results for variables which exhibit a strong signal in terms of BF (*sal2* and *ox2*). For variables not showing this steady increase, there may be none or just a very few SNPs that are really under selection for these variables.

6.4.2 Testing the relationship between run-to-run variability and the number of independent BAYENV runs

To determine whether an increased number of replicate runs would provide a more stable set of significant SNPs, we compared the median statistic from test sets containing 1, 2, 4, 16 and 32 independent runs to the median statistics from a reference analysis consisting of 32 runs (see Section 5.5.2). As for the previous experiments the variables *sal2* and *ox2* displayed the highest agreement with the reference. We saw that the top 88 results from on single run on these variables were practically identical to the median from the reference carried out using 32 independent runs (see Figure 5.19). This suggests that for the most persistent variables, it might be enough to verify one test run with another. For less persistent variables (i.e. *sal1* and *ox1*) however, there is a need for more runs to ensure a stable result. Figure 5.19 shows that the less persistent variables exhibit an increasing agreement with the reference as the number of runs were increased.

This experiment shows that for some variables a couple of runs are enough to verify the results from a BAYENV analysis. For other variables showing less persistence, there is a need to run a substantial number of runs to ensure a stable result. Currently there are (to our knowledge) no method to detect whether a variable is persistent or not prior to the test, thus the solution must be to carry out several independent runs to achieve an acceptable degree of certainty about the results.

6.5 Evaluation of the method of reducing the test set based on maximum allele frequency difference between populations

Since the test phase is by far the most time consuming part of a BAYENV analysis (see Section 3.6.2), we proposed a method were we excluded SNPs based on a maximum allele frequency difference (MAFD) between populations to save time (see Section 4.3). We ran four BAYENV analyses on the *Cod* dataset where we excluded SNPs based on a 90, 95, 97.5 and 99 percentile cutoff on the MAFD distribution. These percentage cutoffs corresponded to a MAFD of 0.32, 0.39, 0.44 and 0.55 respectively.

Figure 5.20 and the spline regression showed that MAFD is highly correlated to the BF results from the BAYENV analysis. With only one exception in *sal2*, all SNPs defined as significant using SDM on the full dataset were found within SNPs with a MAFD greater than 0.32. For the variables with less than 10 significant results (*sal1*, *temp2* and *ox1*) all significant SNPs exhibit a MAFD greater than 0.55 (top 99 percentile). By examining the plots in Figure 5.20 it is evident that most of the interesting results fall within a MAFD interval of [0.3,1]. However, for the variables

sal2 and *ox2* there are SNPs with a MAFD less than 0.2 that achieves a BF greater than 100 (in the top 98 percentile of the distribution). This tells us that caution must be taken when choosing the MAFD cutoff. That said, 54% of the SNPs have a MAFD of less than 0.15, thus removing these from the test set would still offer considerably time savings - especially if the dataset is large.

After running PyBayenv on the reduced sets, we calculated the SDM significance sets and compared it to the corresponding SDM significance sets obtained from the experiment where we ran 32 runs on the entire dataset (see Section 5.4.5). We also computed the median statistics for the reduced datasets and compared the top 88 (*alpha1*) and the top 20 SNPs to the median statistics from the same run on the full dataset. The comparison of the median statistics of top 88 SNPs show a similar trend from what we have seen in the previous experiments: *sal2* and *ox2* show the highest consistency, whereas the other variables show less (see Figure 5.21 B). The corresponding comparison of the top 20 SNPs shows much the same trend, however, without exception all variables exhibit a considerably gain in similarity to the reference (see Figure 5.21 C). This fact indicates that the absolute top ranking SNPs is most likely to be found among SNPs with highest MAFD. The poor performance seen for some variables can be explained by the visualisation in Figure 5.20. For example, the variable *ox1* shows the worst performance both when comparing top 88 and top 20. By examining Figure 5.20 E, we can see that only five of the SNPs with a MAFD greater than 0.5 has a BF above 1. The majority of SNPs with a BF greater than 1 is found in the MAFD interval [0.3, 0.5]. This fact is reflected when the top 88 and 20 was compared to the reference that had no constraints on the MAFD.

The comparison of the SDM significance sets (Figure 5.21 A) shows a different pattern. The two variables (*temp2* and *ox1*) that exhibited the worst performance comparing the median statistics (figure 5.21 B and C) now shows a perfect (100%) similarity to the reference for all MAFD cutoffs. Again, this can be explained by the correlations in Figure 5.20 (D and E). The three SNPs defined as significant by the SDM for these variables when testing all SNPs in the dataset, are all found among SNPs with a MAFD greater than 0.55 (99 percentile). Hence, the SDM defined the same three SNPs when the test set was reduced. The pattern seen for variables *sal2* and *ox2* is more similar to what we obtained using a static cutoff on the median statistics: the similarity to the reference is gradually reduced as the MAFD cutoff is made more stringent. However, the similarity is somewhat poorer using the SDM and there are mainly two factors that are causing this. First, since the SDM defines the cutoff threshold dynamically according to the overall distribution of BFs, the size of the significance sets may differ if these conditions are changed (recall that $\hat{\delta}$ is dependent on statistics from the distribution of the results. See Section 4.2.2 for details). For example the SDM significance set obtained using a 0.32 (90 percentile) cutoff on the MAFD for *sal2*, contains 79 SNPs, whereas the reference set contains 86

SNPs. All the 79 SNPs in the reduced set were included in the reference set, however the set difference is still an issue and becomes evident in the plot (figure 5.21 A). Second, which is most evident when comparing the 88 SNPs with a MAFD greater than 0.55 (99 percentile), is that the significance sets obtained using SDM never contain the two lowest ranking SNPs. Hence, the significance sets contains N-2 SNPs (where N is the number of tested SNPs) at maximum. Just this fact may cause a difference between the test and reference sets - especially if the test sets are small.

The pattern seen for the variables *sal1* and *temp1* when comparing the SDM sets is more unexpected: the similarity increases along with the reduction of the test sets (figure 5.21 A). This pattern can only be explained by the variability already observed on these variables (see Section 5.4.5 and 5.5). By examining the data underlying the results, we can understand why the differences are so extreme. For example, the SDM reference set for *sal1* contains four SNPs, whereas the corresponding SDM significance set obtained after a 0.32 MAFD cutoff (90 percentile) contains two SNPs. The two SNPs missing in the test set were identified in 15 and 13 runs (out of 32) respectively and they were both among the top five ranking SNPs, however they failed to make the cut on variability ($\kappa = 0.7$). The outcome is a 50% difference to the reference. When testing the sets based on a 0.44 and 0.55 MAFD cutoff, one additional SNP made the cutoff for variability and hence increased the similarity to 75%. The lesson from this is that comparing significance sets with few SNPs could have major impacts on the similarity in percent.

The method of reducing the test set based on MAFD has both positive and negative aspects. The main purpose of the method was to reduce the time consumption in the test phase of BAYENV. Since this increases linearly with the number of SNPs tested, we were able to reduce the time usage by up to 99% using this method. However, the time savings comes with some drawbacks that need to be paid attention to. As we have seen (e.g. on *sal2*, Figure 5.20) there is a chance of excluding significant SNPs if the MAFD cutoff is made too strict. Moreover, the lack of a "null" distribution of BFs (the distribution of results from neutral SNPs) could be a disadvantage, especially for the SDM which uses statistics from the full distribution to determine the cutoff ($\hat{\delta}$). A strategy that could work is to use a MAFD cutoff of 0.2, which would have reduced the *Cod* dataset by 72%, and additionally draw 1-5% random samples from the excluded SNPs with low MAFD to get a representative sample to serve as "null" SNPs. A time saving of approximately 70% would be very advantageous - especially if the dataset is large. We find the method of reducing the test set based on MAFD to be very promising, however we refrain from drawing any conclusion before more research has been performed on the subject.

Despite the uncertainties regarding the procedures, we still see some very useful applications for the method. For instance if the researcher wants a quick, initial overview over the most significant results. These

preliminary results can provide useful information by allowing a quick comparison of results from other analysis tools such as LFMM (Frichot et al. 2013), PCADAPT (Duforet-Frebourg, Bazin and Blum 2014) or GINLAND (Guillot 2012). Villemereuil et al. 2014 found that it was possible to greatly reduce the error rates by considering the results from several methods. The method can also provide information on which variables that is consistent or have a high variability between runs (see Section 5.5)).

Another meaningful application would be when the test set is extremely large such as the HGDP data (Foster 2001). This dataset contains 660,918 SNPs and is therefore virtually impossible to analyse using BAYENV on a desktop computer - even on a supercomputer this would take considerable time. By reducing the test set by removing the least "interesting" SNPs, analysing the data immediately becomes more manageable.

In conclusion, before more research has been conducted, the method of excluding SNPs from testing based on the MAFD between populations may serve as a convenient tool for an initial examination of the results from a BAYENV analysis. The method also makes extremely large datasets manageable. The method is included as a feature in the PYBAYENV package.

6.6 Our guidelines for BAYENV

Based on the tests and experiments we have performed on the BAYENV method, we provide a set of general advice as a guideline for how the program could be run to ensure a stable result.

6.6.1 Preparing the covariance matrix

We tested the consistency and convergence of the covariance matrix estimate (see Results Section 5.3 and discussion Section 6.2) and based on these findings we recommend that the covariance matrix is averaged across all single draw estimates output by BAYENV using at least 100,000 MCMC iterations. Considering that the time consumption of estimating the covariance matrix is normally just a fraction of the overall time consumption when performing a complete BAYENV analysis (see Section 3.6.2), we recommend that the covariance matrix is averaged across the outputs from a 500,000 MCMC iterations run.

We refrain from having an opinion on how to choose the SNPs for the covariance matrix estimation, however, it is implicit in the BAYENV model that as many SNPs as possible are used in this phase. If the dataset contains considerably more SNPs than 10,000 (the maximum input for BAYENV), we suggest that the average of the *average matrices* from several different

subsets are used. Moreover, if a large proportion of the SNPs available are known to be candidates for selection, it may be advantageous to exclude these from the set of SNPs that forms the basis for the covariance matrix. This is a measure to ensure the neutrality of the estimate (as we did on the *Maize* dataset, see Section 4.6.4).

6.6.2 The test phase of BAYENV

Based on the findings from the experiments performed on the stability of the BAYENV method (see Section 5.5), we recommend that the tests for correlation to an environmental variable are carried out using at least 100,000 and preferably 500,000 MCMC iterations for each SNP. Using 500,000 instead of 100,000 MCMC iterations will produce a slightly more stable result, however, the performance gain comes with a fivefold increase of the time consumption (see Section 3.6.2). Furthermore, to account for the run-to-run variability we recommend that a minimum of eight and preferably 32 independent runs are used to ensure a stable outcome of the analysis. The number of runs will also greatly affect the time consumption: if PYBAYENV is used on a desktop computer with eight cores, an increase in the number of runs from 8 to 32 would lead to a quadrupling of the time consumption (see Section 5.1.5). If the dataset is large (i.e. more than 50,000 SNPs) we suggest that the dataset is made more manageable by excluding SNPs based on the maximum allele frequency difference (MAFD) between populations (see Section 4.3, 4.8 and 5.6). We stress that this method is still under development, however, a moderate cutoff of 0.2 on the MAFD distribution is most likely quite "safe" and may reduce the time consumption considerably (by ca. 70% in our experiment - see Section 5.6 and 6.5).

6.6.3 Interpretation the BAYENV results

For the interpretation of the BAYENV results we recommend that the SDM is employed and that the TSS (see Section 4.2.3) is computed to account for the run-to-run variability. We found that the SDM provides a flexible and smart cutoff that produces fewer FPs than conventional methods (see Section 5.4.4 and 5.4.5). However, the user should keep in mind that a more stringent cutoff also comes with the possibility of having more FN results. As the method still is under development and testing, we currently recommend that the results are confirmed by other statistics such as the q-value (Storey and Tibshirani 2003). Bayes factors (BFs) could be converted to q-values using the algorithm from Muller, Parmigiani and Rice 2006 as employed in Villemereuil et al. 2014 (see Section 2.6.2).

6.7 Interpreting BAYENV results

The BAYENV model attempts to control for various effects of populations structure, however, it does not claim to fully control the population structure. A high BF indicating that a SNP is linearly correlated to an environmental variable should therefore not be taken at face value for selection (Coop et al. 2010). The alternative model in BAYENV assumes a linear relationship between the population allele frequencies and the environmental variable (see Section 3.2). In theory, there could be various forms of non-linear relationship between these quantities that remain undetected using this method. Thus, a non-significant BF is not a proof for the SNP not being a target of selection for this variable, but only a lack of a linear correlation.

Another fact that needs to be taken into consideration is that the environmental variables tend to co-vary. For example, the significance sets obtained using SDM on the results from the variables *sal2* and *ox2* contained 86 and 77 SNPs respectively. The intersect of these sets is as high as 74 SNPs. In other words, 96% of the SNPs found to be significantly correlated with *ox2* is also significantly correlated with *sal2*. Knowing which variable that actually exerts the selection pressure is therefore difficult: it could be either of the variables, both, or none. Thus drawing a bombastic conclusion about a SNPs (or gene's) association with an environmental variable based on the BAYENV analysis alone is not recommendable.

Linkage Disequilibrium (LD) between SNPs might also complicate the analysis. When a SNP is a (recent) target of selection, the process of recombination could result in high LD in the surrounding regions of the genome. Thus neighbouring SNPs can be hitchhiking along with the SNP really under selection and therefore also be correlated to the same environmental variable. Distinguishing between such SNPs may prove difficult. Berg et al. 2015, *in review* found that most of the SNPs that were highly associated with one or more of the six environmental variables (*sal1*, *sal2*, *temp1*, *temp2*, *ox1*, *ox2*), were found in regions of the Cod genome with high LD.

In conclusion, it is important to be aware of all these facts when assessing the results from a BAYENV analysis. As earlier mentioned, Villemereuil et al. 2014 found that it was possible to significantly reduce the error rates by considering more than one method when assaying SNPs for environmental correlation.

6.8 Future work

We have in this thesis created the SDM (section 4.2) and evaluated it on simulated results (section 5.4.2) and the results from analysing real data (the *Cod* and *Maize* dataset) and found it to be a good alternative to other cutoff methods. However, to be even more confident of the importance of the method, we would suggest that SDM is tested on a simulated dataset where the neutral loci (SNPs) under selection are known in advance. By using simulated data it is easier to evaluate the statistical significance of our methods.

Moreover, we would like to see the SDM as well as the method of reducing the test set based on the maximum allele frequency difference (MAFD, see Section 4.3) further tested on a large dataset that is already thoroughly analysed. For example, the SNPs data from the Human Genome Diversity Project (Foster 2001) would be interesting since this allows comparison of the results to a range of already performed studies (e.g. Blair, Granka and Feldman 2014; Coop et al. 2010; Hancock et al. 2010b; Hancock et al. 2008).

To save time in the test phase of BAYENV, we developed a method of using the measurement maximum allele frequency difference (MAFD) to exclude less interesting SNPs form testing. We showed that the MAFD was significantly correlated to the outcome of a BAYENV analysis and the method showed promising results in our experiments. However, we also saw that a too strict cutoff of the MAFD could potentially exclude SNPs under selection from testing. Hence, in order to be more confident on the MAFD cutoff, more research on the topic is needed.

By developing PYBAYENV , we have not only made BAYENV available for researchers without skills in programming, but we also made a complete framework for streamlining the BAYENV analysis and interpreting the results. However, there are still features that could make PYBAYENV even more efficient and user friendly. For example, a graphical user interface could make PYBAYENV more available for non-technical users. Furthermore, there are a number of features that could be interesting to add to the program. For example, a feature for plotting the BF results as Manhattan plots (see Section 4.4.2), where the significant SNPs defined by SDM are highlighted, would enable the user to inspect how the results are distributed across the genome. Plots of the distribution of the second difference would enable inspection of the SDM cutoff. Venn diagrams (see Section 4.6.3) such as in Figure 5.16 would enable the user to inspect the distribution of the significant results using different cutoff methods. Moreover, to provide the user with information on FDR in the distribution of results, a feature for converting the BFs to q-values could have be implemented in PYBAYENV. Lotterhos and Whitlock 2014 have developed a method for adjusting the empirical p-values in accordance with the results (BFs) from a set of putatively neutral SNPs (determined a priori)

which also could have been calculated by PyBayenv. By having additional statistical measurements such as the q-value and the adjusted p-value, the user can draw conclusion on significance with more confidence.

Furthermore, one could envision a web interface for interpreting the Bayenv results using the SDM in a similar manner as the Evanno method (Evanno, Regnaut and Goudet 2005) for interpreting the Structure results has been made available by Earl and vonHoldt 2012 on the Structure Harvester web page (http://taylor0.biology.ucla.edu/structureHarvester/). The user could upload the Bayenv results to a server that interprets these and returns relevant plots and data.

To make PyBayenv even more efficient, one could envision an implementation of the program on a Galaxy platform (Goecks, Nekrutenko and Taylor 2010) that has access to high performance computing resources such as the *lifeportal.uio.no* (*Lifeportal* 2015). By implementing PyBayenv on *lifeportal.uio.no*, the user is not limited by the number of cores on a desktop computer thus more parallel runs could have been performed simultaneously. Another great advantage with the Galaxy platform is that all steps in the analysis are automatically documented and hence easier to reproduce (Sandve et al. 2013).

6.9 Conclusion

To meet the demand for multiple independent runs of Bayenv, we developed the software package PyBayenv. By parallelizing the Bayenv test phase we were able to drastically reduce the time spent when carrying out multiple analyses. In addition, PyBayenv provides an easy entrance to a quite complicated analysis process by offering features such as format conversion, wrapper functions for the Bayenv steps, interpretation of the results using SDM and reducing the test set based on the maximum allele frequency difference (MAFD) between populations.

In this thesis we developed the SDM, a method for defining a significance threshold for multiple hypothesis testing where the results are given on the form of Bayes factor (BF). The main goal for the method was to assign a cutoff that reflected the shape of the BF distribution instead of employing a conventional cutoff such as a static (Jeffrey's table 2.2) or percentage cutoff. The results from running Bayenv analyses on two SNP datasets show that SDM provides a versatile alternative to the conventional approaches. The results are somewhat similar to what is obtained by converting the BF to q-values (Muller, Parmigiani and Rice 2006) and applying a cutoff threshold of $\alpha_q = 0.01$, with SDM being slightly less conservative for some variables.

We also confirm the findings from Blair, Granka and Feldman 2014 that Bayenv has a high run-to-run variability. However, the results from our datasets suggest that some variables are significantly more unstable than

others. Increasing the number of MCMC iterations did not change this fact. To address this problem, we developed a strategy where the results from several independent runs were examined and only SNPs showing a high consistency between runs were included in the final set of significant results (TSS).

We proposed a method of reducing the overall time consumption of BAYENV by excluding SNPs with low maximum allele frequency difference (MAFD) between populations from testing. When testing the method on real data (the *Cod* dataset), we found that there was a significant correlation between the maximum allele frequency difference and the resulting BF from a BAYENV analysis. These findings suggests that the majority of SNPs may be excluded from testing based on this measurement and thus make a considerable contribution in terms of time savings - especially if the dataset is large. We find the method very promising, however due to the uncertainties regarding the choice of cutoff, we encourage more research on this topic.

Bibliography

Alberto, Florian J., Jérémy Derory, Christophe Boury, Jean-Marc Frigerio, Niklaus E. Zimmermann and Antoine Kremer (Oct. 2013). 'Imprints of Natural Selection Along Environmental Gradients in Phenology-Related Genes of Quercus petraea'. en. In: *Genetics* 195.2. PMID: 23934884, pp. 495–512. DOI: 10.1534/genetics.113.153783.

Almawi, Wassim Y., Hala Tamim, Raghid Kreidy, Georgina Timson, Elias Rahal, Malak Nabulsi, Ramzi R. Finan and Noha Irani-Hakime (June 2005). 'A Case Control Study on the Contribution of Factor V-Leiden, Prothrombin G20210A, and MTHFR C677T Mutations to the Genetic Susceptibility of Deep Venous Thrombosis'. en. In: *Journal of Thrombosis and Thrombolysis* 19.3, pp. 189–196. DOI: 10.1007/s11239-005-1313-x.

Antao, Tiago, Ana Lopes, Ricardo J Lopes, Albano Beja-Pereira and Gordon Luikart (July 2008). 'LOSITAN: A workbench to detect molecular adaptation based on a Fst-outlier method'. In: *BMC Bioinformatics* 9, p. 323. DOI: 10.1186/1471-2105-9-323.

Benjamini, Yoav and Yosef Hochberg (Jan. 1995). 'Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing'. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1, pp. 289–300.

Berg, Paul R., Sissel Jentoft, Bastiaan Star, Kristoffer H. Ring, Halvor Knutsen, Sigbjørn Lien, Kjetill S. Jakobsen and Carl André (2015). 'Adaptation to low salinity promotes genomic divergence in Atlantic cod (Gadus morhua L.)' en. In: *In review*.

Blair, Lily, Julie Granka and Marcus Feldman (2014). 'On the stability of the Bayenv method in assessing human SNP-environment associations'. In: *Human Genomics* 8.1, p. 1. DOI: 10.1186/1479-7364-8-1.

Blanquart, F., S. Gandon and S. L. Nuismer (2012). 'The effects of migration and drift on local adaptation to a heterogeneous environment'. en. In: *Journal of Evolutionary Biology* 25.7, pp. 1351–1363. DOI: 10.1111/j.1420-9101.2012.02524.x.

Blanquart, François, Oliver Kaltz, Scott L. Nuismer and Sylvain Gandon (2013). 'A practical guide to measuring local adaptation'. en. In: *Ecology Letters* 16.9, pp. 1195–1205. DOI: 10.1111/ele.12150.

Bonhomme, Maxime, Claude Chevalet, Bertrand Servin, Simon Boitard, Jihad Abdallah, Sarah Blott and Magali SanCristobal (Sept. 2010). 'Detecting Selection in Population Trees: The Lewontin and Krakauer Test Extended'. en. In: *Genetics* 186.1, pp. 241–262. DOI: 10.1534/genetics.110.117275.

Bradshaw, William E. and Christina M. Holzapfel (Dec. 2001). 'Genetic shift in photoperiodic response correlated with global warming'. en. In: *Proceedings of the National Academy of Sciences* 98.25, pp. 14509–14511. DOI: 10.1073/pnas.241391498.

Campbell, Neil A, Jane B Reece, Lisa A Urry, Michael L Cain, Steven A Wasserman, Peter V Minorsky and Robert B Jackson (June 2008). *Biology, eight edition*. en. San Francisco CA Pearson Benjamin Cummings.

Carlin, Bradley P. and Thomas A. Louis (Mar. 2011). *Bayesian Methods for Data Analysis, Third Edition*. en. CRC Press.

Chen, Jun, Thomas Källman, Xiaofei Ma, Niclas Gyllenstrand, Giusi Zaina, Michele Morgante, Jean Bousquet, Andrew Eckert, Jill Wegrzyn, David Neale et al. (July 2012). 'Disentangling the Roles of History and Local Selection in Shaping Clinal Variation of Allele Frequencies and Gene Expression in Norway Spruce (Picea abies)'. en. In: *Genetics* 191.3. PMID: 22542968, pp. 865–881. DOI: 10.1534/genetics.112.140749.

Coop, Graham, David Witonsky, Anna Di Rienzo and Jonathan K. Pritchard (Aug. 2010). 'Using Environmental Correlations to Identify Loci Underlying Local Adaptation'. In: *Genetics* 185.4, pp. 1411–1423. DOI: 10.1534/genetics.110.114819.

De Mita, Stéphane, Anne-Céline Thuillet, Laurène Gay, Nourollah Ahmadi, Stéphanie Manel, Joëlle Ronfort and Yves Vigouroux (Mar. 2013). 'Detecting selection along environmental gradients: analysis of eight methods and their effectiveness for outbreeding and selfing populations'. eng. In: *Molecular ecology* 22.5. PMID: 23294205, pp. 1383–1399. DOI: 10.1111/mec.12182.

Duforet-Frebourg, N., E. Bazin and M. G. B. Blum (Sept. 2014). 'Genome scans for detecting footprints of local adaptation using a Bayesian factor model'. In: *Molecular Biology and Evolution* 31.9. arXiv: 1402.5321, pp. 2483–2495. DOI: 10.1093/molbev/msu182.

Earl, Dent A. and Bridgett M. vonHoldt (June 2012). 'STRUCTURE HARVESTER: a website and program for visualizing STRUCTURE output and implementing the Evanno method'. en. In: *Conservation Genetics Resources* 4.2, pp. 359–361. DOI: 10.1007/s12686-011-9548-7.

Eckert, Andrew J., Andrew D. Bower, Santiago C. González-Martínez, Jill L. Wegrzyn, Graham Coop and David B. Neale (2010). 'Back to nature: ecological genomics of loblolly pine (Pinus taeda, Pinaceae)'. en. In: *Molecular Ecology* 19.17, pp. 3789–3805. DOI: 10.1111/j.1365-294X.2010.04698.x.

Evanno, G., S. Regnaut and J. Goudet (2005). 'Detecting the number of clusters of individuals using the software structure: a simulation study'. en. In: *Molecular Ecology* 14.8. DOI: 10.1111/j.1365-294X.2005.02553.x.

Evans, Luke M., Gancho T. Slavov, Eli Rodgers-Melnick, Joel Martin, Priya Ranjan, Wellington Muchero, Amy M. Brunner, Wendy Schackwitz, Lee Gunter, Jin-Gui Chen et al. (Oct. 2014). 'Population genomics of Populus trichocarpa identifies signatures of selection and adaptive trait associations'. en. In: *Nature Genetics* 46.10, pp. 1089–1096. DOI: 10.1038/ng.3075.

Falush, Daniel, Matthew Stephens and Jonathan K. Pritchard (Aug. 2003). 'Inference of Population Structure Using Multilocus Genotype Data: Linked Loci and Correlated Allele Frequencies'. en. In: *Genetics* 164.4. PMID: 12930761, pp. 1567–1587.

Fang, Zhou, Tanja Pyhäjärvi, Allison L. Weber, R. Kelly Dawe, Jeffrey C. Glaubitz, José de Jesus Sánchez González, Claudia Ross-Ibarra, John Doebley, Peter L. Morrell and Jeffrey Ross-Ibarra (July 2012). 'Megabase-Scale Inversion Polymorphism in the Wild Ancestor of Maize'. en. In: *Genetics* 191.3, pp. 883–894. DOI: 10.1534/genetics.112.138578.

Foll, Matthieu and Oscar Gaggiotti (Oct. 2008). 'A Genome-Scan Method to Identify Selected Loci Appropriate for Both Dominant and Codominant Markers: A Bayesian Perspective'. en. In: *Genetics* 180.2, pp. 977–993. DOI: 10.1534/genetics.108.092221.

Foster, Morris W (2001). 'Human Genome Diversity Project (HGDP)'. en. In: *eLS*. John Wiley & Sons, Ltd.

Frichot, Eric, Sean D. Schoville, Guillaume Bouchard and Olivier François (July 2013). 'Testing for Associations between Loci and Environmental Gradients Using Latent Factor Mixed Models'. en. In: *Mol Biol Evol* 30.7. PMID: 23543094, pp. 1687–1699. DOI: 10.1093/molbev/mst063.

Fumagalli, Matteo, Manuela Sironi, Uberto Pozzoli, Anna Ferrer-Admettla, Linda Pattini and Rasmus Nielsen (Nov. 2011). 'Signatures of Environmental Genetic Adaptation Pinpoint Pathogens as the Main Selective Pressure through Human Evolution'. In: *PLoS Genet* 7.11, e1002355. DOI: 10.1371/journal.pgen.1002355.

Ganal, Martin W., Gregor Durstewitz, Andreas Polley, Aurélie Bérard, Edward S. Buckler, Alain Charcosset, Joseph D. Clarke, Eva-Maria Graner, Mark Hansen, Johann Joets et al. (Dec. 2011). 'A Large Maize (Zea mays L.) SNP Genotyping Array: Development and Germplasm Genotyping, and Genetic Mapping to Compare with the B73 Reference Genome'. In: *PLoS ONE* 6.12, e28334. DOI: 10.1371/journal.pone.0028334.

Günther, Torsten and Graham Coop (Sept. 2013). 'Robust Identification of Local Adaptation from Allele Frequencies'. en. In: *Genetics* 195.1. PMID: 23821598, pp. 205–220. DOI: 10.1534/genetics.113.152462.

Goecks, Jeremy, Anton Nekrutenko and James Taylor (2010). 'Galaxy: a comprehensive approach for supporting accessible, reproducible, and

transparent computational research in the life sciences'. In: *Genome Biol* 11.8. PMID: 20738864 PMCID: 2945788, R86. DOI: 10.1186/gb-2010-11-8-r86.

Guillot, Gilles (2012). 'Detection of correlation between genotypes and environmental variables. A fast computational approach for genomewide studies'. In:

Guillot, Gilles, Renaud Vitalis, Arnaud le Rouzic and Mathieu Gautier (May 2014). 'Detecting correlation between allele frequencies and environmental variables as a signature of selection. A fast computational approach for genome-wide studies'. In: *Spatial Statistics*. Spatial Statistics Miami 8, pp. 145–155. DOI: 10.1016/j.spasta.2013.08.001.

Hancock, A. M., D. B. Witonsky, E. Ehler, G. Alkorta-Aranburu, C. Beall, A. Gebremedhin, R. Sukernik, G. Utermann, J. Pritchard, G. Coop et al. (May 2010a). 'Human adaptations to diet, subsistence, and ecoregion are due to subtle shifts in allele frequency'. In: *Proceedings of the National Academy of Sciences* 107.Supplement_2, pp. 8924–8930. DOI: 10.1073/pnas.0914625107.

— (May 2010b). 'Human adaptations to diet, subsistence, and ecoregion are due to subtle shifts in allele frequency'. en. In: *Proceedings of the National Academy of Sciences* 107.Supplement_2, pp. 8924–8930. DOI: 10.1073/pnas.0914625107.

Hancock, Angela M, David B Witonsky, Adam S Gordon, Gidon Eshel, Jonathan K Pritchard, Graham Coop and Anna Di Rienzo (Feb. 2008). 'Adaptations to Climate in Candidate Genes for Common Metabolic Disorders'. In: *PLoS Genet* 4.2, e32. DOI: 10.1371/journal.pgen.0040032.

Hancock, Angela M., David B. Witonsky, Gorka Alkorta-Aranburu, Cynthia M. Beall, Amha Gebremedhin, Rem Sukernik, Gerd Utermann, Jonathan K. Pritchard, Graham Coop and Anna Di Rienzo (Apr. 2011a). 'Adaptations to Climate-Mediated Selective Pressures in Humans'. In: *PLoS Genet* 7.4, e1001375. DOI: 10.1371/journal.pgen.1001375.

— (Apr. 2011b). 'Adaptations to Climate-Mediated Selective Pressures in Humans'. In: *PLoS Genet* 7.4, e1001375. DOI: 10.1371/journal.pgen.1001375.

Heerwaarden, Joost van, Matthew B. Hufford and Jeffrey Ross-Ibarra (July 2012). 'Historical genomics of North American maize'. en. In: *Proceedings of the National Academy of Sciences* 109.31, pp. 12420–12425. DOI: 10.1073/pnas.1209275109.

Hemmer-Hansen, Jakob, Einar Eg Nielsen, Dorte Meldrup and Christian Mittelholzer (Mar. 2011). 'Identification of single nucleotide polymorphisms in candidate genes for growth and reproduction in a nonmodel organism; the Atlantic cod, Gadus morhua'. en. In: *Molecular Ecology Resources* 11, pp. 71–80. DOI: 10.1111/j.1755-0998.2010.02940.x.

Hubert, Sophie, Brent Higgins, Tudor Borza and Sharen Bowman (Mar. 2010). 'Development of a SNP resource and a genetic linkage map for Atlantic cod (Gadus morhua)'. en. In: *BMC Genomics* 11.1, p. 191. DOI: 10.1186/1471-2164-11-191.

Huxley, Julian (1938). 'Clines: an Auxiliary Taxonomic Principle'. In: *Nature*.

Jakobsson, Mattias and Noah A. Rosenberg (July 2007). 'CLUMPP: a cluster matching and permutation program for dealing with label switching and multimodality in analysis of population structure'. en. In: *Bioinformatics* 23.14, pp. 1801–1806. DOI: 10.1093/bioinformatics/btm233.

Jobling, Mark, Edward Hollox, Matthew Hurles, Toomas Kivisild and Chris Tyler-Smith (June 2013). *Human Evolutionary Genetics, Second Edition*. en. Garland Science.

Jones, Eric, Travis Oliphant, Pearu Peterson et al. (2001–). *SciPy: Open source scientific tools for Python*.

Kass, Robert E. and Adrian E. Raftery (June 1995). 'Bayes Factors'. In: 90.430, pp. 773–795. DOI: 10.1080/01621459.1995.10476572.

Kawecki, Tadeusz J. and Dieter Ebert (2004). 'Conceptual issues in local adaptation'. en. In: *Ecology Letters* 7.12, pp. 1225–1241. DOI: 10.1111/j.1461-0248.2004.00684.x.

Leinonen, Tuomas, R. J. Scott McCairns, Robert B. O'Hara and Juha Merila (Mar. 2013). 'QST-FST comparisons: evolutionary and ecological insights from genomic heterogeneity'. In: *Nat Rev Genet* 14.3, pp. 179–190. DOI: 10.1038/nrg3395.

Lifeportal (2015). URL: https://lifeportal.uio.no/.

Lischer, H. E. L. and L. Excoffier (Jan. 2012). 'PGDSpider: an automated data conversion tool for connecting population genetics and genomics programs'. en. In: *Bioinformatics* 28.2, pp. 298–299. DOI: 10.1093/bioinformatics/btr642.

Lobell, David B., Marshall B. Burke, Claudia Tebaldi, Michael D. Mastrandrea, Walter P. Falcon and Rosamond L. Naylor (Feb. 2008). 'Prioritizing Climate Change Adaptation Needs for Food Security in 2030'. en. In: *Science* 319.5863, pp. 607–610. DOI: 10.1126/science.1152339.

Lobell, David B., Marianne Bänziger, Cosmos Magorokosho and Bindiganavile Vivek (Apr. 2011). 'Nonlinear heat effects on African maize as evidenced by historical yield trials'. en. In: *Nature Climate Change* 1.1, pp. 42–45. DOI: 10.1038/nclimate1043.

Lotterhos, Katie E. and Michael C. Whitlock (2014). 'Evaluation of demographic history and neutral parameterization on the performance of FST outlier tests'. en. In: *Molecular Ecology* 23.9, pp. 2178–2192. DOI: 10.1111/mec.12725.

MATLAB (2012). *8.0.0.783 (R2012b)*. Natick, Massachusetts: The MathWorks Inc.

Moen, Thomas, Ben Hayes, Frank Nilsen, Madjid Delghandi, Kjersti T. Fjalestad, Svein-Erik Fevolden, Paul R. Berg and Sigbjørn Lien (Feb. 2008). 'Identification and characterisation of novel SNP markers in Atlantic cod: Evidence for directional selection'. en. In: *BMC Genetics* 9.1, p. 18. DOI: 10.1186/1471-2156-9-18.

Muller, Peter, Giovanni Parmigiani and Kenneth Rice (July 2006). 'FDR and Bayesian Multiple Comparisons Rules'. In: *Johns Hopkins University, Dept. of Biostatistics Working Papers*.

Neuenschwander, Samuel, Frédéric Hospital, Frédéric Guillaume and Jérôme Goudet (July 2008). 'quantiNemo: an individual-based program to simulate quantitative traits with explicit genetic architecture in a dynamic metapopulation'. en. In: *Bioinformatics* 24.13, pp. 1552–1553. DOI: 10.1093/bioinformatics/btn219.

Nicholson, George, Albert V. Smith, Frosti Jónsson, Ómar Gústafsson, Kári Stefánsson and Peter Donnelly (2002). 'Assessing population differentiation and isolation from single-nucleotide polymorphism data'. en. In: 64.4, 695–715. DOI: 10.1111/1467-9868.00357.

Nielsen, Einar E., Jakob Hemmer-Hansen, Nina A. Poulsen, Volker Loeschcke, Thomas Moen, Torild Johansen, Christian Mittelholzer, Geir-Lasse Taranger, Rob Ogden and Gary R. Carvalho (Dec. 2009). 'Genomic signatures of local directional selection in a high gene flow marine organism; the Atlantic cod (Gadus morhua)'. en. In: *BMC Evolutionary Biology* 9.1, p. 276. DOI: 10.1186/1471-2148-9-276.

Nielsen, Rasmus (2005). 'Molecular Signatures of Natural Selection'. In: *Annual Review of Genetics* 39.1, pp. 197–218. DOI: 10.1146/annurev.genet.39.073003.112420.

Nordborg, Magnus and Simon Tavaré (Feb. 2002). 'Linkage disequilibrium: what history has to tell us'. In: *Trends in Genetics* 18.2, pp. 83–90. DOI: 10.1016/S0168-9525(02)02557-X.

North, B. V., D. Curtis and P. C. Sham (Aug. 2002). 'A Note on the Calculation of Empirical P Values from Monte Carlo Procedures'. In: *The American Journal of Human Genetics* 71.2, pp. 439–441. DOI: 10.1086/341527.

Pennisi, Elizabeth (Sept. 2012). 'ENCODE Project Writes Eulogy for Junk DNA'. en. In: *Science* 337.6099. PMID: 22955811, pp. 1159–1161. DOI: 10.1126/science.337.6099.1159.

Porras-Hurtado, Liliana, Yarimar Ruiz, Carla Santos, Christopher Phillips, Ángel Carracedo and Maria V. Lareu (May 2013). 'An overview of STRUCTURE: applications, parameter settings, and supporting software'. In: *Frontiers in Genetics* 4. DOI: 10.3389/fgene.2013.00098.

Pritchard, J K, M Stephens and P Donnelly (June 2000). 'Inference of population structure using multilocus genotype data'. eng. In: *Genetics* 155.2. PMID: 10835412, pp. 945–959.

Pritchard, Jonathan K. and Anna Di Rienzo (Oct. 2010). 'Adaptation – not by sweeps alone'. en. In: *Nature Reviews Genetics* 11.10, pp. 665–667. DOI: 10.1038/nrg2880.

Pujolar, J. M., M. W. Jacobsen, T. D. Als, J. Frydenberg, K. Munch, B. Jonsson, J. B. Jian, L. Cheng, G. E. Maes, L. Bernatchez et al. (2014). 'Genome-wide single-generation signatures of local selection in the pan-

mictic European eel'. In: *Molecular Ecology* 23.10, pp. 2514–2528. DOI: 10.1111/mec.12753.

Pyhäjärvi, Tanja, Matthew B. Hufford, Sofiane Mezmouk and Jeffrey Ross-Ibarra (Jan. 2013). 'Complex Patterns of Local Adaptation in Teosinte'. en. In: *Genome Biology and Evolution* 5.9. PMID: 23902747, pp. 1594–1609. DOI: 10.1093/gbe/evt109.

Raymond, M. and F. Rousset (1995). 'GENEPOP (Version 1.2): Population Genetics Software for Exact Tests and Ecumenicism'. In: *Journal of Heredity* 86.3, pp. 248–249. eprint: http://jhered.oxfordjournals.org/content/86/3/248.full.pdf+html.

Sandve, Geir Kjetil, Anton Nekrutenko, James Taylor and Eivind Hovig (Oct. 2013). 'Ten Simple Rules for Reproducible Computational Research'. In: *PLoS Comput Biol* 9.10, e1003285. DOI: 10.1371/journal.pcbi.1003285.

Savolainen, Outi, Martin Lascoux and Juha Merila (Nov. 2013). 'Ecological genomics of local adaptation'. In: *Nat Rev Genet* 14.11, pp. 807–820.

Savolainen, Outi, Tanja Pyhäjärvi and Timo Knürr (2007). 'Gene Flow and Local Adaptation in Trees'. In: *Annual Review of Ecology, Evolution, and Systematics* 38.1, pp. 595–619. DOI: 10.1146/annurev.ecolsys.38.091206.095646.

Schlötterer, Christian (Dec. 2002). 'Towards a molecular characterization of adaptation in local populations'. In: *Current Opinion in Genetics & Development* 12.6, pp. 683–687. DOI: 10.1016/S0959-437X(02)00349-0.

Sobel, James M., Grace F. Chen, Lorna R. Watt and Douglas W. Schemske (Feb. 2010). 'The Biology of Speciation'. en. In: *Evolution* 64.2, pp. 295–315. DOI: 10.1111/j.1558-5646.2009.00877.x.

Stefano, V. De and G. Leone (Jan. 1995). 'Resistance to activated protein C due to mutated factor V as a novel cause of inherited thrombophilia'. en. In: *Haematologica* 80.4, pp. 344–356.

Storey, John D. and Robert Tibshirani (Aug. 2003). 'Statistical significance for genomewide studies'. en. In: *Proceedings of the National Academy of Sciences* 100.16, pp. 9440–9445. DOI: 10.1073/pnas.1530509100.

Villemereuil, Pierre de, Éric Frichot, Éric Bazin, Olivier François and Oscar E. Gaggiotti (2014). 'Genome scan methods against more complex models: when and how much should we trust them?' en. In: *Molecular Ecology* 23.8, pp. 2006–2019. DOI: 10.1111/mec.12705.

Westengen, Ola T., Paul R. Berg, Matthew P. Kent and Anne K. Brysting (Oct. 2012). 'Spatial Structure and Climatic Adaptation in African Maize Revealed by Surveying SNP Diversity in Relation to Global Breeding and Landrace Panels'. In: *PLoS ONE* 7.10, e47832. DOI: 10.1371/journal.pone.0047832.

Westengen, Ola T., Mark Atam Okongo, Leo Onek, Trygve Berg, Hari Upadhyaya, Siri Birkeland, Siri Dharma Kaur Khalsa, Kristoffer H. Ring, Nils C. Stenseth and Anne K. Brysting (Sept. 2014a). 'Ethnolinguistic

structuring of sorghum genetic diversity in Africa and the role of local seed systems'. en. In: *Proceedings of the National Academy of Sciences* 111.39, pp. 14100–14105. DOI: 10.1073/pnas.1401646111.

Westengen, Ola T., Kristoffer H. Ring, Paul R. Berg and Anne K. Brysting (Jan. 2014b). 'Modern maize varieties going local in the semi-arid zone in Tanzania'. en. In: *BMC Evolutionary Biology* 14.1, p. 1. DOI: 10.1186/1471-2148-14-1.

Ye, Kaixiong, Jian Lu, Srilakshmi Madhura Raj and Zhenglong Gu (Aug. 2013). 'Human expression QTLs are enriched in signals of environmental adaptation'. en. In: *Genome Biology and Evolution*. PMID: 23960253, evt124. DOI: 10.1093/gbe/evt124.

Chapter 7

Appendix I

Supporting information can be found on the following webpage by providing the username *sdm* and password *PyBayenv*:

http://folk.uio.no/kristori/thesis/SI/

The python package PyBayenv can be downloaded from the following address (same username and password as above):

http://folk.uio.no/kristori/thesis/pybayenv/