universität
wien

DIPLOMARBEIT

Titel der Diplomarbeit

A Simulation Approach for Multilocus Selection-Migration Models

Verfasser

Peter Kepplinger

angestrebter akademischer Grad

Magister der Naturwissenschaften
(Mag. rer. nat.)

Wien, 2012

Studienkennzahl:	A 405
Studienrichtung:	Mathematik
Betreuer:	Ao. Univ.-Prof. Dr. Reinhard Bürger

Abstract

In this diploma thesis, the implementation of a software package is presented, which facilitates the simulation and analysis of multilocus migration-selection models. The deterministic, discrete simulation iterates the underlying difference equation to find the equilibria of the dynamical system. As an application, the equilibrium structure of a population under quadratic stabilizing selection is investigated.

First, we state the biological assumptions and introduce the mathematical model. We define the investigated dynamical system and introduce fitness functions, recombination, and migration models. Furthermore, we define important quantities to measure properties of the genetic composition and of differentiation at equilibrium.

Then, we review related work concerning forward-time simulations and quadratic stabilizing selection. Moreover, we discuss the limiting case of strong migration.

This is followed by a discussion of the implementation of the developed software. This comprises the object model, the database architecture, and a discussion of algorithmic issues.

Finally, the results obtained by the application of the program to the case of quadratic stabilizing selection are presented. First, the case of a diallelic two-locus panmictic population is investigated, allowing for arbitrary optimum position. Then, two demes are considered, displacing the optima symmetrically within the demes, and assuming the Deakin migration model.

Deutsche Zusammenfassung

Diese Diplomarbeit stellt eine Implementierung eines Software-Packets zur Simulation von Migrations-Selektionsmodellen vor. Die deterministische, diskrete Simulation iteriert die zugrunde liegende Differenzengleichung, um die Gleichgewichte des dynamischen Systems zu finden. Als Anwendungsfall wird die Gleichgewichtsstruktur einer Population unter quadratisch-stabilisierender Selektion untersucht.

Zuerst wird das zugrunde liegende mathematische Modell eingeführt und die getroffenen biologischen Annahmen erklärt. Das untersuchte dynamische System wird definiert, Fitnessfunktionen, Rekombination und Migrationsmodelle werden abgehandelt. Weiters definieren wir wichtige Größen, die erlauben die genetische Zusammensetzung und die Differenzierung in Gleichgewichten zu messen.

Danach werden relevante Publikationen besprochen, die Simulationen vorwärts in der Zeit und quadratisch stabilisierende Selektion betreffen. Weiters wird der Grenzfall starker Migration betrachtet.

Dem folgt eine detailierte Beschreibung der Implementierung. Dies umfasst das Objekt-Modell, die Datenbankarchitektur und eine Diskussion algorithmischer Be-

lange.

Schließlich werden die Resultate duch Anwendung der vorgestellten Simulation auf den Fall quadratisch stabilisierender Selektion vorgestellt. Zuerst wird der Fall einer panmiktischen Population mit zwei Allelen auf zwei Loci untersucht, wobei das Optimum der Fitnessfunktion beliebig ist. Anschließend werden zwei Deme mit symmetrisch verschobenen Optima behandelt, um Migration anhand des Deakin-Modells zu untersuchen.

Danksagung

Ich möchte Herrn Professor Reinhard Bürger für die Betreuung und Unterstützung meiner Diplomarbeit danken, der auf jede meiner Fragen mit viel Geduld eingegangen ist. Ebenso war ich für die finanzielle Unterstützung im Rahmen des FWF-Projekts P21305 dankbar.

Weiters bedanke ich mich ganz herzlich bei Johannes Wessely, der mit mir viele Stunden diskutiert hat und sich immer wieder neue Ansätze meinerseits angehört und ergänzt hat. Viele nötige Ideen für die Implementierung gehen auf Gespräche mit ihm zurück. Auf das Heinrichsche Institut!

Ganz großer Dank gilt meiner Freundin Liv Gaupp-Berghausen, die mir auch in dieser Zeit, in der die Arbeit entstanden ist, ihre Unterstützung geschenkt hat. Wie vieles in meinem Leben, habe ich ihr auch die Möglichkeit zu verdanken, mein Studium nun endlich abschließen zu können. Mögen wir drei davon profitieren!

Contents

1 Model

This chapter gives an overview of the biological assumptions and states the mathematical model used for implementation.

1.1 Biological Assumptions

Throughout this section, we follow the definitions given in Futuyma (2005). A *diploid* population, i.e., of organisms with two sets of homologous chromosomes, with discrete, non-overlapping generation is modeled, without taking sexual differentiation into account. The population is subdivided into *demes*, also referred to as *niches*. The population consisting of all the demes is also referred to as a *metapopulation*. For the analysis we consider multiple *alleles*, i.e., different DNA sequences, on multiple *loci*, by which we mean gene location on the DNA.

Within the randomly mating subpopulations *viability selection* acts on the zygotes, and *recombination* events occur at a constant rate in time. Viability selection reflects the probability of survival until reproductive age. We allow multiple crossover events between pairs of homologous chromosomes as recombination, which results in new combinations of genes.

Individuals *migrate* from one deme to another depending on probabilities taking only geographical but not genetical differences into account. In general, migration refers to "*gene flow among populations*" (Futuyma, 2005, p. 550).

The population size is considered as infinitely large, thus, *random genetic drift* can be ignored.

1.2 Mathematical Model

Diagram (1.1) depicts the life cycle of an individual within the population.

$$Zygote \xrightarrow{\text{Selection}} Adult \xrightarrow{\text{Migration}} Adult \xrightarrow[\text{Recombination}]{\text{Reproduction}} Zygote \tag{1.1}$$

Viability selection acts on the offspring, potentially changing the relative size of the demes. After selection, adult individuals migrate independently of their genotype, supposing no individuals are lost during migration. Due to migration, the relative deme size may change again. Finally, reproduction including recombination with random mating results in the next generation of zygotes.

Individuals are genetically modeled by a multiallelic multilocus system. Thus, (1.1) is a special case of the life cycle stated in (Nagylaki, 1992, p. 133) and extended to multiple loci. Based on the formulation and notation in Bürger (2009), we consider $L \geq 1$ loci, and on each locus a set of $I \geq 2$ different alleles. The population is subdivided in Γ discrete demes. Let $A_{i_n}^{(n)}$ for $i_n = 1, ..., I$ and $n = 1, ..., L$ denote allele i_n on locus n. The multi index $i = (i_1, ..., i_L)$ allows us to denote the frequency of the gamete i, i.e., $A_{i_1}^{(1)}...A_{i_L}^{(L)}$, immediately after gametogenesis within deme α by $p_{i,\alpha}$. The number of different gametes is then given by $N = I^L$. Since distinction of the gametic frequencies belonging to different demes is necessary, we have to keep track of $\mathcal{N} = N \cdot \Gamma = I^L {\cdot} \Gamma$ gamete frequencies.

The set of alleles is denoted by $\mathsf{I} = \{1, ..., I\}$, the set of loci by $\mathsf{L} = \{1, ..., L\}$, the set of demes by $\mathsf{G} = \{1, ..., \Gamma\}$, and the set of gametes by N.

We are interested in the simplex as state space of the proposed model. The simplex representing a single deme is given by

$$\Delta_N = \left\{ x \in \mathbb{R}^N : \sum_{i \in \mathsf{N}} x_i = 1, x_i \geq 0 \, \forall i \in \mathsf{N} \right\}. \tag{1.2}$$

Thus, the state space of the whole population, the Γ-fold cartesian product, can be denoted by Δ_N^Γ.

Selection operating on the genotypes within demes is based on fitness values, which represent the probability of survival until reproductive age (viability selection). We denote the fitness of the genotype ij in deme α by $w_{ij,\alpha}$, assuming $w_{ij,\alpha} \geq 0$ and independence of the order of the alleles, i.e., $w_{ij,\alpha} = w_{ji,\alpha}$.

Then, the *marginal fitness* of gamete i in deme α and the *mean fitness* in deme α are given by

$$w_{i,\alpha} = \sum_j w_{ij,\alpha} p_{j,\alpha}, \tag{1.3a}$$

$$\bar{w}_\alpha = \sum_i w_{i,\alpha} p_{i,\alpha}. \tag{1.3b}$$

Let the probability that an individual within deme α immigrated from deme β be given by $m_{\alpha\beta}$, and let $\tilde{m}_{\alpha\beta}$ denote the probability of an individual in deme α migrating to deme β. Obviously, to ensure that every individual of a deme either stays or migrates, the $\Gamma \times \Gamma$ *forward migration* $\tilde{M} = (\tilde{m}_{\alpha\beta})$ and *backward migration* matrix $M = (m_{\alpha\beta})$ are stochastic, i.e., nonnegative and satisfy

$$\sum_\beta \tilde{m}_{\alpha\beta} = 1, \tag{1.4a}$$

and

$$\sum_\beta m_{\alpha\beta} = 1. \tag{1.4b}$$

Clearly, the frequency of the gametes can be calculated from the genotype frequencies by

$$p_{i,\alpha} = \sum_j x_{ij,\alpha}, \tag{1.5}$$

where $x_{ij,\alpha}$ denotes the frequency of the genotype ij in deme α. Since the adult individuals migrate, the frequency of the genotype ij after selection and migration is given by

$$x^*_{ij,\alpha} = \frac{p_{i,\alpha} p_{j,\alpha} w_{ij,\alpha}}{\bar{w}_\alpha} \tag{1.6a}$$

and

$$x^{**}_{ij,\alpha} = \sum_\beta m_{\alpha\beta} x^*_{ij,\beta}, \tag{1.6b}$$

respectively. Then, by using (1.5), we conclude that the frequency of gamete i in deme α after selection and migration is given by

$$p^*_{i,\alpha} = \frac{w_{i,\alpha}}{\bar{w}_\alpha} p_{i,\alpha} \tag{1.7a}$$

and

$$p^{**}_{i,\alpha} = \sum_\beta m_{\alpha\beta} p^*_{i,\beta}, \tag{1.7b}$$

respectively.

Let $R_{i,jl}$ be the probability that during gametogenesis a gamete of type i is formed by the genotypes j and l. Then the gametic frequency after random mating and recombination is

$$p'_{i,\alpha} = \sum_{j,l} R_{i,jl} x^{**}_{jl,\alpha} \qquad \forall i \in \mathsf{N}, \forall \alpha \in \mathsf{G}. \tag{1.8}$$

Inserting (1.6) into (1.8) shows that migration and recombination commute, which is possible due to the assumption of genotype-independent migration. Thus, we can reduce the dynamics to one depending only on gamete frequencies (Bürger, 2009, pp.945)

$$p'_{i,\alpha} = \sum_l \sum_j \sum_\beta R_{i,jl} m_{\alpha\beta} x^*_{jl,\beta} = \sum_\beta m_{\alpha\beta} \sum_l \sum_j R_{i,jl} x^*_{jl,\beta} = \sum_\beta m_{\alpha\beta} p^\#_{i,\beta}, \tag{1.9a}$$

where

$$p^\#_{i,\alpha} = \sum_l \sum_j R_{i,jl} \frac{p_{j,\alpha} p_{l,\alpha} w_{jl,\alpha}}{\bar{w}_\alpha} \tag{1.9b}$$

describes the change in the frequency of gamete i due to selection and recombination in deme α.

Equations (1.9a) and (1.9b) fully describe the dynamics of the population.

It follows that we can represent the order of events in the life cycle (1.1) of our model by

$$\underset{Zygote}{c_\alpha, p_{i,\alpha}} \xrightarrow[\text{Recombination}]{\text{Selection}} \underset{Adult}{c^\#_\alpha, p^\#_{i,\alpha}} \xrightarrow{\text{Migration}} \underset{Adult}{c'_\alpha, p'_{i,\alpha}} \xrightarrow{\text{Reproduction}} \underset{Zygote}{c'_\alpha, p'_{i,\alpha}}, \tag{1.10}$$

where c_α, $c^\#_\alpha$ and c'_α describe the relative deme size of deme α before selection, after selection and recombination, and after migration, respectively.

To simplify notation, we introduce the following vectors:

$$p_i = (p_{i,1}, \dots, p_{i,\Gamma})^T \in \mathbb{R}^\Gamma, \tag{1.11a}$$

$$p_{(\alpha)} = (p_{1,\alpha}, \dots, p_{N,\alpha})^T \in \Delta_N, \tag{1.11b}$$

$$p = (p_{(1)}^T, \dots, p_{(\Gamma)}^T)^T \in \Delta_N^\Gamma. \tag{1.11c}$$

Moreover, note that the frequency of allele $A_{i_k}^{(k)}$ on locus k among gametes in deme α is

$$p_{i_n,\alpha}^{(k)} = \sum_{i|i_n} p_{i,\alpha}, \tag{1.12}$$

where the sum runs over all gametes i, such that the n^{th} locus exhibits allele i_n.

1.2.1 Selection

As stated above, selection acts on the newly formed offspring. Viability of the zygotes is mathematically modeled by fitness values. Fitness values of zygotes depend on their genetic make-up, i.e., the phenotype resulting from the paternal gametes.

The phenotypic value G_{ij} of the individual consisting of the genotype ij depends on the *gametic contributions* g_i and g_j. We assume the absence of *dominance* and of *epistasis* on the level of the trait values, so that G is additive, i.e., the sum of gametic contributions,

$$G_{ij} = g_i + g_j = \sum_{n\in\mathsf{L}} \gamma_{i_n}^{(n)} + \sum_{n\in\mathsf{L}} \gamma_{j_n}^{(n)}, \tag{1.13}$$

where $\gamma_{i_n}^{(n)}$ denotes the *contribution of allele* i_n at locus n,

$$g_i = \sum_{n\in\mathsf{L}} \gamma_{i_n}^{(n)}. \tag{1.14}$$

In general, the genotypic and the phenotypic value may differ, but in the absence of environmental effects, they coincide.

We assume that the fitness of an individual of type ij in deme α is given by a positive function $W_\alpha : \mathbb{R} \to \mathbb{R}^+$, assigning a fitness value to each phenotype,

$$w_{ij,\alpha} = W_\alpha(G_{ij}). \tag{1.15}$$

We refer to W_α as the *fitness function* in deme α. We will be able to define the fitness setup for a population by providing a set of functions $\{W_\alpha : \alpha \in \mathsf{G}\}$ and the additive allelic contributions $\{\gamma_{i_n}^{(n)} : n \in \mathsf{N}, i_n \in \mathsf{I}\}$.

Remark 1.2.1. Note that the absence of epistasis and dominance only applies on the level of the genetic trait here. The fitness values given by the function W_α still may exhibit dominance and epistatic effects (Bürger, 2000; Futuyma, 2005).

We consider three different fitness functions: The *quadratic fitness function* is defined as

$$\Phi(G) := 1 - s(G - P_O)^2, \tag{1.16}$$

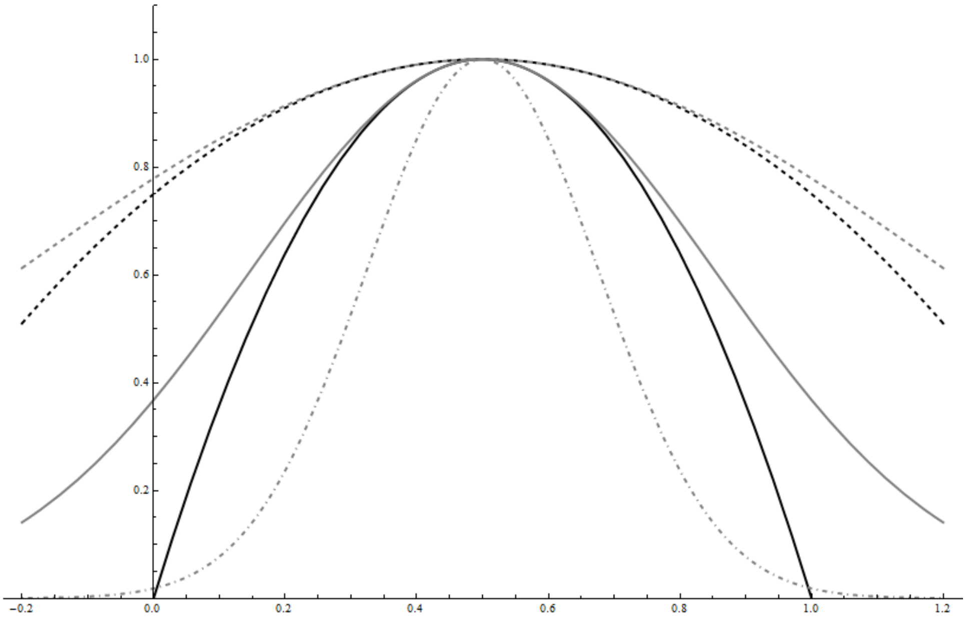


Figure 1.1: Quadratic (black) and Gaussian (gray) optimum functions with $P_O = 0.5$. Dashed lines refer to a selection coefficient $s = 1$, solid lines to $s = 4$ and the dashed-dotted line to $s = 16$. Since we require positive fitness values for the genotypic values between 0 and 1, only the Gaussian is plotted for a selection coefficient of $s = 16$.

where P_O defines the position of the optimum and s is the selection coefficient. Without loss of generality, we may assume that the genotypic values G are constrained by the interval $[\check{G}, \hat{G}]$, where $\check{G} < \hat{G}$. This ensures positivity of the fitness values $\Phi(G)$ for all phenotypes, if s is restricted to the set

$$\mathsf{S}_q = [0, s_{\max}]\,, \text{ where } s_{\max} = \min\left\{\frac{1}{(\check{G} - P_O)^2}, \frac{1}{(\hat{G} - P_O)^2}\right\}, \tag{1.17}$$

and $\check{G}$, $\hat{G}$ respectively, denote the minimum and maximum genotypic values. We will also use the normalized selection coefficient defined by

$$\tilde{s} := \frac{s}{s_{max}}. \tag{1.18}$$

Remark 1.2.2. The quadratic optimum model has been intensively studied, e.g. in Bürger (2000), and originated from Wright's work (Wright, 1935).

The *Gaussian fitness function* is defined as

$$w(G) := \exp(-s(G - P_O)^2), \tag{1.19}$$

where s defines the selection coefficient and P_O the position of the optimum.

The *linear fitness function* is defined as

$$\Psi(G) := 1 + sG, \tag{1.20}$$

where s defines the selection coefficient, which is restricted to

$$\mathsf{S}_l = \left[-s_{\max}, s_{\max}\right], \text{ where } s_{\max} = \min\left\{\frac{1}{|\check{G}|}, \frac{1}{|\hat{G}|}\right\}. \tag{1.21}$$

Remark 1.2.3. Both, the quadratic (1.16) and the Gaussian fitness function (1.19) are used to model *stabilizing selection*, where the maximum value is attained at an intermediate genotypic value (Endler, 1986, p. 177). Both functions are symmetric with respect to the optimum and decrease monotonically from it.

The linear fitness function (1.20) exhibits *directional selection*, i.e., larger trait values are favored. This has also be referred to as *no dominance* on the allelic level, cf. the concept of *addtive inheritance* in (Futuyma, 2005, p. 176, p.282).

Because selection acts on the offspring, the deme sizes may change. We shall use two extreme assumptions here: Soft selection describes a population regulated within each niche, as occurs in the competition for a limiting factor, e.g. space or food. In the case of hard selection, the size of the whole population is controlled, and survival of the individual depends on its absolute fitness regarding the metapopulation (Nagylaki, 1992; Futuyma, 2005).

Soft selection assumes that the relative deme sizes are fixed, i.e.,

$$c_\alpha^* = c_\alpha \; \forall \alpha \in \mathsf{G}. \tag{1.22}$$

Hard selection assumes that the relative deme sizes change proportional to the demes mean fitness, i.e.,

$$c_\alpha^* = c_\alpha \frac{\bar{w}_\alpha}{\bar{w}}, \tag{1.23a}$$

$$\bar{w} = \sum_\alpha c_\alpha \bar{w}_\alpha, \tag{1.23b}$$

where $\bar{w}$ denotes the population's mean fitness.

1.2.2 Recombination

Let the recombination probability $R_{i,jl}$, i.e., the probability that recombination of gametes j and l results in gamete i, be based on independent crossover probabilities, i.e., we negleckt *interference* and *position effects*. Thus, we model $R_{i,jl}$ as the sum of the joint probabilities of recombination events between loci. Given a vector of pairwise recombination probabilities,

$$\rho = (\rho_1, \rho_2, \ldots, \rho_{L-1})^T, \tag{1.24}$$

where ρ_l is the probability of a crossover between locus l and locus $l+1$, we conclude that the recombination probability is given by

$$R_{i,jl} = \sum_{R \in \mathfrak{R}_{i,jl}} \prod_{k \in R} \prod_{n \in R^c} \rho_k (1 - \rho_n), \tag{1.25}$$

where $\mathfrak{R}_{i,jl}$ denotes the set of all recombination events of the multi indices j and l resulting in i. Each recombination event $R \in \mathfrak{R}$ is encoded as a set of indices $R = \{i_1, \dots, i_r\} \subseteq \{1, \dots, L-1\}$, such that for every $i_k \in R$ crossover event occurs between locus k and $k+1$. The complement of R is defined by $R^c := \{1, \dots, L-1\} \backslash R$.

Clearly, the following equations must hold:

$$\sum_i R_{i,jl} = 1, \tag{1.26a}$$

$$R_{i,ii} = 1. \tag{1.26b}$$

Remark 1.2.4. Here we modified the approach by Bürger (2009, 2000) and Nagylaki (1993), using the absence of interference to construct the $R_{i,jl}$ from the recombination probabilities of adjacent loci. This allows to directly use the definition for the implementation, see Section 3.3.2.

1.2.3 Migration

Since we assume no individuals are lost during migration and, both, M and $\tilde{M}$ are stochastic, we observe

$$c_\beta^{**} = \sum_\alpha c_\alpha^* \tilde{m}_{\alpha\beta}, \tag{1.27a}$$

$$c_\alpha^* = \sum_\beta c_\beta^{**} m_{\beta\alpha}. \tag{1.27b}$$

Suppose the backward migration matrix M, and $c^* = (c_1^*, ..., c_\Gamma^*)^T$, the deme sizes after selection, are given. Then, according to (1.27a), we can calculate the new deme sizes after migration.

Also, by using the fact that

$$c_\alpha^* \tilde{m}_{\alpha\beta} = c_\beta^{**} m_{\beta\alpha}, \tag{1.28}$$

and (1.27a), we deduce

$$m_{\beta\alpha} = \frac{c_\alpha^* \tilde{m}_{\alpha\beta}}{\sum_\gamma c_\gamma^* \tilde{m}_{\gamma\beta}}. \tag{1.29}$$

We introduce two specific migration models here, defining the backward migration rates: For the homogeneous Deakin model, the backward migration rates are given by

$$m_{\alpha\beta} = \mu c_\beta^*, \text{ for } \alpha \neq \beta, \tag{1.30a}$$

and

$$m_{\alpha\alpha} = 1 - \mu + \mu c_\alpha^*, \tag{1.30b}$$

where the constant $\mu \in [0,1]$ describes the proportion of outbreeding individuals (Deakin, 1966). The term homogeneous refers to the constant proportion of homing individuals, i.e., homing rates $1 - \mu$. In the case $\mu = 1$ one obtains the *Levene model* (Karlin, 1982, p. 78).

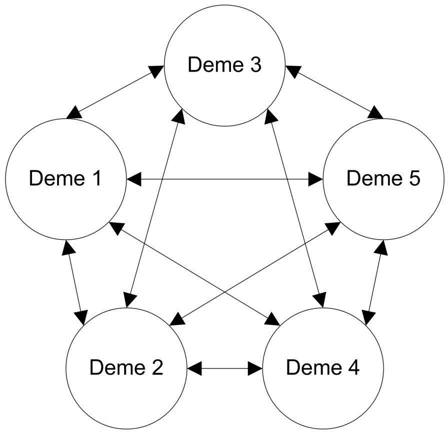


Figure 1.2: Migration according to the Deakin model for 5 demes.

Remark 1.2.5. The homogeneous Deakin model is *conservative* (Nagylaki, 1992, p. 136), i.e., deme proportions stay constant, as can easily be deduced by using (1.27b), definition (1.30), and stochasticity of M:

$$c^*_\alpha = \sum_\beta c^{**}_\beta m_{\beta\alpha} = \sum_{\beta\neq\alpha} c^{**}_\beta \mu c^*_\alpha + c^{**}_\alpha(1-\mu+\mu c^*_\alpha) = \sum_\beta c^{**}_\beta \mu c^*_\alpha + (1-\mu)c^{**}_\alpha = c^{**}_\alpha. \tag{1.31}$$

Thus, in case of soft selection (1.22), the backward migration matrix of the Deakin model is constant.

For the *stepping-stone model*, the backward migration matrix is given by

$$M = \begin{pmatrix} \frac{c^*_1(1-\mu_1)}{d_1} & \frac{c^*_2\mu_2}{d_1} & 0 & \dots & 0 \\ \frac{c^*_1\mu_1}{d_2} & \frac{c^*_2(1-2\mu_2)}{d_2} & \frac{c^*_3\mu_3}{d_2} & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & \frac{c^*_{\Gamma-2}\mu_{\Gamma-2}}{d_{\Gamma-1}} & \frac{c^*_{\Gamma-1}(1-2\mu_{\Gamma-1})}{d_{\Gamma-1}} & \frac{c^*_\Gamma\mu_\Gamma}{d_{\Gamma-1}} \\ 0 & \dots & 0 & \frac{c^*_{\Gamma-1}\mu_{\Gamma-1}}{d_\Gamma} & \frac{c^*_\Gamma(1-\mu_\Gamma)}{d_\Gamma} \end{pmatrix}, \tag{1.32}$$

where

$$d_1 = (1-\mu_1)c^*_1 + \mu_2 c^*_2, \tag{1.33a}$$

$$d_\alpha = \mu_{\alpha-1}c^*_{\alpha-1} + (1-2\mu_\alpha)c^*_\alpha + \mu_{\alpha+1}c^*_{\alpha+1}, \text{ for } 1 < \alpha < \Gamma, \tag{1.33b}$$

and

$$d_\Gamma = \mu_{\Gamma-1}c^*_{\Gamma-1} + (1-\mu_\Gamma)c^*_\Gamma. \tag{1.33c}$$

The constant $\mu_\alpha \in [0, 0.5]$ describes the fraction of individuals migrating from deme α to the neighboring demes. This is an adaptation of the model proposed in Kimura and Weiss (1964), neglecting long-range dispersal and allowing different migration rates for each deme. The denominators d_α are normalization factors to assure the stochasticity of the backward migration matrix.

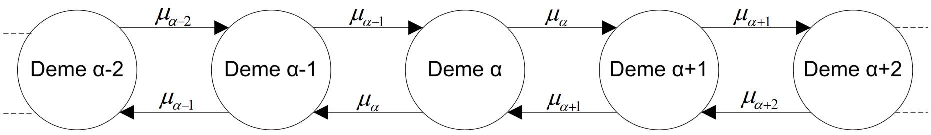


Figure 1.3: Migration according to the stepping-stone model.

Remark 1.2.6. Given the forward migration matrix

$$\tilde{M} = \begin{pmatrix} (1-\mu_1) & \mu_1 & 0 & \dots & 0 \\ \mu_2 & (1-2\mu_2) & \mu_2 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & \mu_{\Gamma-1} & (1-2\mu_{\Gamma-1}) & \mu_{\Gamma-1} \\ 0 & \dots & 0 & \mu_\Gamma & (1-\mu_\Gamma) \end{pmatrix}. \tag{1.34}$$

the backward migration matrix (1.32) follows directly by using (1.29). To reduce the number of parameters involved, we will also use the special case where migration is homogeneous, i.e., $\mu_\alpha = \mu$ for every α. Although (1.32) is a more general model, it still assumes symmetric migration, cf. the *general form* of the stepping stone model as given in (Karlin, 1982, p. 79).

The two migration patterns, (1.30) and (1.32), serve as two extreme prototypes in the analysis. While the Deakin model allows for migration independent of the demes, the stepping-stone model exhibits isolation by distance (Wright, 1943), which "*occurs in subdivided populations, when subpopulations exchange genes at a rate dependent upon the distance, ...*" (Hardy and Vekemans, 1999, p. 1). Figures 1.2 and 1.3 illustrate the differences of these two patterns.

1.2.4 Equilibria

Following (LaSalle, 1976, pp. 1-8), the difference equation (1.9) defines a dynamical system given by

$$p' = T(p), \tag{1.35}$$

where $T : \Delta_N^\Gamma \subset \mathbb{R}^{\Gamma \cdot N} \to \Delta_N^\Gamma \subset \mathbb{R}^{\Gamma \cdot N}$ is continuously differentiable. The set $\{T^n(p) : n \in \mathbb{N}\}$ is called the *orbit* or *trajectory* of p. A point $\hat{p} \in \Delta_N^\Gamma$ is called an *equilibrium point* or *equilibrium state* of the dynamical system if it is a fixed point of the map T, i.e.,

$$\hat{p} = T(\hat{p}). \tag{1.36}$$

Thus, the relative frequencies of the gametes remain constant over generations. In the following, we shall indicate equilibria by the circumflex.

An equilibrium point $\hat{p}$ of T is said to be *stable*, if for a given neighborhood U of $\hat{p}$, there exists a neighborhood W containing U, such that $T^n(W) \subset U$, for all $n \in \mathbb{N}$. Furthermore, an equilibrium point $\hat{p}$ is an *attractor* if there exists a neighborhood U

of $\hat{p}$ such that for each $p \in U$, $T^n(p) \to \hat{p}$ as $n \to \infty$. An attracting, stable equilibrium point is called *asymptotically stable*. Analogously, a set of points can be stable or asymptotically stable. An equilibrium $\hat{p}$ is said to be *hyperbolic*, if no eigenvalue of $T(\hat{p})$ is of modulus 1.

We call an equilibrium *monomorphic* if only one gamete is present at equilibrium, i.e.,

$$\hat{p}_i = (1, \dots, 1) \text{ for a specific } i \in N. \tag{1.37}$$

If more than one gamete is present, the equilibrium is called *polymorhic*. If on every of the L loci, all I alleles are present, we call the equilibrium *fully polymorphic*. Analogously, we distinguish for a single locus, whether it is momomorphic, i.e., only one allele is present, polymorphic, i.e., more than one alleles are present, or fully-polymorphic, i.e., all alleles are present.

1.3 Important Quantities

In the following, we introduce some important quantities measuring various properties of the genetic composition of a population, as well as differentiation among subpopulations.

We want to measure the disparity of the allelic contributions. To achieve this, we need to measure the *effect of allelic substitution*, cf. (Bürger and Gimelfarb, 1999; Nagylaki, 1989): Under the assumption of no dominance and no epistasis, as stated above, see (1.13), the *average effect of allelic substitution* of allele i_k on locus k is given by $|\gamma_{i_k}^{(k)}|$, the contribution of allele i_k on locus k (Bürger, 2000, p. 75). Thus, the *average absolute allelic effect of locus* k is given by

$$\kappa_k = \frac{\sum_{i_k \in \mathsf{I}} |\gamma_{i_k}^{(k)}|}{I}. \tag{1.38}$$

Note the fact that our genetic contributions are deme independent, thus, the measurement defined above, (1.38) is independent of the index α.

We define the *average excess* of genotype ij in deme α as, cf. (Bürger, 2000, pp. 58-59)

$$e_{ij,\alpha} = G_{ij} - \bar{G}_\alpha, \tag{1.39a}$$

where

$$\bar{G}_\alpha = \sum_{ij} G_{ij} x_{ij,\alpha}, \tag{1.39b}$$

denotes the *mean genotypic value* in deme α. Then, the *total genetic*, or *genotypic variance*, in deme α is defined by

$$\sigma^2_{\mathrm{G},\alpha} = \sum_{ij} x_{ij,\alpha} e^2_{ij,\alpha}. \tag{1.40}$$

For the whole metapopulation, we define the *total genetic variance* by

$$\sigma^2_{\mathrm{G}} = \sum_{ij} x_{ij} e^2_{ij}, \tag{1.41a}$$

where

$$e_{ij} = G_{ij} - \sum_{i,j} x_{ij} G_{ij}, \tag{1.41b}$$

and

$$x_{ij} = \sum_{\alpha} x_{ij,\alpha} c_{\alpha}, \tag{1.41c}$$

define the average excess of genotype ij, and the frequency of genotype ij in the metapopulation, respectively.

Analogously, we define the *total gametic variance* in deme α (cf. Ewens, 2004, p. 246), as

$$\sigma^2_{\mathrm{Gam},\alpha} = \sum_i p_{i,\alpha}(w_{i,\alpha} - \bar{w}_{\alpha})^2. \tag{1.42}$$

For the whole metapopulation averaging over the deme sizes yields the *mean gametic variance*:

$$\sigma^2_{\mathrm{Gam}} = \sum_{i,\alpha} c_{\alpha} p_{i,\alpha}(w_{i,\alpha} - \bar{w}_{\alpha})^2. \tag{1.43}$$

Remark 1.3.1. The measures (1.40) and (1.41) depend on the current genetic composition of the population, i.e., the genotype frequencies in the subpopulations. In our simulation, these will be calculated only at equilibrium. Since zygotes are in Hardy-Weinberg proportions, the measures may be expressed in gametic frequencies rather than zygotic frequencies.

To measure the differentiation among the subpopulations, we consider the *variance of gamete i* among subpopulations (Nagylaki, 1992, p. 40),

$$V_i = \overline{p_i^2} - \overline{p_i}^2, \tag{1.44}$$

where

$$\overline{p_i} = \sum_j c_{\alpha} p_{i,\alpha}. \tag{1.45}$$

Averaging over the gametes yields the *average gametic variance among subpopulations*,

$$\bar{V} = \frac{1}{N} \sum_i V_i. \tag{1.46}$$

Moreover, we define the measure Q_{ST} of population differentiation (Spitze, 1993; Edelaar and Björklund, 2011) by

$$Q_{ST} = \frac{\sigma^2_G - \bar{\sigma}^2_{G,\alpha}}{\sigma^2_G}, \tag{1.47}$$

where σ^2_G and $\bar{\sigma}^2_{G,\alpha}$ denote the genetic variance of the trait among the whole population and the mean genetic variance within subpopulations, respectively. Q_{ST} ranges from 0 (no differentiation) to 1 (complete differentiation).

We will measure the *linkage disequilibrium* at equilibrium, following the definition in Nagylaki (1993). The linkage disequilibrium for gamete i in deme α is defined as

$$D_{i,\alpha} = p_{i,\alpha} - \prod_k p^{(k)}_{i_k,\alpha}, \tag{1.48}$$

where the product runs over all alleles, as introduced in (1.12). We define the *average linkage disequilibrium* as

$$\bar{D} = \frac{1}{N\Gamma} \sum_{i,\alpha} |D_{i,\alpha}|. \tag{1.49}$$

Remark 1.3.2. For two loci this definition equals the absolute value of the common linkage disequilibrium $D_{1,\alpha}$, since $\bar{D} = -D_{1,\alpha} = D_{2,\alpha} = D_{3,\alpha} = -D_{4,\alpha}$.

Moreover, in the numerical simulation of our model, the number of alleles present at a single locus at equilibrium, as well as the number of polymorphic and fully polymorphic loci will serve as basic measures of genetic variation, see Table 3.2.

The definition of the following measures are in accordance with Gimelfarb (1998). Let $\tau(\hat{p}), \tau : \Delta_N^\Gamma \to \mathbb{R}$, be the fraction of trajectories converging to the stable equilibrium $\hat{p}$, and let $\lambda(\hat{p}), \lambda : \Delta_N^\Gamma \to \mathsf{L}$, be the number of polymorphic loci in $\hat{p}$. The *polymorphic fraction of the genome* is defined as

$$\frac{1}{L} \sum_{l=0}^{L} l \sum_{\hat{p}:\lambda(\hat{p})=l} \tau(\hat{p}). \tag{1.50}$$

The *expected genetic load for l polymorphic loci* in deme α is given by

$$\sum_{\hat{p}:\lambda(\hat{p})=l} \tau(\hat{p}) \frac{\hat{w}_\alpha - \bar{w}_\alpha}{\hat{w}_\alpha}, \tag{1.51}$$

where $\hat{w}_\alpha = \max_{i,j} w_{ij,\alpha}$. By summing over all $l \in \mathsf{L}$, the *total genetic load* can be calculated:

$$\sum_{l \in \mathsf{L}} \sum_{\hat{p}:\lambda(\hat{p})=l} \tau(\hat{p}) \frac{\hat{w}_\alpha - \bar{w}_\alpha}{\hat{w}_\alpha}. \tag{1.52}$$

Remark 1.3.3. (1.50) can be interpreted as the expected fraction of polymorphic loci in the multilocus genome. (1.52) measures the reduction in mean fitness relative to the maximum possible fitness.

2 Related Work

This chapter is intended to serve as both a review of related work and a motivation for the special cases studied in Chapter 4. Every section briefly reviews relevant publications and then discusses some of the results in more detail to provide a basis of knowledge to enable an interpretation of our own numerical results.

2.1 Simulation

In most cases, it is analytically impossible to find the equilibria of the dynamical system (1.9). Typical methods are reduction of dimensionality (alleles, loci, demes) and restriction to special models and additional assumptions (e.g. Levene migration model, linkage equilibrium, non-epistatic fitness). Another approach to gain insight in the qualitative behavior of the systems is by making assumptions that lead to perturbations of limiting cases (e.g. weak selection relative to migration, see Section 2.3.1) or approximations.

In this thesis, we want to pursue a different approach. We simulate the dynamical system numerically for different parameter combinations until reaching an equilibrium state. This allows us to investigate the equilibrium structure for cases that so far are not understood analytically. Our numerical approach, implemented as discussed in Chapter 3, is based on the work by Gimelfarb (1998).

Gimelfarb investigated measures and properties at equilibrium for panmictic multilocus systems. He simulated the system for two to five loci, each for 10 initial values, 6 recombination rates, and 4000 fitness sets. Instead of using a fine grid to cover possible parameter values and initial values, he used a different approach, reducing the computational cost significantly. Fitness values were generated randomly, ensuring a minimal distance between two fitness values for genotypes, and normalizing fitnesses such that the fittest genotype exhibits a fitness of 1 in each set. Recombination rates were selected to cover weak to strong linkage. 10 different initial value sets were also chosen randomly. Gimelfarb compared the expected number of simultaneously stable equilibria, polymorphic fraction of the genome, genetic load, and linkage disequilibrium for two- to five-locus systems, presented for the different recombination rates. His results showed that multi-locus systems can maintain polymorphisms in a large number of loci in the absence of migration or mutation. In Bürger and Gimelfarb (1999) this approach was modified and applied to investigate the effects of quadratic stabilizing selection in multi-locus systems, as will be discussed in Section 2.2.

For the system investigated in this thesis, cf. Chapter 1, to our knowledge, no other comparable software package is available. Thus, we analyzed the implementation of user-oriented tools for forward simulation in the area of population genetics. Over the past years, multiple such tools have been published, based on statistical data analysis, not on the computation of equilibrium structures of dynamical systems. In all the cited publications, finite populations are simulated forward in time to track changes in the composition of a population subject to evolutionary forces such as mutation, selection, recombination, or migration.

To provide the user with an easily applicable program, many authors created user-friendly environments. Parreira et al. (2009) extended the existing *ms* command line application by Hudson (2002) and also implemented a graphical user interface to ease the simulation process for the user. Sanford et al. (2007) even incorporated a web user interface in the program *Mendel's Accountant* to plot the results out-of-the-box. Lambert et al. (2008) used wrapper scripts to automate graphical output in *R*.

To allow extensibility of the program, Guillaume and Rougemont (2006) and O'Fallon (2010) used object-oriented programming. Guillaume and Rougemont (2006) implemented the program *Nemo* in *C++* and allow the user to extend the program via implementing interfaces. The implementation of *TreesimJ* by O'Fallon (2010) in *Java* allows to extend the simulation by implementing a class inheriting from an appropriate base class.

We combined these ideas in our implementation, using an object-oriented approach, allowing extensibility and implementation of a user-friendly interface. Additionally, we also provide out-of-the box analytical tools. Details will be given in Chapter 3.

2.2 Stabilizing Selection

One of the major goals in theoretical population genetics is to identify mechanisms which may account for the high genetic variation observed in quantitative traits in nature. Stabilizing selection is considered as one of the most common forms of selection in nature and is generally assumed to deplete genetic variation. Still, in natural populations, characters which are subject to stabilizing selection often show high genetic variation, c.f. Ridley (2004).

Since the early work of Wright (1935), stabilizing selection towards an intermediate optimum has been investigated using different approaches. His study, which uses a quadratic fitness function acting on a system of two diallelic loci with an additively controlled trait, showed that for pure selection dynamics no stable full polymorphism can exist. Hastings (1987) even extended his result to stabilizing selection and arbitrary recombination, but still assuming symmetric effects of the loci, as has Wright before him. His result is based on the fact that the system leads to a special case of the symmetric viability model, studied by Karlin and Feldman (1970).

Other publications allowing arbitrary effects of the loci supported the view that stabilizing selection can maintain high genetic variation. Gale and Kearsey (1968) used a triangular fitness function to model stabilizing selection and showed that all loci can be stably polymorphic, provided the disparity of the effects of the loci is high enough. When recombination becomes weaker, the necessary diversity of allelic effects for existence of a stable equilibrium becomes smaller. In Kearsey and Gale (1968), the authors were able to find stable equilibria for a three locus system using computer simulations. In both cases selection was assumed to be strong. Nagylaki (1989) investigated the model for two diallelic loci and allowed for arbitrary fitness function monotonically and symmetrically decreasing from the optimum at the value of the double heterozygote. Neglecting linkage disequilibrium, he showed that if the ratio of the effects of the loci exceeds a critical value both loci can be stably polymorphic.

Gavrilets and Hastings (1993) calculated all possible equilibria and their stability properties for a symmetric quadratic fitness function and arbitrary effects of the loci. They proved that if selection is sufficiently strong relative to recombination, a stable polymorphic equilibrium can exist, provided a high enough disparity in the effects of the loci. Moreover, they investigated a model with arbitrary position of the optimum, and showed that for strong selection relative to recombination, polymorphic equilibria exist.

Bürger and Gimelfarb (1999) simulated a model for two to five diallelic loci, as well as arbitrary recombination rates and selection intensity. The results showed that if the number of loci contributing additively to the quantitative trait increases, the maintained genetic variation in the population at equilibrium declines rapidly. This gave further support to the early results, indicating that in multi-locus systems stabilizing selection as the only evolutionary force can not account for high genetic variation.

2.2.1 Symmetric Model - Optimum at the Double Heterozygote

We discuss the symmetric viability model for 2 alleles, 2 loci and unequal allelic effects. The dynamics are given by (1.9), the fitness function is defined by

$$W(G) = 1 - s\left(G - \frac{1}{2}\right)^2 . \tag{2.1}$$

We further assume, as explained in Section 3.3.3, that the average contributions of the alleles are given by $\bar{\gamma}_1 = 0$ and $\bar{\gamma}_2 = \gamma_1 + \gamma_2 = \frac{1}{2}$. For brevity and consistency with other publications, we use the notation γ_i for $\gamma_2^{(i)}$. Then, if we denote the alleles at the two loci by A and B, the fitness values are given by

$$\begin{array}{c} \\ A_1A_1 \\ A_1A_2 \\ A_2A_2 \end{array} \begin{array}{c} \begin{array}{ccc} B_1B_1 & B_1B_2 & B_2B_2 \end{array} \\ \begin{pmatrix} 1-d & 1-b & 1-a \\ 1-c & 1 & 1-c \\ 1-a & 1-b & 1-d \end{pmatrix} \end{array}, \tag{2.2}$$

where $a = s\left(\gamma_1 - \gamma_2\right)^2$, $b = s\gamma_1^2$, $c = s\gamma_2^2$, and $d = s\left(\gamma_1 + \gamma_2\right)^2 = \frac{1}{4}s$. The symmetric viability model was studied by Karlin and Feldman (1970). We use the analysis by Gavrilets and Hastings (1993) and further investigations on quantitative characters by Bürger and Gimelfarb (1999). We briefly discuss their findings here.

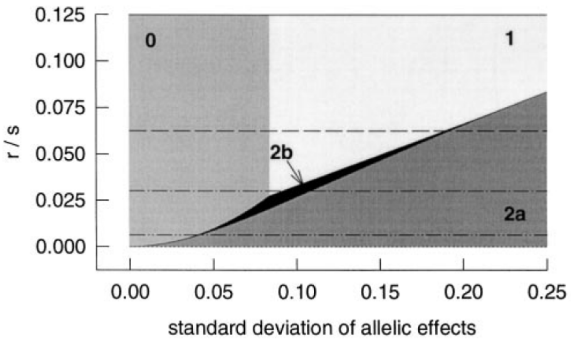


Figure 2.1: Stability of the equilibria for the symmetric model in dependency on the standard deviation of allelic effects and the ratio of recombination to selection. 0: two monomorphic equilibria; 1: two equilibria with the major locus polymorphic; 2a: two asymmetric fully polymorphic equilibria; 2a: one symmetric fully polymorphic equilirbium. For strongest possible selection $s = 4$ the shown parameter region $\frac{r}{s} < 0.125$ covers the whole range. (From Bürger and Gimelfarb (1999))

The system may exhibit four types of equilibria: two monomorphic equillibria for fixation of A_1B_2 and A_2B_1, two major-locus polymorphisms, two fully-polymorphic asymmetric equilibria and one symmetric equilibrium with both loci polymorphic.

Classical analyses studied existence and stability in dependence on the ratio of allelic effects $\frac{\gamma_2}{\gamma_1}$ as a parameter. Bürger and Gimelfarb used the standard deviation of average allelic effects, denoted as σ_γ from now on, which allows for comparison with results for arbitrary loci. In our implementation we followed their suggestion, cf. Section 3.2.2.

Figure 2.1 shows the stability regions of the different equilbria. Stability of equilibria depends on the strength of recombination relative to selection and on the disparity of allelic effects. The more similar the average allelic effects of the loci are, the tighter linkage must be to allow for existence and stability of a fully-polymorphic equilibrium. Even if recombination becomes stronger relative to selection, at least one locus can be maintained polymorphic at equilibrium, if the disparity in allelic effects is big enough.

2.3 Migration

In nature, many populations are spatially structured and gene flow occurs among the subpopulations by migration. Different fitness schemes can act on the subpopulations due to environmental differences. Since the early work (Wright, 1943; Kimura and Weiss, 1964), population division in discrete demes has been modeled. As already stated above, for this work, we restrict ourselves to two special cases of migration, the Deakin model and the stepping-stone model. For a review of the single locus results regarding these two specific migration schemes, refer to Karlin (1982). Here, we state general results on the limiting case applying to a system of arbitrary number of loci and alleles.

2.3.1 Strong Migration

Here we present a result for the case of strong migration relative to selection derived by Bürger (2009). If selection is absent, the dynamics can be described by the weak selection limit, a system of differential equations of deme-independent averaged allele frequencies. Moreover, the system converges to spatial homogeneity at a geometric rate. By introducing weak selection, the dynamics of strong migration can then be understood as a pertubation of the weak-selection limit.

Suppose the backward migration matrix M is constant and ergodic, i.e., irreducible and aperiodic, then there exists a principal left eigenvector μ of M to the eigenvalue 1, such that

$$\mu^T M = \mu^T, \text{and } 1 > |\lambda|, \text{ for all other eigenvalues } \lambda \text{ of } M. \tag{2.3}$$

μ is the stationary distribution of the Markov chain defined by the transition matrix M. By the convergence theorem for ergodic matrices, it follows that for every κ with $|\lambda_0| < \kappa < 1$, where λ_0 is a simple eigenvalue of M,

$$||M^t z - e\mu^T z|| \leq c_z \kappa^t, \tag{2.4}$$

where c_z is independent of t and $e = (1, \dots, 1)^T \in \mathbb{R}^\Gamma$. Thus, in the absence of other evolutionary forces, the population composition will converge to the stationary distribution at a geometric rate. Motivated by this observation, we average the gamete frequencies with respect to the stationary distribution μ:

$$P_i = \mu^T p_i \text{ and } P = (P_1, ..., P_N)^T \in \Delta_N, \tag{2.5}$$

and measure the heterogeneity between demes by defining

$$q_{i,\alpha} = p_{i,\alpha} - P_i, \tag{2.6a}$$
$$q_i = (q_1, ..., q_\Gamma), \tag{2.6b}$$
$$q_{(\alpha)} = p_{(\alpha)} - P, \tag{2.6c}$$
$$q = (q_{(1)}^T, ...q_{(\Gamma)}^T). \tag{2.6d}$$

Recall that p_i and $p_{(\alpha)}$ were defined in (1.11).

To model weak selection, we define the fitness differences to be small compared to migration and recombination:

$$w_{ij,\alpha} = 1 + \epsilon r_{ij,\alpha} \tag{2.7a}$$

for $\epsilon > 0$ small and $|r_{ij,\alpha}| < 1$. From (1.3a) and (1.3b) it follows that

$$w_{i,\alpha}\left(p_{(\alpha)}\right) = 1 + \epsilon r_{i,\alpha}\left(p_{(\alpha)}\right), \text{ and } \bar{w}_\alpha\left(p_{(\alpha)}\right) = 1 + \epsilon \bar{r}_\alpha\left(p_{(\alpha)}\right). \tag{2.7b}$$

Here,

$$r_{i,\alpha}\left(p_{(\alpha)}\right) = \sum_j r_{ij,\alpha} p_{j,\alpha}, \text{ and } \bar{r}_\alpha\left(p_{(\alpha)}\right) = \sum_i r_{i,\alpha}\left(p_{(\alpha)}\right) p_{i,\alpha}. \tag{2.7c}$$

We denote the set of the equilibria of the dynamical system (1.9), with the fitnesses given above, by $\Xi_\epsilon \subset \Delta_N^\Gamma$. We average the allele frequencies at every locus according to our stationary distribution and define the vector

$$\pi = \left(P_1^{(1)}, ..., P_N^{(1)}, ..., P_1^{(N)}, ..., P_N^{(N)}\right)^T. \tag{2.8}$$

We set the average selection coefficients of allele i_n at locus n and the entire population to be

$$\omega_{i_n}^{(n)}(\pi) = \sum_{\alpha} \mu_\alpha r_{i_n,\alpha}^{(n)}(\pi) = \sum_{\alpha} \mu_\alpha \sum_{i|i_n} r_{i,\alpha}(\pi) \prod_{k\neq n} p_{i_k,\alpha}^{(k)}. \tag{2.9a}$$

$$\bar{\omega}(\pi) = \sum_{\alpha} \mu_\alpha \bar{r}_\alpha(\pi). \tag{2.9b}$$

The *weak-selection limit* of our dynamical system (1.9) is then given by

$$\frac{dP_{i_n}^{(n)}}{dt} = P_{i_n}^{(n)} \left(\omega_{i_n}^{(n)}(\pi) - \bar{\omega}(\pi) \right), \tag{2.10a}$$

$$q = 0. \tag{2.10b}$$

We denote the set of the equilibria of the weak selection limit by $\Xi_0 \subset \Delta_N^\Gamma$.

Theorem 2.3.1 (Weak Selection)**.** *Suppose the backward migration matrix M is constant and ergodic and all equilibria of* (2.10) *are hyperbolic, the recombination rates $R_{i,jl}$ are fixed, and $\epsilon > 0$ is sufficiently small.*

(i) The sets Ξ_0 and Ξ_ϵ contain only isolated points. As $\epsilon \to 0$, each equilibrium in Ξ_0 converges to the corresponding equilibrium in Ξ_ϵ.

(ii) In the neighborhood of each equilibrium in Ξ_0, there exists exactly one corresponding equilibrium in Ξ_ϵ. The stability of the corresponding equilibrium in Ξ_ϵ is the same of the corresponding equilibrium in Ξ_0.

(iii) Every solution of (1.9) *with fitnesses given in* (2.7a) *converges to one of the equilibrium points in Ξ_ϵ.*

Remark 2.3.1. Recall that soft selection exhibits a constant migration matrix M, while hard selection does not. Thus, the theorem applies to the former. Hyperbolicity of an equilibrium was defined in Section 1.2.4. Statement (ii) of Theorem 2.3.1 tells us that none of the boundary equilibria moves outside the simplex Δ_N^Γ. Statement (iii) allows us to conclude that no complicated dynamics can occur, such as cycling, since all trajectories converge.

3 The Simulation Approach

The model proposed in Chapter 1 was implemented using an object-oriented approach in *C#*, *.Net 4.0*. The goal was not only to provide a one-time solution to numerically study equilibrium properties for selected migration-selection dynamics, but to build a program, which may be adapted and extended to different scenarios. The main ansatz was to use the life cycle given in (1.10).

We implemented two applications: one for simulation and one for analysis. The *Simulator* allows to setup a scenario and simulates the dynamical system for a given set of parameters and initial values by iteration until a steady state is reached. It stores the final gamete frequencies and quantities of interest in the database. The *Analyzer*, on the other hand, allows to retrieve this information from the database and to create statistics, tables and plots based on it. They are used for the numerical investigations in Chapter 4.

In Section 3.1, we will motivate our ansatz. The implementation of the simulation will be discussed in Section 3.2, where the object model is explained and details on the design of the relational database as well as on the parameter handling and extensibility is provided. Also, the tool implemented to analyze the data will be presented in Section 3.2.5. Then, we present algorithmic details of already implemented scenarios in Section 3.3.

3.1 Modeling the Life Cycle

First, let us summarize the assumptions of the model stated in Chapter 1 which, thus, also apply to our implementation:

- The population is infinitely large and consists of diploid individuals.
- The sexes are equivalent as, e.g., in a monoecious species.
- Generations are non-overlapping and discrete.
- At each locus there is the same number of possible alleles.
- Selection acts based on viability and is independent of the current composition of the population and of time.
- Migration occurs among adults and is independent of genotype.
- Reproduction is random. This ensures Hardy-Weinberg proportions among the zygotes in each deme.

Remark 3.1.1. Clearly, the above stated assumptions allow for arbitrary fitness models. The models stated in Section 1.2.1 only provide a very small catalogue of possibilities. Soft and hard selection, cf. (1.22) and (1.23), although the most common selection models, may be generalized. Also, for the scenarios discussed in this thesis, we assumed the absence of epistasis and dominance, (1.13), however, our implementation is independent of this choice. Migration has to fullfill equations (1.4), and may depend on the deme size. Thus, migration rates may change over time, cf. (1.32). Other obvoius choices for migration models would be the *island model*, *circulant* or *directional migration*, the inhomogeneous Deakin model, or a higher-dimensional stepping stone model, cf. Karlin (1982). Also note that our implementation does not apriori restrict the number of alleles and loci. However, these numbers are restricted by memory and processing capacities.

To model and simulate different possible scenarios for the life cycle given in (1.10), we decomposed it into the following parts:

1. Basic simulation settings: The numerical accuracy, the maximum number of generations, memory pool size, allele extinction error.

2. Definition of the number of loci L, the number of alleles I, and the number of demes Γ.

3. Generation of the initial gamete frequencies, to which we will refer as initial values from now on.

4. Setting the initial deme sizes $\{c_\alpha : \alpha \in \mathsf{G}\}$.

5. Generation of the fitness values $\{w_{ij,\alpha} : i, j \in \mathsf{N}, \alpha \in \mathsf{G}\}$ for all possible zygotes.

6. Generation of the recombination probabilities $\{R_{i,jl} : i, j, l \in \mathsf{N}\}$

7. Selection. Adjustment of the deme sizes $c_\alpha \mapsto c_\alpha^\#$ and gamete frequencies $p_{i,\alpha} \mapsto p_{i,\alpha}^\#$.

8. Migration. Generation of the forward and backward migration rates, i.e., $\{m_{\alpha\beta}, \tilde{m}_{\alpha\beta} : \alpha, \beta \in \mathsf{G}\}$. Adjustment of the deme sizes $c_\alpha^\# \mapsto c'_\alpha$ and gamete frequencies $p_{i,\alpha}^\# \mapsto p'_{i,\alpha}$.

One *simulation run* creates results for many different sets of parameters. Typically, the dynamics is computed for various initial gamete frequencies, for multiple migration rates and selection coefficients. For a given migration pattern (e.g. the Deakin model), multiple migration parameters need to be tested. This led to the current approach, which is based on the idea of creating multiple populations where each corresponds to one set of parameters. Such a combination of parameters is represented by a set of *patterns*. We distinguish six types of patterns: initial value patterns, deme patterns, fitness patterns, recombination patterns, selection patterns, and migration patterns. These patterns correspond to steps three to eight of the list above. Step one and two will be fixed for a single simulation run.

3.2 Implementation

3.2.1 The Object Model

Here we discuss the implementation of our simulation approach, all mentioned classes and interfaces and their relations are sketched in Figure 3.1. In the following we use the typewriter font and upper camel notation to refer to classes, class-instances, or methods and properties defined within the implementation.

Every instance of the `Population` class holds a field of `SubPopulations` it is composed of. Every `SubPopulation` holds the current composition of gametes, the relative deme size, and the fitness values including marginal and mean deme fitnesses. Starting a new simulation run results in a new instance of the `Simulation` class, which creates `Populations` based on the defined scenario. The two structs `PopulationDimensions` and `SimulationSettings` correspond to step one and two of the list above and define the basic terms for the simulation run.

Steps three to eight of the list above were mapped to interfaces, providing all necessary input for a population run. Thus, construction of a new `Population` requires the applied patterns, implementing these interfaces. *Generators* allow to create multiple patterns. Each of the patterns is created by a generator, which is implemented to be an enumeration of the corresponding pattern via the `IEnumerable` interace, c.f. Drayton et al. (2003). Table 3.1 provides an overview on the interfaces involved and their purpose. Every pattern interface inherits from the base interface `IPattern`.

For example, a homogeneous Deakin model can be implemented as follows: A `MyDeakinGenerator` class implementing the `IMigrationModelGenerator` interface. Therefore, it provides a graphical user interface to define the set of migration rates $(\mu_1, ...\mu_n)$. The generator creates an instance of the `MyDeakinPattern` class for each migration rate μ_i. The `MyDeakinPattern` class implements the `IMigrationPattern` interface.

An implementation of the `IInitialValueGenerator` may allow to set the number of generated initial values and their euclidian distance. Based on this information, instances implementing the `IInitialValuePattern` interface are constructed and accessible via the generator.

The `Simulation` creates `Populations` based on all possible combinations of patterns provided by the generators. The responsible `StartSimulation` method, implemented by the `Simulation` class, asynchronously starts the `Evolve` method on all `Populations`. This method creates the initial composition of the population and simulates the given dynamics based on the provided patterns; see the pseudo code of Algorithm 1. If the gamete frequencies $\{p_{i,\alpha} : i \in \mathsf{N}, \alpha \in \mathsf{G}\}$ stagnate for a certain number of generations as to the significant decimal place, the population is assumed to have reached an equilibrium. The number of generations and the precision of the stagnation, i.e., significant decimal place, are set by the user. Former is reflected by the `RepeatCount` property, and latter by the `Error` property of the `SimulationSettings`

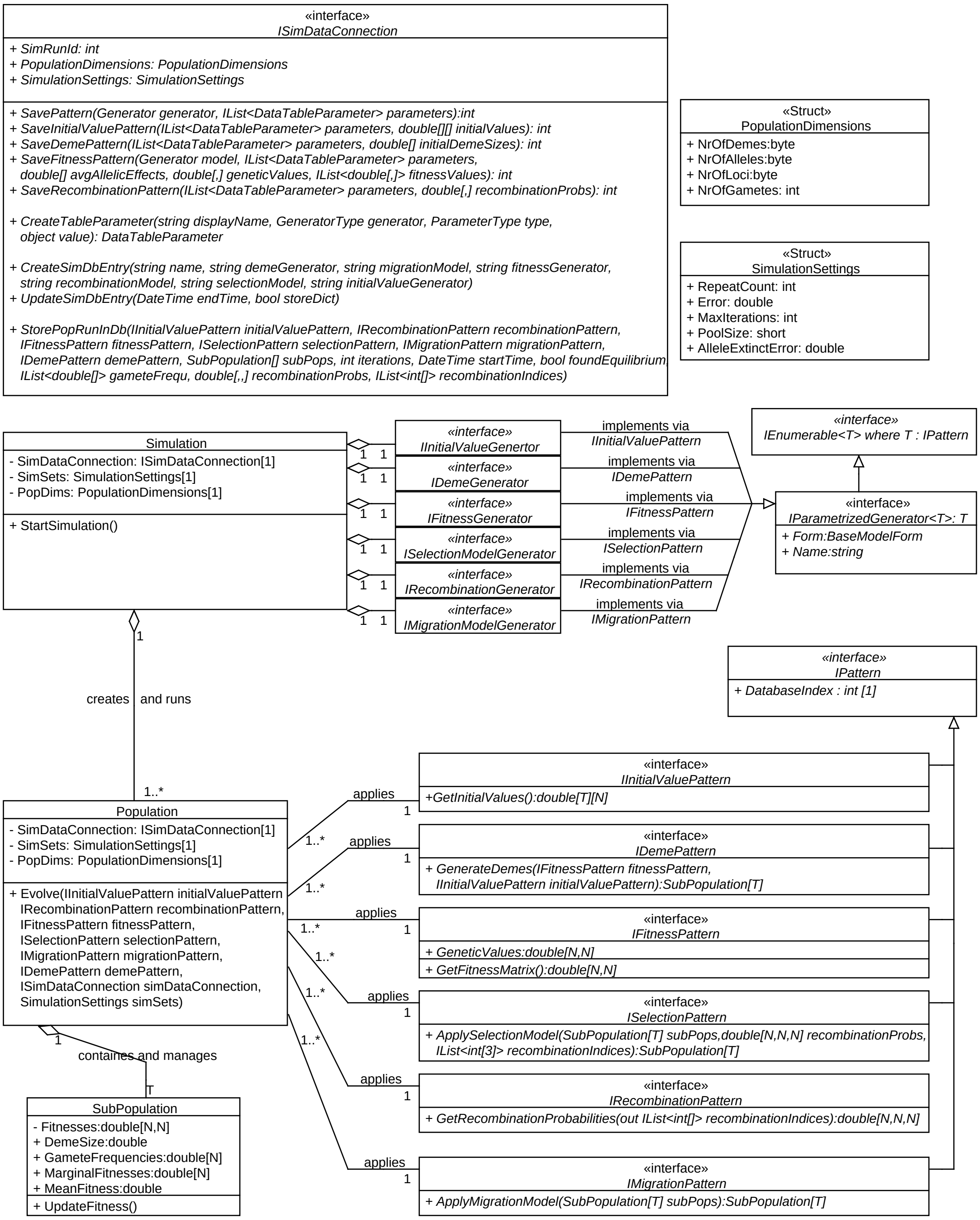

«interface»
ISimDataConnection
+ SimRunId: int
+ PopulationDimensions: PopulationDimensions
+ SimulationSettings: SimulationSettings
+ SavePattern(Generator generator, IList<DataTableParameter> parameters):int
+ SaveInitialValuePattern(IList<DataTableParameter> parameters, double[][] initialValues): int
+ SaveDemePattern(IList<DataTableParameter> parameters, double[] initialDemeSizes): int
+ SaveFitnessPattern(Generator model, IList<DataTableParameter> parameters,
double[] avgAllelicEffects, double[,] geneticValues, IList<double[,]> fitnessValues): int
+ SaveRecombinationPattern(IList<DataTableParameter> parameters, double[,] recombinationProbs): int
+ CreateTableParameter(string displayName, GeneratorType generator, ParameterType type,
object value): DataTableParameter
+ CreateSimDbEntry(string name, string demeGenerator, string migrationModel, string fitnessGenerator,
string recombinationModel, string selectionModel, string initialValueGenerator)
+ UpdateSimDbEntry(DateTime endTime, bool storeDict)
+ StorePopRunInDb(IInitialValuePattern initialValuePattern, IRecombinationPattern recombinationPattern,
IFitnessPattern fitnessPattern, ISelectionPattern selectionPattern, IMigrationPattern migrationPattern,
IDemePattern demePattern, SubPopulation[] subPops, int iterations, DateTime startTime, bool foundEquilibrium,
IList<double[]> gameteFrequ, double[,,] recombinationProbs, IList<int[]> recombinationIndices)
«Struct»
PopulationDimensions
+ NrOfDemes:byte
+ NrOfAlleles:byte
+ NrOfLoci:byte
+ NrOfGametes: int
«Struct»
SimulationSettings
+ RepeatCount: int
+ Error: double
+ MaxIterations: int
+ PoolSize: short
+ AlleleExtinctError: double
Simulation
- SimDataConnection: ISimDataConnection[1]
- SimSets: SimulationSettings[1]
- PopDims: PopulationDimensions[1]
+ StartSimulation()
1 1
«interface»
IInitialValueGenertor
«interface»
IDemeGenerator
«interface»
IFitnessGenerator
«interface»
ISelectionModelGenerator
«interface»
IRecombinationGenerator
«interface»
IMigrationModelGenerator
implements via
IInitialValuePattern
implements via
IDemePattern
implements via
IFitnessPattern
implements via
ISelectionPattern
implements via
IRecombinationPattern
implements via
IMigrationPattern
«interface»
IEnumerable<T> where T : IPattern
«interface»
IParametrizedGenerator<T>: T
+ Form:BaseModelForm
+ Name:string
«interface»
IPattern
+ DatabaseIndex : int [1]
1
creates
and runs
1..*
Population
- SimDataConnection: ISimDataConnection[1]
- SimSets: SimulationSettings[1]
- PopDims: PopulationDimensions[1]
+ Evolve(IInitialValuePattern initialValuePattern
IRecombinationPattern recombinationPattern,
IFitnessPattern fitnessPattern,
ISelectionPattern selectionPattern,
IMigrationPattern migrationPattern,
IDemePattern demePattern,
ISimDataConnection simDataConnection,
SimulationSettings simSets)
applies
1
1..*
«interface»
IInitialValuePattern
+GetInitialValues():double[T][N]
«interface»
IDemePattern
+ GenerateDemes(IFitnessPattern fitnessPattern,
IInitialValuePattern initialValuePattern):SubPopulation[T]
«interface»
IFitnessPattern
+ GeneticValues:double[N,N]
+ GetFitnessMatrix():double[N,N]
«interface»
ISelectionPattern
+ ApplySelectionModel(SubPopulation[T] subPops,double[N,N,N] recombinationProbs,
IList<int[3]> recombinationIndices):SubPopulation[T]
«interface»
IRecombinationPattern
+ GetRecombinationProbabilities(out IList<int[]> recombinationIndices):double[N,N,N]
«interface»
IMigrationPattern
+ ApplyMigrationModel(SubPopulation[T] subPops):SubPopulation[T]
1
containes and manages
T
SubPopulation
- Fitnesses:double[N,N]
+ DemeSize:double
+ GameteFrequencies:double[N]
+ MarginalFitnesses:double[N]
+ MeanFitness:double
+ UpdateFitness()

Figure 3.1: Simplified UML representation of the main classes and interfaces. Please note that the number of demes is denoted by T instead of Γ as usual.

IParametrizedGenerator	IPattern	**Pattern Description**
IInitialValueGenerator	IInitialValuePattern	Provides the initial gamete frequencies.
IFitnessGenerator	IFitnessPattern	Generates the fitness values for all possible combination of gametes, i.e., the zygotes.
IDemeGenerator	IDemePattern	Generates the initial subpopulations based on the initial gamete frequencies and the fitness values.
IRecombinationGenerator	IRecombinationPattern	Provides all possible recombinations and their probabilities.
ISelectionModelGenerator	ISelectionPattern	Updates the gamete frequencies and the deme sizes depending on the provided recombination probabilities, fitness values, and current population composition.
IMigrationModelGenerator	IMigrationPattern	Updates the gamete frequencies and deme sizes depending on the current population composition.

Table 3.1: The `IParametrizedGenerator` interfaces, their corresponding `IPattern` interfaces, and the patterns function within the life cycle.

class. The `MaxIterations` property sets the maximal number of iterations before the population run is aborted.

Our implementation allows for parallel computing on multi-core systems. Using the `System.Threading.ThreadPool` class provided by the *.Net Framework*, (Drayton et al., 2003), `Populations` created by the `Simulation` are executed parallel on the CPUs. The maximum number of parallel instanced and executed `Populations` is set by the `SimulationSettings.PoolSize` property.

3.2.2 The Database

To store the data gathered by simulation runs, a *Microsoft SQL Server 2008 Express* instance was used, utilizing the *LINQ to SQL* technology to create an object relational mapping, (Kansy, 2010). Via the `ISimDataConnection` interface, the `Simulation`, `Population`, and all patterns share an instance handling all database related issues, as depicted in Figure 3.1. The `ISimDataConnection` interface provides methods to store information on simulations, populations and patterns in the database. The used relational database comprises one table for the simulation runs, one for the created populations, and one for each type of patterns to minimize redundancy, please refer to Figure 3.2.

Algorithm 1 Population.Evolve

Description: Starts the simulation of the population. Calls fitness calculations, selection and migration procedures, checks for an equilibrium and stores results in the database as soon as equilibrium was found or iterations exceed the maximum number.

Input: *initialValuePattern, demePattern, fitnessPattern, recombinationPattern, selectionPattern, migrationPattern* (IPattern): The set of patterns.
simDataConnection (ISimDataConnection): The connection to the database.
simSets (SimulationSettings): The basic simulation settings.

Output: void

```
IList<double[ ]> $recombinationIndices$; //setup the population
double[ , , ] $recProbs = recombinationPattern.$
    $GetRecombinationProbabilities($out $recombinationIndices)$;
IList<SubPopulation> $subPops =$
    $demePattern.GenerateDemes(fitnessPattern, initialValuePattern)$;
bool $foundEquilibrium = false$; //set initial variables
int $stagnationCount = 0$;
int $iterations = 0$;
for $iterations = 1$ to $simSettings.MaxIterations$ do //life cycle iteration
    List<double[ ]> $startGameteFreq = CopyGameteFrequencies(subPops)$;
    for each $subPop$ in $subPops$ do //update all fitness values
        $subPop.UpdateFitness()$;
    end for
    $selectionPattern.ApplySelectionModel(ref subPops, recProbs,$
    $recombinationIndices)$;
    $migrationPattern.ApplyMigrationModel(ref subPops)$;
    if $GameteFrequDiffer(startGameteFreq, subPops, simSets.Error)$ then
    //check for stagnation
        $stagnationCount = 0$;
    else
        $stagnationCount$++;
        if $stagnationCount \geq simSets.RepeatCount$ then
            $foundEquilibrium = true$;
            break; //end loop and store population
        end if
    end if
end for
$simDataConnection.StorePopRunInDb(initialValuePattern, demePattern,$
$recombinationPattern, fitnessPattern, selectionPattern, migrationPattern,$
$subPops, recProbs, recombinationIndices, iterations, foundEquilibrium)$;
```

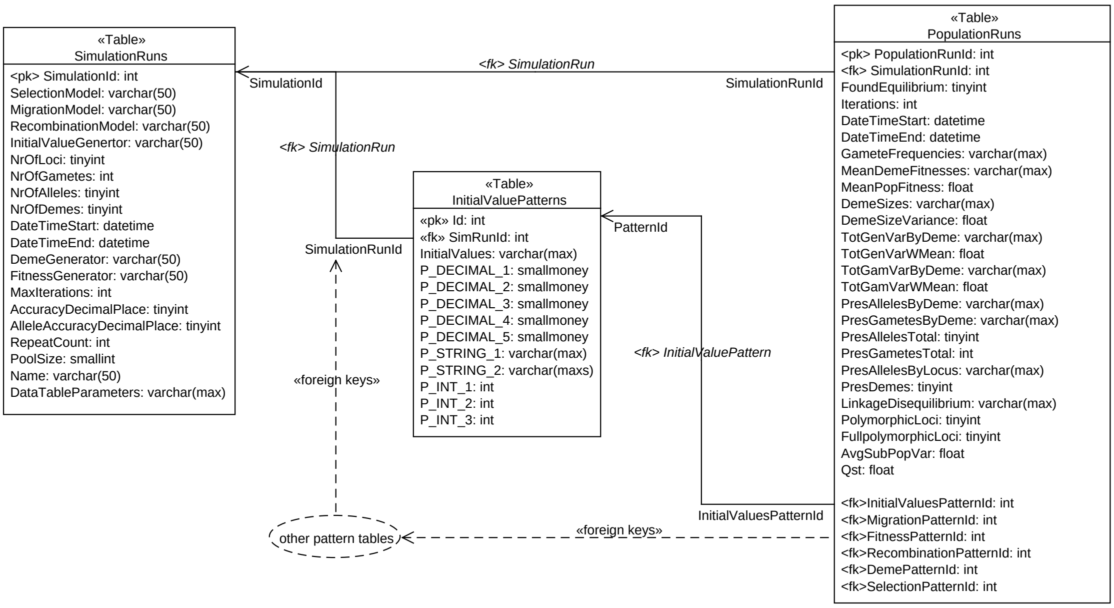


Figure 3.2: The design of the relational database to store information on simulation runs. We only represented the table for the initial value patterns, and ommited the other pattern's tables for clarity. Analogously, all pattern tables are referenced by a foreign key within the *PopulationRuns* table and contain a foreign key to the *SimulationRuns* table, as denoted by the dashed lines. Moreover, all patterns share the parameter columns, starting with 'P', as well as the columns 'Id' and 'SimRunId', defining the primary and foreign key, respectively.

A simulation run table entry holds the population dimensions, the simulation settings, information on the used generators and a serialization of a dictionary on the *data table parameters*, which will be discussed in more detail later in Section 3.2.3.

All pattern tables hold various columns for data table parameters of different type (`varchar, int, smallmoney`) and may hold additional dimensions, as for example, the *InitialValuePatterns* holds a column of type `varchar` named *InitialValues* to store the serialized initial gamete frequencies. The general and special columns of all parameter tables are listed in Table 3.2.

The *PopulationRuns* table references the set of patterns and the simulation run via foreign keys. Additionally, information on the population composition and measures, as defined in Section 1.3, are stored in the columns, see Table 3.3. These measures are calculated and stored in the database by calling the `StorePopRunInDb` method on the `ISimDataConnection` class, see Figure 3.1.

3.2.3 Dynamic Parameter Handling

Every generator may exhibit multiple *parameters*. Internally, a `DataTableParameter` has a type, either `Int`, `Decimal`, or `String`, or `Double[]`, as well as a display name, and refers to a specific generator. Construction of a new pattern then necessitates a list of

Table	Column	Description
All Pattern Tables	Id:int	The primary key.
	SimRunId:int	The foreign key to the simulation run.
	P_String_*: varchar(max)	Parameter columns of type varchar, string, respectively.
	P_Decimal_*: smallmoney	Parameter columns of type smallmoney, decimal, respectively.
	P_Int_*: int	Parameter columns of type int.
InitialValuePatterns	InitialValues: varchar(max)	Serialized initial gamete frequencies.
FitnessPatterns	GeneticValues: varchar(max)	Serialized genetic values for all zygotes.$\{G_{ij} : i, j \in \mathsf{N}\}$
	FitnessValues: varchar(max)	Serialized fitness values for all zygotes in all demes. $\{w_{ij,\alpha} : i, j \in \mathsf{N}, \alpha \in \mathsf{G}\}$
	AvgAllelicEffects:varchar(max)	Serialized allelic effects of all loci.$\{\kappa_k : k \in \mathsf{I}\}$
	AvgAllelicEffect: float	Mean average allelic effect. $\bar{\kappa} = \sum_k \kappa_k / I$
	AvgAllelicEffectStDev: float	Standard deviation of average allelic effects. $\sigma_{\bar{\kappa}} = \sqrt{\sum_k (\kappa_k - \bar{\kappa})^2 / I}$
RecombinationPatterns	RecombinationProbabilities: varchar(max)	Serialized recombination probabilities, $\{R_{i,jl} : i, j, l \in \mathsf{N}\}$.
DemePatterns	InitialDemeSizes: varchar(max)	Serialized initial deme sizes, $\{c_\alpha : \alpha \in \mathsf{G}\}$.

Table 3.2: Pattern table columns.

Column	Description & Formula
AvgLinkageDisequilibrium:float	Average linkage disequilibrium $\bar{D}$, as defined in (1.49).
AvgSubPopVar:float	The average variance of gametes within subpopulations $\bar{V}$, as defined in (1.46).
DemeSizes:varchar(max)	Serialized deme sizes. $\{c_\alpha : \alpha \in \mathsf{G}\}$
FoundEquilibrium:tinyint	1 if an equilibrium was reached, 0 else.
FullPolymorphicLoci:tinyint	Total number of full polymorphic loci. $\mathrm{card}(\{k \in \mathsf{L} : p^{(k)}_{i_k} > \epsilon_a$for all $i_k \in \mathsf{I}\})$
GameteFrequencies:varchar(max)	Serialized gamete frequencies. $\{p_{i,\alpha} : i \in \mathsf{N}, \alpha \in \mathsf{G}\}$
Iterations:int	The number of simulated generations until equilibrium state or abortion.
LinkageDisequilibrium:varchar(max)	Serialization of the linkage disequilibrium coefficients stored separately for all demes and gametes. $\{D_{i,\alpha} : i \in \mathsf{N}, \alpha \in \mathsf{G}\}$
MeanDemeFitness:varchar(max)	Serialized mean deme fitnesses. $\{\bar{w}_\alpha : \alpha \in \mathsf{G}\}$
MeanPopFitness:float	Mean population fitness, $\bar{w}$.
PresAllelesByDeme:varchar(max)	Serialized numbers of present alleles within all demes. $\{\mathrm{card}(\{p^{(k)}_{i_k,\alpha} > \epsilon_A : i_k \in \mathsf{I}, k \in \mathsf{L}\}) : \alpha \in \mathsf{G}\}$
PresAllelesByLocus:varchar(max)	Serialized numbers of present alleles by locus. $\{\mathrm{card}(\{p^{(k)}_{i_k,\alpha} > \epsilon_A : i_k \in \mathsf{I}, \alpha \in \mathsf{G}\}) : k \in \mathsf{L}\}$
PresAllelesTotal:int	The total number of present alleles. $\mathrm{card}(\{p^{(k)}_{i_k,\alpha} > \epsilon_A : i_k \in \mathsf{I}, k \in \mathsf{L}\, \alpha \in \mathsf{G}\}$
PresDemes:tinyint	Number of demes with a positive relative size. $\mathrm{card}(\{c_\alpha : c_\alpha > \epsilon_D\})$
PresGametesByDeme:varchar(max)	Serialized numbers of present gametes within all demes. $\{\mathrm{card}(\{p_{i,\alpha} > \epsilon_A : i \in \mathsf{N}\}) : \alpha \in \mathsf{G}\}$
PresGametesTotal:int	Total number of present gametes. $\mathrm{card}(\{p_{i,\alpha} > \epsilon_A : i \in \mathsf{N}, \alpha \in \mathsf{G}\})$
PolymorphicLoci:tinyint	Total number of polymorphic loci. $\mathrm{card}(\{k \in \mathsf{L} : p^{(k)}_{i_k} > \epsilon_a$for at least two$i_k\})$
QST:float	Q_{ST} as defined in (1.47).
TotalGamVarByDeme:varchar(max)	Serialized total gametic variances within all demes. $\{\sigma^2_{\mathrm{Gam},\alpha} : \alpha \in \mathsf{G}\}$
TotalGamVarWMean:float	The mean total gametic variance, σ^2_{Gam}.
TotalGenVarByDeme:varchar(max)	Serialized genetic variances within all demes. $\{\sigma^2_{G,\alpha} : \alpha \in \mathsf{G}\}$
TotalGenVar:float	The total genetic variance in the metapopulation, σ^2_{G}, as defined in (1.41).

Table 3.3: Description of the measure columns contained in the *PopulationRuns* table. Here, ϵ_D and ϵ_A denote the error defined by the `SimulationSettings.Error` property and the `SimulationSettings.AlleleExtinctError` property, respectively.

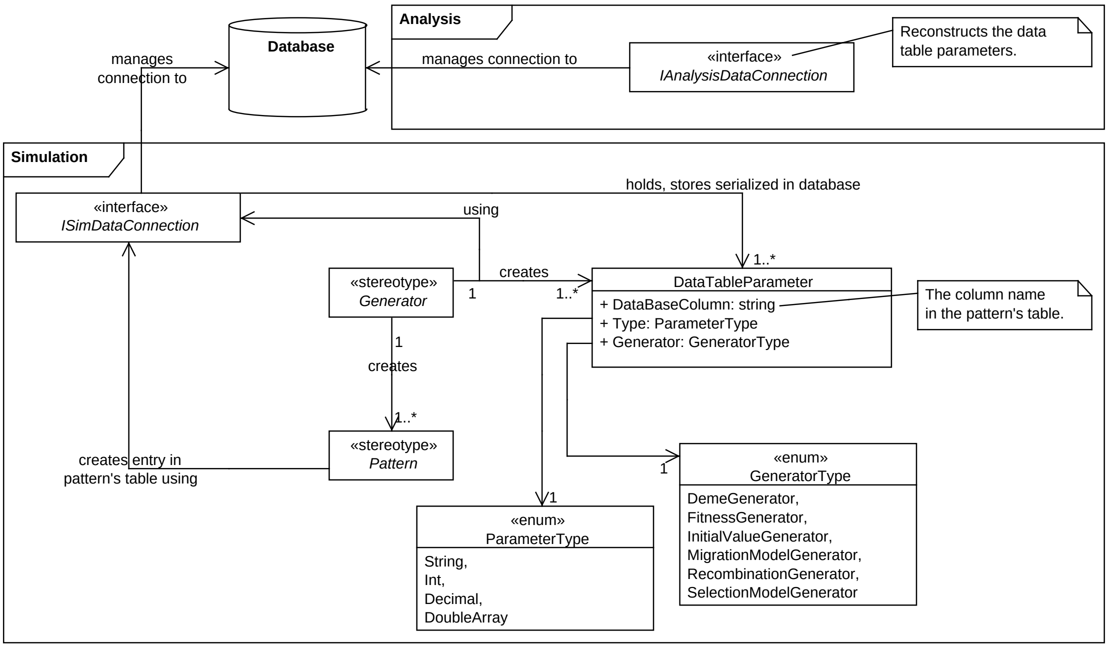


Figure 3.3: The `DataTableParameter` class. The *generator* and *pattern* stereotypes represent instances of the different specializations of the `IParametrizedGenerator`, and `IPattern` interfaces.

parameter values, representing the current parameter set of the pattern. For example, in the case of a quadratic fitness generator, a typical parameter set would be given by the selection coefficient, the position of the optimum, and allelic contributions. The resulting pattern will then apply one such specific parameter set. We need to keep track of the used parameters. This way, we can retrieve information on the parameters used for a single population run for the analysis later on. The database tables for the patterns provide rows to store the parameter values and every stored simulation run holds a serialized dictionary of the parameter mapping. Using our approach of centralized management of the parameters by the `ISimDataConnection` interface, new parameters are created dynamically during runtime and all information regarding the used parameter set may be retrieved during analysis. This way, new generators and patterns, providing different parameters, may easily be implemented. The analysis tool then reconstructs the parameters and allows to create plots and tables based on them. Figure 3.3 shows the workflow regarding creation and handling of the `DataTableParameters` and details on their implementation.

3.2.4 User Interface

The simulator provides a simple user interface to set up new simulation runs. To embed new implementations of generators and patterns, every `IParametrizedGenerator` implementation also may provide a `Windows.Form`, to be integrated as user interface. If an implementation of the `IParametrizedGenerator` interface provides such an user interface, accessible via the `Form` property, see Figure 3.1, it is then integrated automatically into the frontend of the application.

3.2.5 The Analytic Tool

After data collection via simulation, often another program is used to visualize the data properly. Therefore, extraction of the data and data conversion into the desired format has to be accomplished. To provide an easier way to visualize the data of a simulation run, we implemented an analytical tool, from now on simply called the *Analyzer*. The Analyzer can directly access the data stored for the completed simulation runs and provides a plotting functionality, as well as it facilitates the creation of simple tables. This allows the user to quickly navigate through the data to reveal interesting patterns without any overhead on data conversion issues. The basic design of the implementation is sketched in Figure 3.4.

All plots presented in this thesis were generated with the Analyzer Plotter Gui. The tabled data was generated and exported using the Table Gui.

The Plotter allows to adapt the visual depiction via color-coding and shape-coding, labeling, zooming, and scaling. Moreover, it provides functionalities to save the plots in different graphic formats, *.png, .jpg, .gif.* It allows to save and reload plotted data in a proper format. The visualization of the data and user interaction handling uses and is based on the open source project *ZedGraph*[1].

The Table provides export functionality for the free software R[2] and in a *.csv* format.

3.3 Algorithmic Details

This section provides details on algorithms. Section 3.3.1 concerns a general data handling issue. All the other presented algorithms concern implementations of generators and patterns used for the simulations presented in Chapter 4.

3.3.1 Allelic Composition of a Gamete

Our implementation of the `SubPopulation` class represents the gamete frequencies $\{p_{i,\alpha} : i \in \mathsf{N}\}$ for deme α as an array of double precision floating point numbers of dimension N. We need to keep track only of the gamete frequencies to apply the selection and migration dynamics. Setting up the fitness values or calculation of recombination rates depends on the *allelic composition of the gamete*, i.e., which allele is present on which locus. Recall that this was described by a multi-index, see Section 1.2.

Thus, we have to be able to switch from one representation to the other, i.e., to calculate the allelic composition from the array-index of the gamete. This mapping is

[1] http://sourceforge.net/projects/zedgraph/; visited on December 22nd 2011

[2] http://www.r-project.org; visited on December 22nd 2011

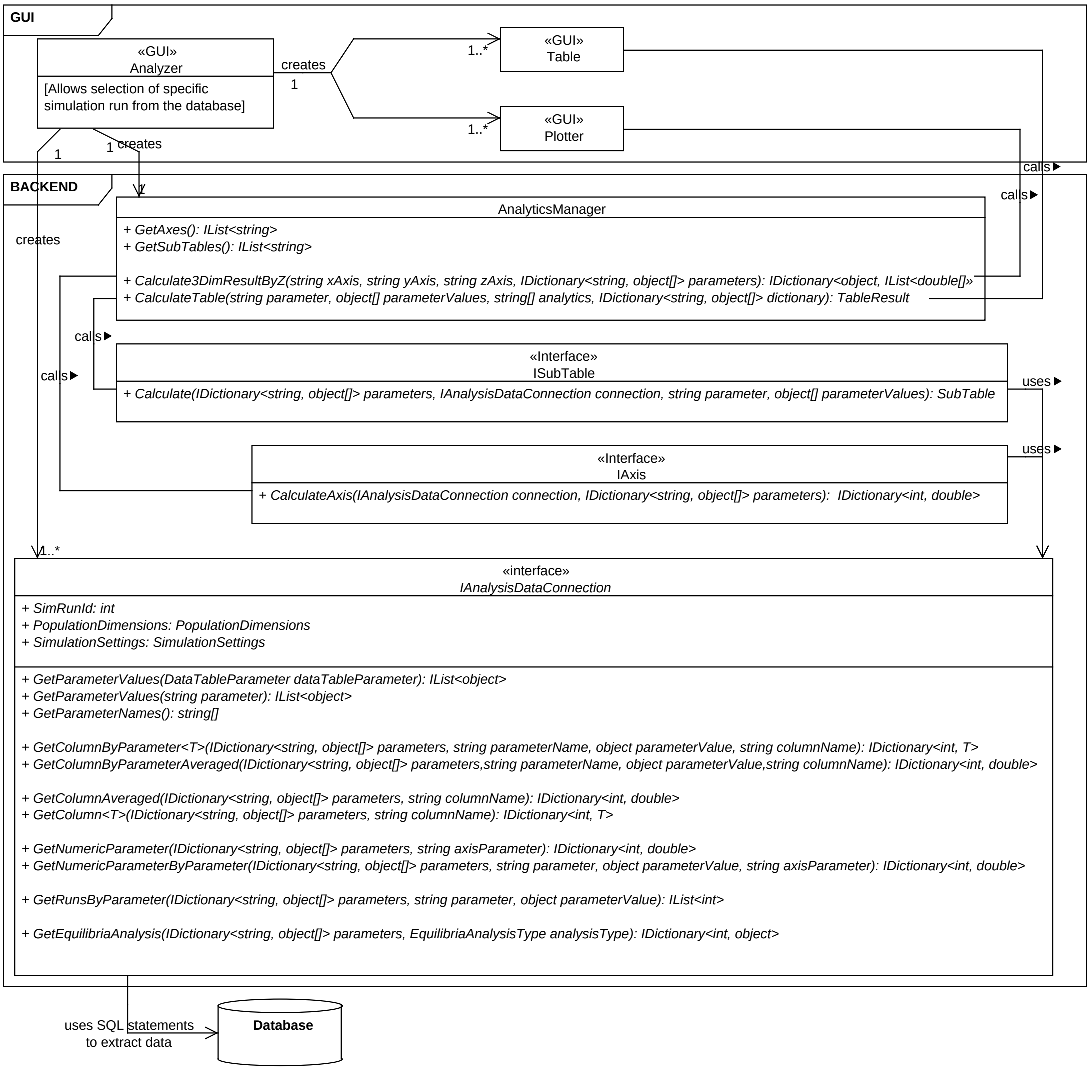

GUI
«GUI»
Analyzer
[Allows selection of specific
simulation run from the database]
creates
1
1..*
«GUI»
Table
1..*
«GUI»
Plotter
1
1
creates
calls ▶
BACKEND
1
calls ▶
creates
AnalyticsManager
+ GetAxes(): IList<string>
+ GetSubTables(): IList<string>
+ Calculate3DimResultByZ(string xAxis, string yAxis, string zAxis, IDictionary<string, object[]> parameters): IDictionary<object, IList<double[]»
+ CalculateTable(string parameter, object[] parameterValues, string[] analytics, IDictionary<string, object[]> dictionary): TableResult
calls ▶
calls ▶
«Interface»
ISubTable
+ Calculate(IDictionary<string, object[]> parameters, IAnalysisDataConnection connection, string parameter, object[] parameterValues): SubTable
uses ▶
«Interface»
IAxis
+ CalculateAxis(IAnalysisDataConnection connection, IDictionary<string, object[]> parameters): IDictionary<int, double>
uses ▶
1..*
«interface»
IAnalysisDataConnection
+ SimRunId: int
+ PopulationDimensions: PopulationDimensions
+ SimulationSettings: SimulationSettings
+ GetParameterValues(DataTableParameter dataTableParameter): IList<object>
+ GetParameterValues(string parameter): IList<object>
+ GetParameterNames(): string[]
+ GetColumnByParameter<T>(IDictionary<string, object[]> parameters, string parameterName, object parameterValue, string columnName): IDictionary<int, T>
+ GetColumnByParameterAveraged(IDictionary<string, object[]> parameters,string parameterName, object parameterValue,string columnName): IDictionary<int, double>
+ GetColumnAveraged(IDictionary<string, object[]> parameters, string columnName): IDictionary<int, double>
+ GetColumn<T>(IDictionary<string, object[]> parameters, string columnName): IDictionary<int, T>
+ GetNumericParameter(IDictionary<string, object[]> parameters, string axisParameter): IDictionary<int, double>
+ GetNumericParameterByParameter(IDictionary<string, object[]> parameters, string parameter, object parameterValue, string axisParameter): IDictionary<int, double>
+ GetRunsByParameter(IDictionary<string, object[]> parameters, string parameter, object parameterValue): IList<int>
+ GetEquilibriaAnalysis(IDictionary<string, object[]> parameters, EquilibriaAnalysisType analysisType): IDictionary<int, object>
uses SQL statements
to extract data
Database

Figure 3.4: The `IAnalysisDataConnection` interface provides access to the data stored in the database. The Analyzer provides two graphical user interfaces to display the data, a table-based depiction and a plotter, the two classes called `Table` and `Plotter`. Both inherit the `Windows.Form` base class. For a selected simulation run, an arbitrary number of these `Forms` can be created and used parallel to facilitate comparison of the data. A `Table` allows to create a table of averaged measures from the data stored in the database. Every measure available for selection implements the `ISubTable` interface. This allows to extend the implementation by implementing additional measures, if needed. Additionally, the results can be grouped by a specific `DataTableParameter`. The `Plotter` uses implementation of the `IAxis` class, which defines one axis of the plotter. The `CalculateAxis` method returns a dictionary of population run ids and the corresponding axis value. This way, the axes can be calculated parallelized and the results joined later on in the analysis. Access to the data of a completed simulation run stored in the database is handled via the `IAnalysisDataConnection` interface. This interface is used by the `IAxis` and `ISubTable` implementations. The implementation of the `IAnalysisDataConnection` then uses SQL statements to extract the data from the database. Many methods are generically and this way cover the different use of `IAxis` and `ISubTable` implementations.

defined by a function $\mathscr{C} : \mathsf{N}_0 \rightarrow \mathsf{N}_0^L, i \mapsto \mathbf{i}$, where $\mathbf{i} = (i_1, ..., i_L)$ denotes the multi-index and i the array-index of the gamete. We define the bijective function $\mathscr{C}$ by the inverse $\mathscr{C}^{-1}$

$$\mathscr{C}^{-1}(\mathbf{i}) = i_1 I^{L-1} + i_2 I^{L-2} + ... + i_L I^0 \tag{3.1}$$

Implementation of the functions is given in pseudo code, see Algorithm 2 and Algorithm 3. Note that for uniqueness of the decomposition, we have to begin numeration of the gametes as well as of the alleles at zero. The first possible allele is thus always denoted by zero, as usual for indices within arrays, e.g., the gamete consisting of the first allele on every locus, is given by $(0, ..., 0)$.

Algorithm 2 CalculateAllelicComposition

Description: Calculates the allelic composition of the gamete index

Input: i (int): The index of the gamete.
L (int): The number of loci
I (int): The number of alleles

Output: (int[L]): The allelic composition

```
int $total = i$;
int[] $alleleComp$ = new int[$L$];
for int $index = L - 1$ to 0 step $-1$ do
    double $basis = I^{index}$;
    int $k = \lfloor \frac{total}{basis} \rfloor$;
    $total = total \% basis$;
    $alleleComp[i] = k$;
end for
return $alleleComp$;
```

Algorithm 3 CalculateGameteIndex

Description: Calculates the gamete index from the allelic composition

Input: *alleleComp* (int[L]): The allelic composition
L (int): The number of loci
I (int): The number of alleles

Output: int: The index of the gamete

```
int $i = 0$;
int $j = 0$;
for int $index = L - 1$ to 0 step $-1$ do
    $i+ = alleleComp[j] I^{index}$;
    $j + +$;
end for
return $i$;
```

3.3.2 Calculation of Recombination Probabilities

We implemented a special case of recombination, as stated in Section 1.2.2. We used equation (1.25) to design an algorithm that, given $\rho = (\rho_1, ..., \rho_{L-1})$, calculates the probability $R_{i,jl}$ that the gamete i is formed by recombination of the parental gametes j and l.

The applied algorithm, see Algorithm 4, proceeds from one locus to the next. The current locus is represented by the *index* parameter. If either of both provided parents exhibits the same allele as the offspring on the locus, the algorithm is called recursively with adapted joint probability p for the next locus. If neither of the parental gametes has the same allele on the current locus, 0 is returned. To enable the algorithm to deal with the starting $index = 0$ correctly, we redefine the recombination rates to $\hat{\rho} = (0.5, \rho_1, ..., \rho_{L-1})$, a vector of length L, before calling this method.

In each iteration step, i.e., for each generation, the provided `ISelectionPattern` of a population calculates the right hand side of (1.9b). To reduce the computational costs, we only summed over those indices, exhibiting strictly positive recombination rates. Therefore, the recombination pattern does not only provide the recombination probabilities $R(i, jl)$, but also the indices (i, jl) exhibiting a recombination probability not equal to zero, i.e., $R(i, jl) > 0$. Thus, (1.9b) is implemented as

$$p_{i,\alpha}^{\#} = \frac{1}{\bar{w}_\alpha} \sum_{\phi \in \Phi} P(\phi) p_{\phi_1,\alpha} p_{\phi_2,\alpha} w_{\phi_1 \phi_2,\alpha}, \tag{3.2}$$

where, the recombination probabilities $R(i, jl)$ are stored in a three dimensional array P. The positive recombination indices Φ are stored as a list of vectors. Every $\phi \in \Phi$ is a vector of positive integers $\phi = (\phi_0, \phi_1, \phi_2)$, where $\phi_i \in \mathbb{N}^+$ for $i = 0, 1, 2$.

Example: Let the number of loci be $L = 3$, the number of alleles $I = 2$. The recombination probabilities between the loci are given by $\rho = (0.2, 0.3)$. Let's calculate the recombination probability for the two gametes $G_1 = A^{(1)}B^{(2)}B^{(3)}$ and $G_2 = A^{(1)}A^{(2)}A^{(3)}$ to have an offspring $G_3 = A^{(1)}B^{(2)}A^{(3)}$. Obviously, there are two possible recombination events, see Figure 3.5.

Therefore, we use Algorithm 4. $p = 1, L = 3$, $index = 0$, $child = (0, 1, 0)$, $currentParent = (0, 1, 1)$, $otherParent = (0, 0, 0), \hat{\rho} = (0.5, 0.2, 0.3)$.

Since both parents have the offspring allele A on the first locus, both cases occur (line 6 and 10). Let's take a look at the first case (recombination 1 in Figure 3.5). We will come back to line 10 later. Here, we call the algorithm recursively with $p = 0.5(1 - 0.5) = 0.25, index = 1$, the other parameters stay the same. This reflects no recombination. Still, the $currentParent = G_1$ is correct, exhibiting allele B at the second locus. Again, no recombination is necessary, see line 7; the algorithm is called again for $p = 0.25(1 - 0.2) = 0.2, index = 2$. This time a recombination is necessary to result in allele A on the third locus, thus line 10 results in true, the algorithm is called for $p = 0.2 \cdot 0.3 = 0.06, index = 3$. This time, line 1 returns true since the we reached the last locus. The algorithm returns $p = 0.06$. Recall, we

Algorithm 4 CalculateRecombinationProbability

Description: Calculates the recombination probability for the parental gametes j and l to have an offspring gamete i.

Input: p (double): The joint probability
L (int): The number of loci.
index (int): The current locus.
child (int[L]): The allelic composition of the offspring gamete.
currentParent (int[L]): The allelic composition of the first parent gamete.
otherParent (int[L]): The allelic composition of the second parent gamete.
$\hat{\rho}$ (int[L]): The recombination rates.

Output: double: The probability for a recombination resulting in the offspring.

```
if $index = L$ then //On the last index, return the current probability
    return $p$;
end if
bool $caseHit = false$;
double $sum = 0$;
if $currentParent[index] = child[index]$ then //If current parent is correct, proceed without recombination via recursive call
    $sum+ = CalculateRecombinationProbability(p(1 - \hat{\rho}[index]), L, index + 1, child, currentParent, otherParent)$;
    $caseHit = true$;
end if
if $otherParent[index] = child[index]$ then //If other parent is correct, proceed with recombination via recursive call
    $sum+ = CalculateRecombinationProbability(p\hat{\rho}[index], L, index + 1, child, otherParent, currentParent)$;
    $caseHit = true$;
end if
if $caseHit$ then //If no case has been hit, recombination is not possible, return 0
    return $sum$;
else
    return $0$;
end if
```

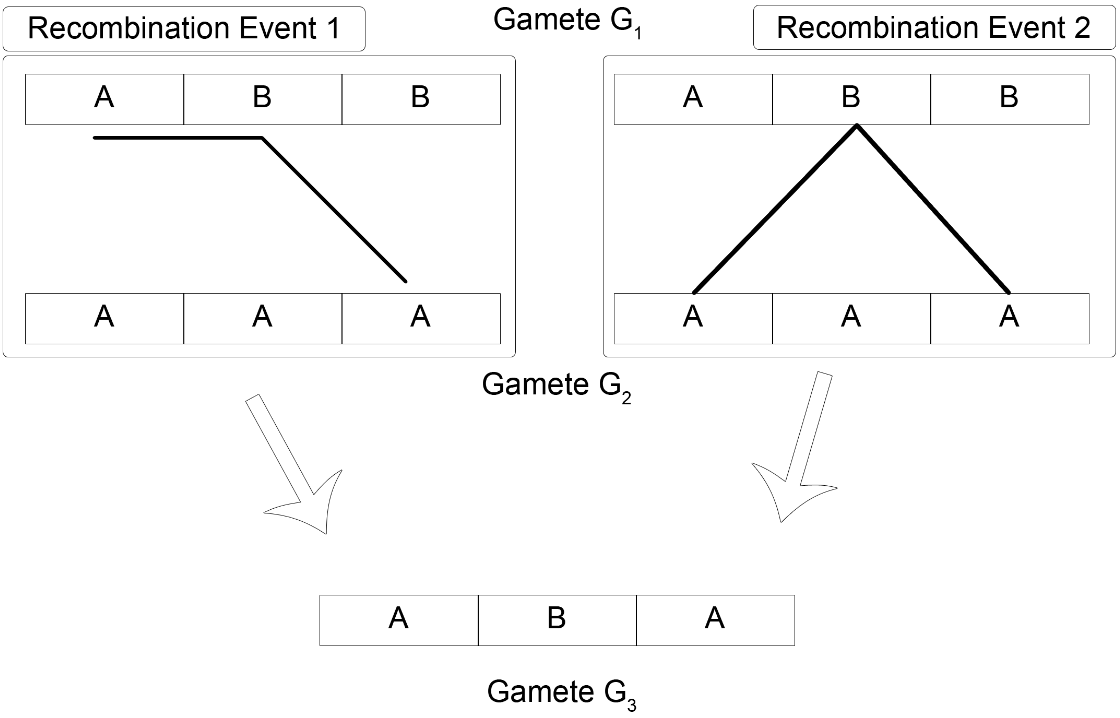


Figure 3.5: Example for two possible recombination events resulting from different combinations of crossover events in the case of 3 loci.

still have another case untracked, the call on line 6 from the beginning returns 0.06, which is added to the variable *sum*. Then, line 10 returns true, resulting in another algorithm call (recombination 2 in Figure 3.5). This reflects the possibility of starting with the second parent as *currentParent* (This is the reason why we replaced ρ, the recombination probabilities between loci, by $\hat{\rho}$). This time the recursive calls result in $p = (0.5 \cdot 0.5)(0.2)(0.3) = 0.015$, reflecting the discussed change of parent and two recombination events. This is also added to the variable $sum = 0.06 + 0.015 = 0.075$, which is returned as the probability by the algorithm on line 15.

Internally, the allelic compositions of the gametes are stored by arrays of integers, i.e., $G_1 = [0, 1, 1]$, $G_2 = [0, 0, 0]$, $G_3 = [0, 1, 0]$. Since the recombination probability for this case is not zero, a recombination index would be created. Therefore, we also need to calculate the gamete indices for our example. Therefore, we use Algorithm 3. For the parameters $alleleComp = G_1 = [0, 1, 1], L = 3, I = 2$, the algorithm returns the gamete index$g_1 = (0 \cdot 2^2) + (1 \cdot 2^1) + (1 \cdot 2^0) = 3$. Analogously, we calculate the gamete index $g_2 = 0$ and $g_3 = 2$ for gametes G_2 and G_3, respectively. This results in the recombination index $\phi_1 = [2, 3, 0]$ and of course $\phi_2 = [2, 0, 3]$. The recombination probability array Φ is adapted to $\Phi[\phi_1] = \Phi[\phi_2] = \Phi[2, 3, 0] = \Phi[2, 0, 3] = 0.075$.

3.3.3 Fitness Values Construction

We calculated the fitness values by applying a fitness function W_α in each deme α to the genotypic values G_{ij}, see Section 1.2.1. A genotypic value is the sum of the allelic contributions. To construct the allelic contributions, we used the same approach as stated by Bürger and Gimelfarb (1999).

We allow to set the sum of allelic contributions for a single allele, and, create randomized sets of allelic contributions summing-up correctly. This means, we define $\bar{\gamma}_{\mathbf{n}} = \sum_k \gamma^{(k)}_{\mathbf{n}_k}$, where $\mathbf{n} = (\mathbf{n}_1 = n, ..., \mathbf{n}_L = n)$ denotes the gamete exhibiting allele n on every locus.

The implementations of the implemented fitness generators work exactly reverse, accepting a set $\{\bar{\gamma}_{\mathbf{n}} : n \in \mathsf{I}\}$ for all alleles as input and generate positive random numbers $\{\gamma_{\mathbf{n}_k} : k \in \mathsf{L}\}$ for each allele n summing up correctly.

Moreover, we allowed to define the minimal distance ϵ_A in standard deviation of allelic effects between the sets. This allows us to force construction of nearly uniform distribution in standard deviation of allelic effects, see Table 3.2 and (1.38).

3.3.4 Initial Value Construction

Our implementation to construct the initial value sets ensures the initial gamete frequencies to have a minimum Euclidean distance in at least one deme. This means that for two initial value sets $\mathsf{P} = \{p_{i,\alpha} : i \in \mathsf{N}, \alpha \in \mathsf{G}\}$ and $\mathsf{Q} = \{q_{i,\alpha} : i \in \mathsf{N}, \alpha \in \mathsf{G}\}$,

$$\sqrt{\sum_i (p_{i,\alpha} - q_{i,\alpha})^2} > \epsilon_E \tag{3.3}$$

for at least one α, where ϵ_E defines the minimal Euclidean distance.

4 Quadratic Stabilizing Selection

Previous results on quadratic stabilizing selection, cf. Section 2.2, on the one hand show that high genetic variability can be preserved in a two locus system, given enough disparity of the allelic effects. On the other hand, numerical results suggest that the genetic variance and the probability for polymorphic loci declines rapidly for increasing number of loci. How does population subdivision and migration influence this behavior, and can it account for a higher genetic variability? Therefore, we model a quantitative trait under stabilizing selection in multiple demes. To model different selection pressures within the demes, we define the quadratic fitness functions to exhibit different positions of the maximum. Thereby we want to address the following questions:

- Genetic variability: Can a higher amount of genetic variability be maintained by introducing migration?

- Polymorphism: Does migration allow for higher probability of polymorphic loci and full polymorphisms?

- Equilibria: How many stable equilibria can coexist in such a system? Depending on migration, how does the equilibrium structure change?

- Strong migration: How strong must migration be relative to selection, i.e., how weak must selection be, to create results which can be understood as pertubation of the weak selection limit? How does the relationship of selection and migration influence the behavior of the dynamical system?

- Weak migration: How weak must migration be relative to selection and recombination, so that the resulting equilibria can be considered as a perturbation of the equilibria of the system without migration?

We will start with a single simulation run, as defined in Section 3.1, for the case of two alleles and two loci in a panmictic population, in order to have a point of comparison for our further findings. Starting with the symmetric model, we have a full analysis at our disposal, also serving as a test case for our simulation. Afterwards, we will simulate the panmictic model for arbitrary position of the optimum in order to quantify the dependence on the deviation of the optimum from the double heterozygote. Following this preliminary results, we continue with a simulation of a Deakin migration model with two diallelic demes exhibiting symmetrically displaced optima within the demes.

Alleles / Loci / Demes	2 / 2 / 1
Max Iterations	300000
Error / Stagnation Count	10^{-10} / 10
Allele Loss	10^{-4}
Initial Values	20 / $\epsilon_E = 0.25$
Average Allelic Contributions	$(\bar{\gamma}_\mathbf{1}, \bar{\gamma}_\mathbf{2}) = (0, 0.5)$ / 40 sets, $\epsilon_A = 0.005$
Optimum Positions	$P_O \in \{0.5, 0.51, 0.52, 0.53, 0.54, 0.55, 0.6, 0.7, 0.8, 1\}$
Normalized Selection Coefficients	$\mathscr{U}(0.25, 1, 10)$
Recombination Rates	$\mathscr{U}(0, 0.5, 10)$

Table 4.1: Parameters and settings for the simulation of a diallelic two-locus panmictic population under quadratic stabilizing selection.

Analysis of the executed simulation runs is based on the discussion of the equilibrium structure and quantitative measures of variation and local adaption at equilibrium. The quantities used throughout this chapter were calculated as described in Section 1.3 and Table 3.3.

4.1 A Single Deme

Here we present our results for a single deme under quadratic stabilizing selection. We simulated the dynamics for 40000 constructed parameter sets and iterated each for 20 initial values. Table 4.1 provides an overview on the settings for this simulation run. $\mathscr{U}(a, b, k)$ refers to k values for a parameter, randomly chosen from a uniform distribution with a support of $[a, b]$.

We use normalized selection coefficients $\tilde{s}$, where a value of $\tilde{s} = 1$ refers to the strongest possible selection, as defined in (1.18). This is necessary to make the strength of selection comparable, since the strongest possible selection varies for different optimum positions, cf. (1.17). The selection coefficient was randomly chosen between 0.25 and 1, since we already know that the behavior changes only on this range in the symmetric case, cf. Figure 2.1.

The recombination rate was randomly chosen from tight linkage $\rho = 0$ up to no linkage $\rho = 0.5$. We also chose ten different values for the position of the fitness function maximum P_O, increasing the step size, as the values reach 1.

The allelic contributions $\bar{\gamma} = (\bar{\gamma}_\mathbf{1}, \bar{\gamma}_\mathbf{2}) = (0, 0.5)$ result in a genetic value of $G = 0.5$ for the double heterozygote. For all 40 sets of randomly chosen allelic effects, we ensured a distance of $\epsilon_A = 0.005$ between the standard deviations of two sets. Details on the construction of allelic effects were given in Section 3.3.3.

Symmetric Case First, we discuss the symmetric case, where the optimum of the quadratic fitness function is assumed by the double heterozygote, given our normalization of the selection intensity $s = 4\tilde{s}$, i.e.,

$$W_1(G) = \Psi(G) = 1 - s(G - 0.5)^2. \tag{4.1}$$

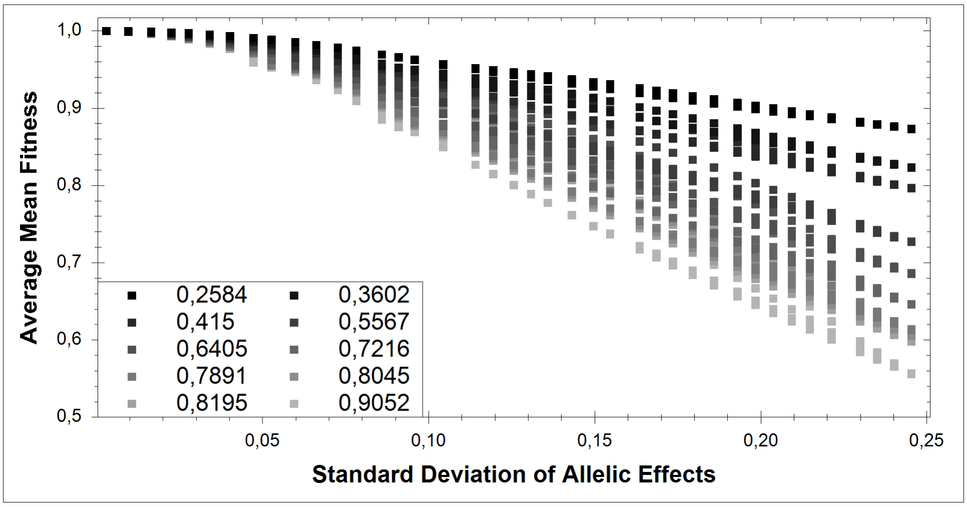


Figure 4.1: Average mean fitness as a function of the standard deviation of allelic effects for the symmetric case $P_O = 0.5$. Gray levels refer to different randomly chosen selection coefficients. Black refers to the lowest selection strength, the lightest gray to the highest. Every data point is calculated as the average mean fitness $\bar{\omega}$ for one parameter combination $(r, \tilde{s}, \bar{\gamma})$ averaged over 20 initial values. Mean fitness decreases for higher disparity in allelic effects. For stronger selection, the decline in average mean fitness is faster.

As the probability to reach a polymorphic equilibrium becomes higher with increasing disparity in allelic effects, the genetic variance increases, see Figure 4.3. This is consistent with the results achieved by Bürger and Gimelfarb (1999). Mean fitness decreases, see Figure 4.1. For higher selection pressure, the decrease of mean fitness is even stronger. For a fully polymorphic equilibrium, the mean fitness is lower, because, due to recombination, gametes of lower marginal fitness are present. For stronger selection, existence and stability of a fully polymorphic equilibrium is more frequent, and, thus, the mean fitness decreases even faster.

Asymmetric Optimum By shifting the optimum of the quadratic function towards 1, the genetic variance, the mean fitness, and the probability for loci to be polymorphic change drastically. For the same parametrization as in the symmetric case, cf. Table 4.1, we allowed ten different positions of the optimum. Table 4.2 shows the numerical results. As the position of the optimum increases towards 1, the polymorphic fraction decreases. Already for $P_O = 0.7$, no more than 2 gametes are found to be present at the same time, resulting in a maximum of one polymorphic locus. In the extreme case of an optimum at $P_O = 1$, which leads to directional selection, genetic variation is depleted completely, but mean fitness is maximized.

Based on these numerical results, we conclude, that only a slight shift of the optimum seems reasonable to model stabilizing selection in two demes. Otherwise, the behavior of the dynamical system due to the fitness landscape would not be an accurate model for stabilizing selection, which is assumed to act stabilizing towards an intermediate optimum, cf. Endler (1986). The observation by Gavrilets and Hastings

p_O	Polymorphic Loci			Polymorphic	Mean	Linkage	Genetic
	0	1	2	Fraction	Fitness	Disequilibrium	Variance
0.5	0.3183	0.5128	0.1689	0.4253	0.8631	0.0184	0.0468
0.51	0.319	0.546	0.135	0.4081	0.8676	0.0152	0.0467
0.52	0.339	0.5466	0.1143	0.3876	0.8743	0.0128	0.0455
0.53	0.3336	0.5627	0.1036	0.385	0.8812	0.0108	0.0438
0.54	0.3466	0.5583	0.0951	0.3743	0.8882	0.0089	0.0418
0.55	0.3638	0.5521	0.084	0.3601	0.8951	0.0072	0.0394
0.6	0.4855	0.4787	0.0357	0.2751	0.9269	0.002	0.0265
0.7	0.6562	0.3438	-	0.1719	0.9689	0	0.0078
0.8	0.55	0.45	-	0.225	0.9864	0	0.0077
1	1	-	-	-	1	0	0

Table 4.2: Average ratio of polymorphisms and average polymorphic fraction of the genome, average mean fitness, average linkage disequilibrium, and average genetic variance for ten different positions of the optimum. All averages taken over 4000 parameter sets $(r, \tilde{s}, \bar{\gamma})$ and 20 initial values.

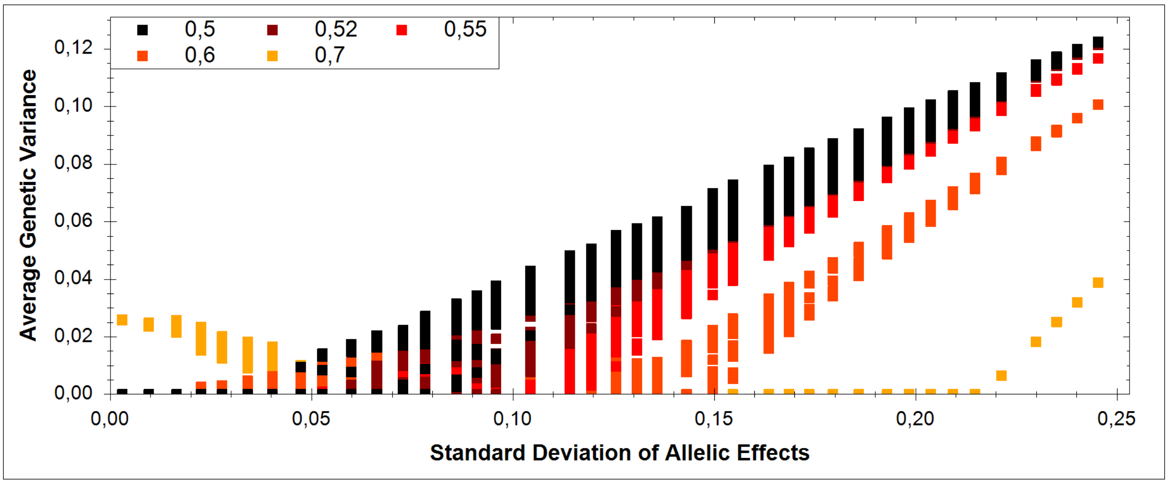


Figure 4.2: Average genetic variance as a function of the standard deviation of allelic effects for different optimum positions. Every data point reflects the genetic variance $\sigma^2_{G,\alpha}$ for a given parameter set $(r, \tilde{s}, \bar{\gamma})$ averaged over 20 initial values. Color-coded by the optimum position. Average genetic variance decreases for higher deviations of the optimum from the mean, given the standard deviation of allelic effects is high enough.

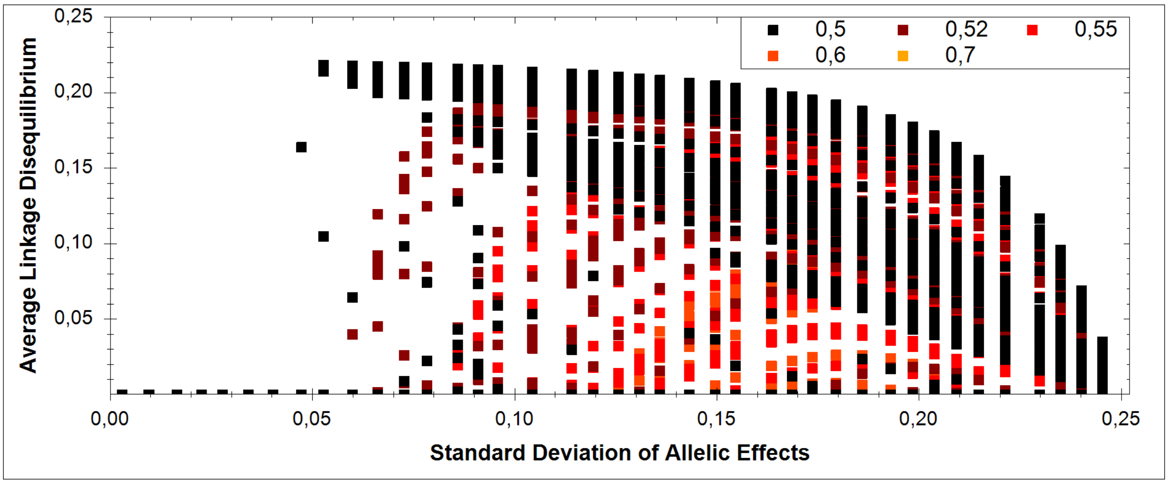


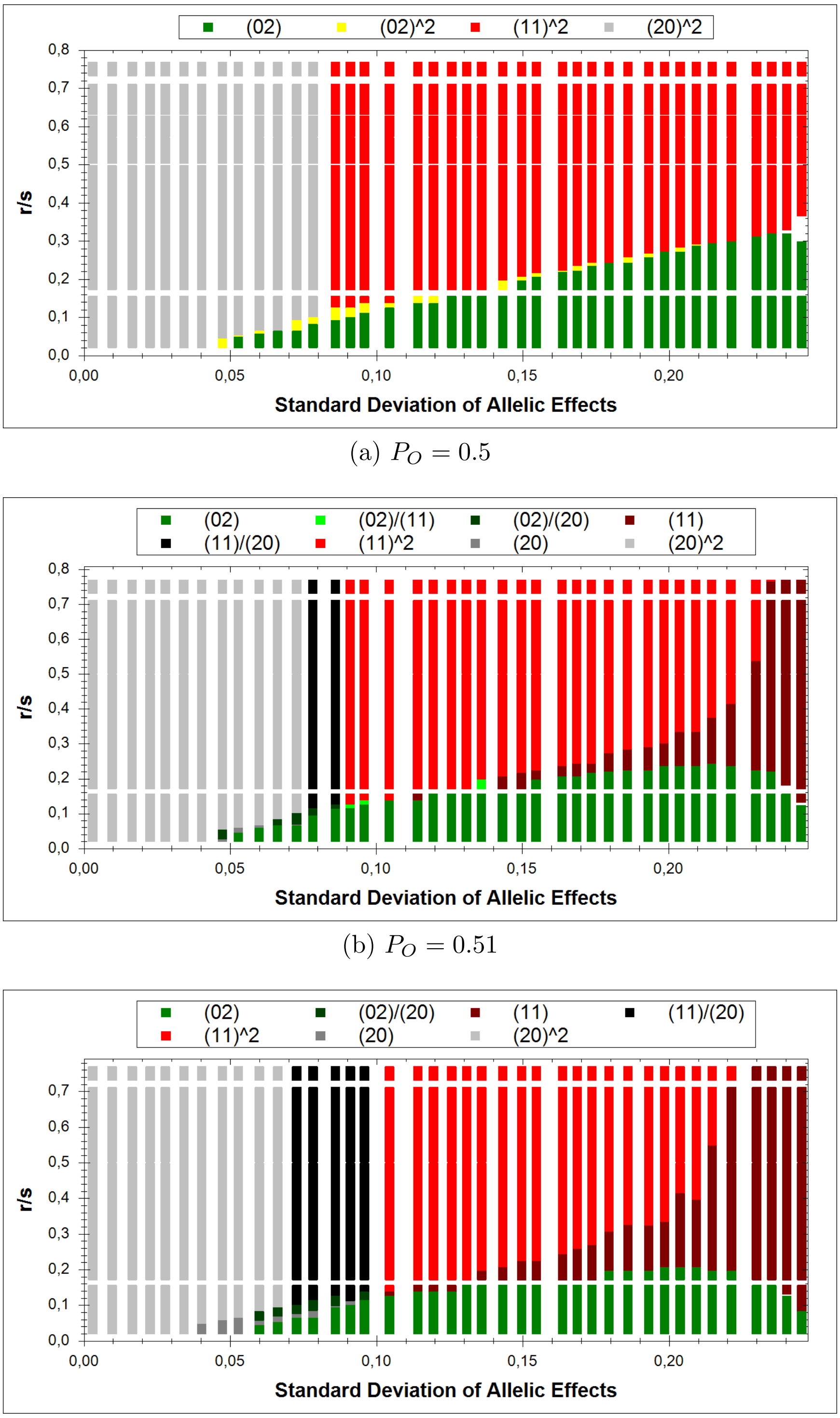
Figure 4.3: Average linkage disequilibrium as a function of the standard deviation of allelic effects for different optimum positions. Every data point reflects linkage disequilibrium for a given parameter set $(r, \tilde{s}, \bar{\gamma})$ averaged over 20 initial values. Color-coded by the optimum position. Linkage disequilibrium was found to be only significant for slightly displaced optimum, cf. Table 4.2.

(1993) of declining genetic variance as the optimum deviates further suggests small deviations of the optimum from symmetry to maintain high genetic variance.

Stable Equilibria As we have seen in the symmetric case, the existence and stability of polymorphic equilibria depends on the relation of selection to recombination as well as the allelic effects, cf. Figure 2.1. If the optimum deviates from the double heterozygote, such a general analysis of the equilibria is not available. Nevertheless, our simulation approach allows us to study the existence of stable equilibria as in the symmetric case. From now on, if not stated otherwise, we will only refer to stable equilibria, since unstable equilibria are not included in our numerical analysis.

In order to compare our result to the qualitative behavior of the symmetric case as stated in Section 2.2.1, note that the analysis for the symmetric case in Figure 2.1 depicts an ordinate of $r/s \in [0, 0.125]$, since this covers the area, where the behavior depends on this ratio. Since we rescaled selection, $\tilde{s} = 4s$, this translates to the interval $r/\tilde{s} \in [0, 0.5]$.

Figure 4.4 shows the numerical results of the equilibrium analysis for the cases $P_O = 0.5$, 0.51, 0.52, 0.55, 0.6, and 0.7. We also included the symmetric case, on the one hand to serve as a verification of our methods and implementation, on the other hand to make it easier for the reader to compare the result as displayed here. Clearly, the numerical results for the symmetric case coincide with the analytical ones stated in Section 2.2.1.

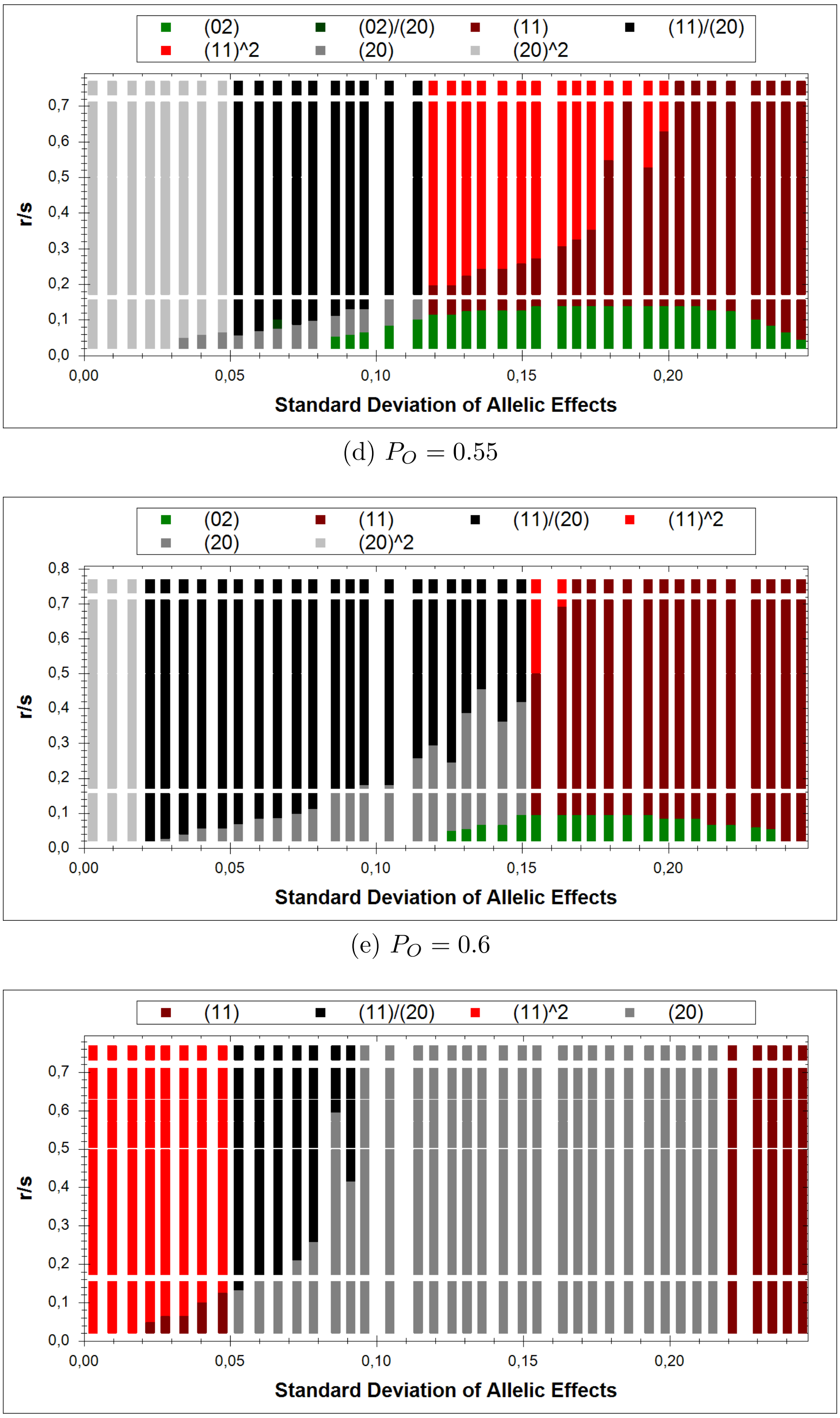


(a) $P_O = 0.5$

(b) $P_O = 0.51$

(c) $P_O = 0.52$

Figure 4.4: Numerically determined equilibrium structure.



(d) $P_O = 0.55$

(e) $P_O = 0.6$

(f) $P_O = 0.7$

Figure 4.4: Numerically determined equilibrium structure as a function of the standard deviation of allelic effects and $r/\tilde{s}$ for different optimum positions. The colors refer to different equilibrium structures. An expression of the form $(a, b)\hat{}n$ refers to an equilibrium type and its frequency. Here, a is the number of monomorphic loci, b to the number of polymorphic loci, n denotes the number of equilibria of this type found for the different simulated initial values. If only one such equilibrium was found, $\hat{}1$ is omitted. If more than one type of equilibria has been found, different types are separated by a slash '/'.

For the smallest simulated deviation of the optimum from symmetry, i.e., $P_O = 0.51$, as in the symmetric case, two monomorphic equilibria exist (gray), if allelic effects exhibit only small deviations and recombination is strong enough relative to selection. In this case of relatively strong recombination, for higher disparity in allelic effects, $\sigma_{\bar{\kappa}} \in [0.075, 0.9]$, two equilibria exist (black). One exhibiting one polymorphic locus, the other one being fully polymorphic. A single full polymorphic equilibrium (green) was only found for a smaller parameter area compared to the symmetric model. For high disparity in allelic effects, $\sigma_{\bar{\kappa}} > 0.75$, at least one locus always was found to be polymorphic (red, dark red, black, green and light green). At the boundary of the area exhibiting one polymorphic equilibrium (green), four different equilibrium structures that do not occur in the case of no migration can be found: Between the regions of a full polymorphism and two monomorphisms, a structure exhibiting one full polymorphism and one monomorphism (dark green), as well as a single monomorphic equilibrium was found (dark gray). The area for a single monomorphic equilibrium will expand further as the position of the optimum increases towards 1. It represents the area of a single homozygote matching the optimum closely and finally becoming fixed. Between the area exhibiting a full polymorphism and the area of two single locus polymorphisms, an equilibrium structure with a full polymorphism and a one locus polymorphism was found. Moreover, an area with only one equilibrium with one locus polymorphic (dark red) can be located. This area has been found to increase further for higher optimum deviation up to 0.6. Already for the case $P_O = 0.6$, the area exhibiting two coexisting monomorphic stable equilibria is reduced significantly (gray). For the case of $P_O = 0.7$ this area can no longer be located in our numerical result and no full polymorphism was found (green).

As the position of the optimum increases towards one, the area reflecting a single monomorphic equilibrium expands. For complete directional selection $P_O = 1$ (graph not shown), this was the only equilibrium structure in the searched parameter space. For all positions of the optimum, no more than two stable coexisting equilibria were found, as was proven to be true for the symmetric case in (Bürger and Gimelfarb, 1999) and (Bürger, 2000, p.206).

4.2 Two Demes - Symmetrical Optima

To introduce migration, we simulated a population subdivided into two demes. Fitness values were calculated using quadratic fitness functions with symmetrical optimum positions between the demes, i.e.,

$$W_1(G) = 1 - s(G - (0.5 - d))^2, \text{ and} \tag{4.2a}$$

$$W_2(G) = 1 - s(G - (0.5 + d))^2, \tag{4.2b}$$

where $d \in [0, 0.5]$ defines the magnitude of displacement, which for a first simulation run was set to be $d = 0.05$, based on the results gathered in Section 4.1. Furthermore, using the Deakin migration scheme, we allowed for ten different migration rates ranging

Alleles / Loci / Demes	2 / 2 / 2
Max Iterations	$300,000$
Error / Stagnation Count	10^{-10} / 10
Allele Loss	10^{-4}
Initial Values	40 / $\epsilon_E = 0.25$
Average Allelic Contributions	$(\bar{\gamma}_\mathbf{1}, \bar{\gamma}_\mathbf{2}) = (0, 0.5)$ / 40 sets
Quadratic Optimum Positions	0.45/0.55
Normalized Selection Coefficients	$\mathscr{U}(0.25, 1, 10)$
Recombination Rates	$\mathscr{U}(0, 0.5, 10)$
Migration Rates	$\mu \in \{0, 0.001, 0.01, 0.03, 0.06, 0, 1, 0.15, 0.2, 0.5, 1\}$
Population Regulation	Soft selection

Table 4.3: Parameters and settings for the simulation of a two deme population with two diallelic loci, subject to quadratic stabilizing selection with symmetrical optimum positions and Deakin migration.

from weak ($\mu \leq 0.1$) to strong ($\mu \geq 0.2$) migration, including the special case of the Levene model ($\mu = 1$). Soft selection was assumed and the selection coefficient always exceeded 0.25 as discussed in Section 4.1. Settings are outlined in Table 4.3.

Again, we start by discussing quantitative measures of variation and local adaptation at equilibrium, followed by an analysis of the stable equilibria.

Measures of local adaption and variation Table 4.4 shows the observed quantities for different migration rates. Since in the absence of migration, the two demes contain two independent populations, which independently reach a equilibrium, polymorphic loci may also result from the combination of two different monomorphic equilibria within the demes. Thus, the polymorphic fraction of the genome decreases as migration becomes stronger. The ratios of single-locus polymorphisms and monomorphisms increase, while two-locus polymorphisms become less frequent. The decrease in mean fitness can be explained by the fact that migration allows for presence of gametes within a deme in which they may not exceed the other gametes in fitness, but by immigration still exist at equilibrium.

As migration becomes stronger relative to selection, our results suggest that the population approaches spatial quasi homogeneity. Table 4.4 supports this interpretation, as it reveals the average variance of gametes among subpopulations to decrease drastically for increasing migration rates, and already is less than 10^{-4} for a migration rate of 0.5. As an analysis of the equilibrium structure will reveal, the Levene model shows a similar structure as the panmictic symmetric case. Comparing the size of genetic variance, linkage disequilibrium, polymorphic fraction of the genome, and mean fitness to the results for the panmictic symmetric case, also confirms this similarity, cf. Table 4.4 and Table 4.2. This is accounted for by the symmetric displacement of the optima. The strong migration limit, in this case the Levene model, results in the spatial homogeneous averaged gamete frequencies, as discussed in Section 2.3.1.

The highest linkage disequilibrium was found for a migration rate of $\mu = 0.01$. In

μ	Polymorphic Loci 0	1	2	Polymorphic Fraction	Linkage Disequilibrium	Genetic Variance
0	0.1014	0.1116	0.787	0.8428	0.0051	0.0557
0.001	0.107	0.1179	0.7751	0.834	0.0102	0.0538
0.01	0.1456	0.1769	0.6774	0.7659	0.0268	0.0538
0.03	0.2348	0.3737	0.3915	0.5784	0.0265	0.0506
0.06	0.2815	0.482	0.2364	0.4774	0.0186	0.0475
0.1	0.2949	0.4892	0.2159	0.4605	0.0175	0.0467
0.15	0.3193	0.4721	0.2086	0.4446	0.017	0.0461
0.2	0.3264	0.4685	0.2051	0.4394	0.0168	0.0456
0.5	0.3514	0.4471	0.2015	0.4251	0.0166	0.0446
1	0.3581	0.4399	0.202	0.422	0.0167	0.0441

μ	Q_{ST}	Variance of Gametes among Subpopulations	Mean Fitness	Gametic Variance of Fitness
0	0.4513	0.0743	0.8933	0.0001
0.001	0.3621	0.0649	0.8963	0.0001
0.01	0.1831	0.0327	0.8907	0.0006
0.03	0.0788	0.0078	0.8871	0.0006
0.06	0.044	0.0021	0.8878	0.0003
0.1	0.0297	0.0014	0.887	0.0004
0.15	0.02	0.0009	0.8863	0.0004
0.2	0.0141	0.0006	0.8857	0.0005
0.5	0.0023	0.0001	0.8834	0.0009
1	-	-	0.8814	0.0014

Table 4.4: Average fraction of polymorphisms, average linkage disequilibrium, average genetic variance, average Q_{ST}, average variance of gametes among subpopulations, average mean fitness, and average gametic variance of fitness for different migration rates and optimum displacement $d = 0.05$. Averages are always taken over 40 initial values and 4000 parameter combinations $(r, \tilde{s}, \bar{\gamma})$.

Figure 4.7a we plotted the average linkage disequilibriumas a function of standard deviation of allelic effects,

Stable Equilibria Numerical analysis of the equilibrium structure is shown in Figure 4.5. Note that we only depict equilibrium types according to their gametic composition in the meta population and not for each deme separately. Otherwise, distinction of the equilibria by demes would cause to many different possible combinations.

We first discuss the dynamics in the absence of migration Figure 4.5a, and then compare them to the results in the analogous panmictic case, where $P_O = 0.55$, cf. Figure 4.4a.

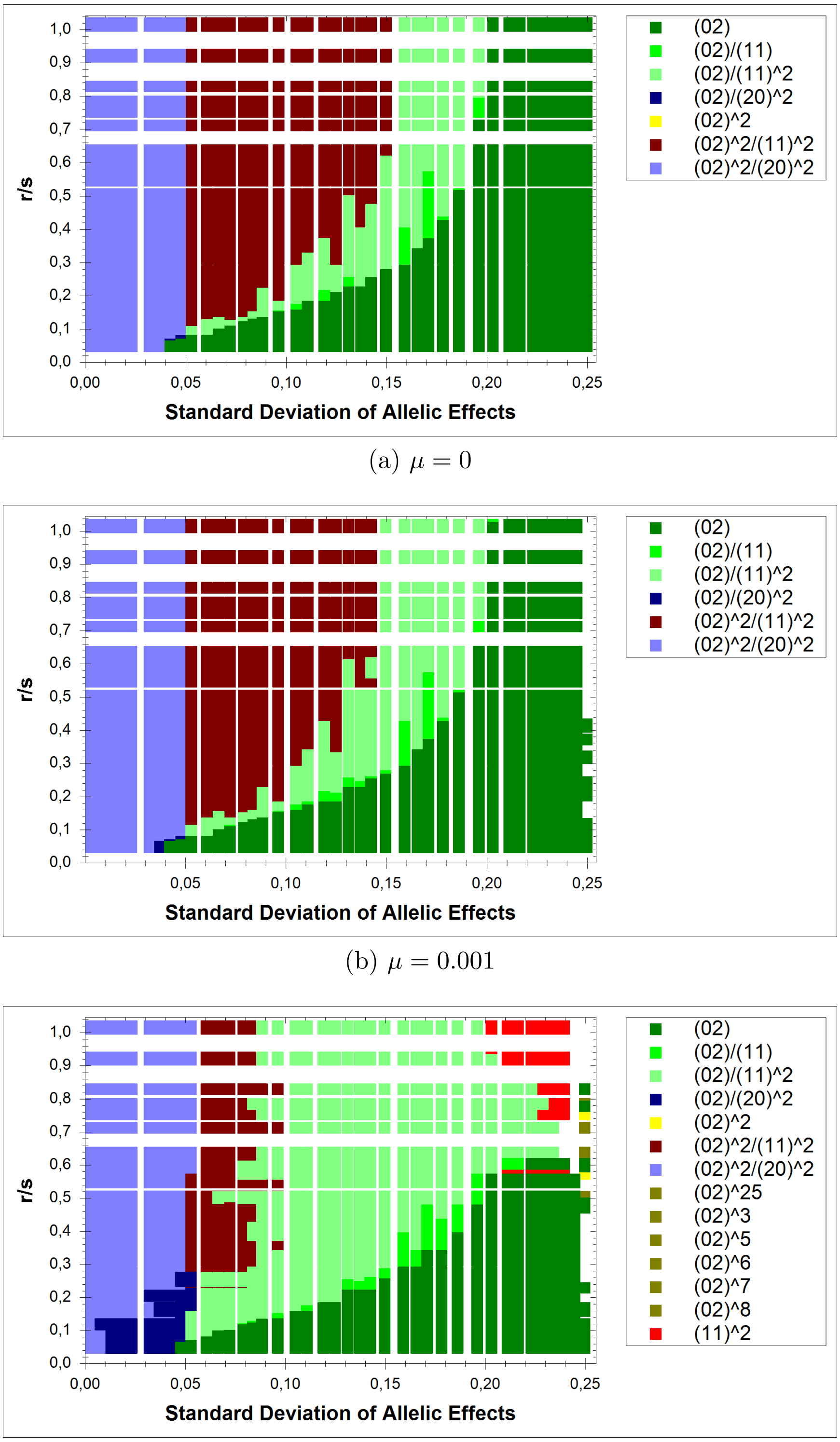
(Blue): If disparity of allelic effects is small enough $\sigma_{\bar{\kappa}} < 0.6$, by combination of the monomorphic equilibria within the demes, four equilibria are possible, two fully polymorphic and two monomorphic.

(Dark red / light green): Recall, that in the panmictic case one equilibrium with a single locus polymorphic as well as one monomorphic equilibrium exists or two equilibria with one locus polymorphic coexist. Combinations result in full polymorphisms or at least one locus polymorphic. As for small values of $\sigma_{\bar{\kappa}} < 0.15$, up to four stable equilibria can simultaneously coexist. Since the optima within the demes are displaced in opposite direction, different zygotes exhibit the genetic value closest to the position of the optimum. This is why the combination of two monomorphic single deme equilibria can result in an equilibrium with one locus polymorhic here.

(Green): For all chosen values of selection and recombination, for high enough disparity in allelic effects, only one fully polymorphic equilibrium exists.

For a migration rate of $\mu = 0.001$, cf. Figure 4.5b, no significant changes can be observed. This findings suggest that the weak-migration approximation applies if $\mu \leq 0.001$.

For an even higher migration rate, cf. Figure 4.5c and Figure 4.5d, the areas exhibiting full polymorphic equilibria resulting from the combination of monomorphic single deme equilibria decrease, until for strongest possible migration, Figure 4.5f, they cease to exist. As already discussed above, the Levene model results in deme independent gamete frequencies, and, thus the analysis of the stable equilibria is similar to the symmetric panmictic case, cf. Figure 4.5f and Figure 4.4a. We may assume that differences at the right edge of Figure 4.5f are due to numerical instabilities. Note that also in the cases for $\mu = 0.01, 0.01, 0.1$, for highest standard deviation in allelic effects, some numerical inaccuracies occured. This is why we plotted all equilibria of the form $(02)\hat{}n$, for $n \geq 3$ in the same color. These equilibria only occured for highest values of $\sigma_{\bar{\kappa}}$, exactly where most runs had to be excluded from the analysis since the iterations exceeded 300000. Figure 4.6a shows a heat map revealing that the highest average iterations (until equilibrium has been reached), were found for highest disparity in allelic effects. Few population runs exhibit the highest used value in standard deviation of allelic effects and reached a single fully polymorphic equilibrium, cf. Figure 4.5f. These show a very low average in iterations until equilibrium. Thus, we can assume, that the gamete frequencies of these populations in fact change so slowly causing the euclidean distance to stagnate according to the defined minimal change

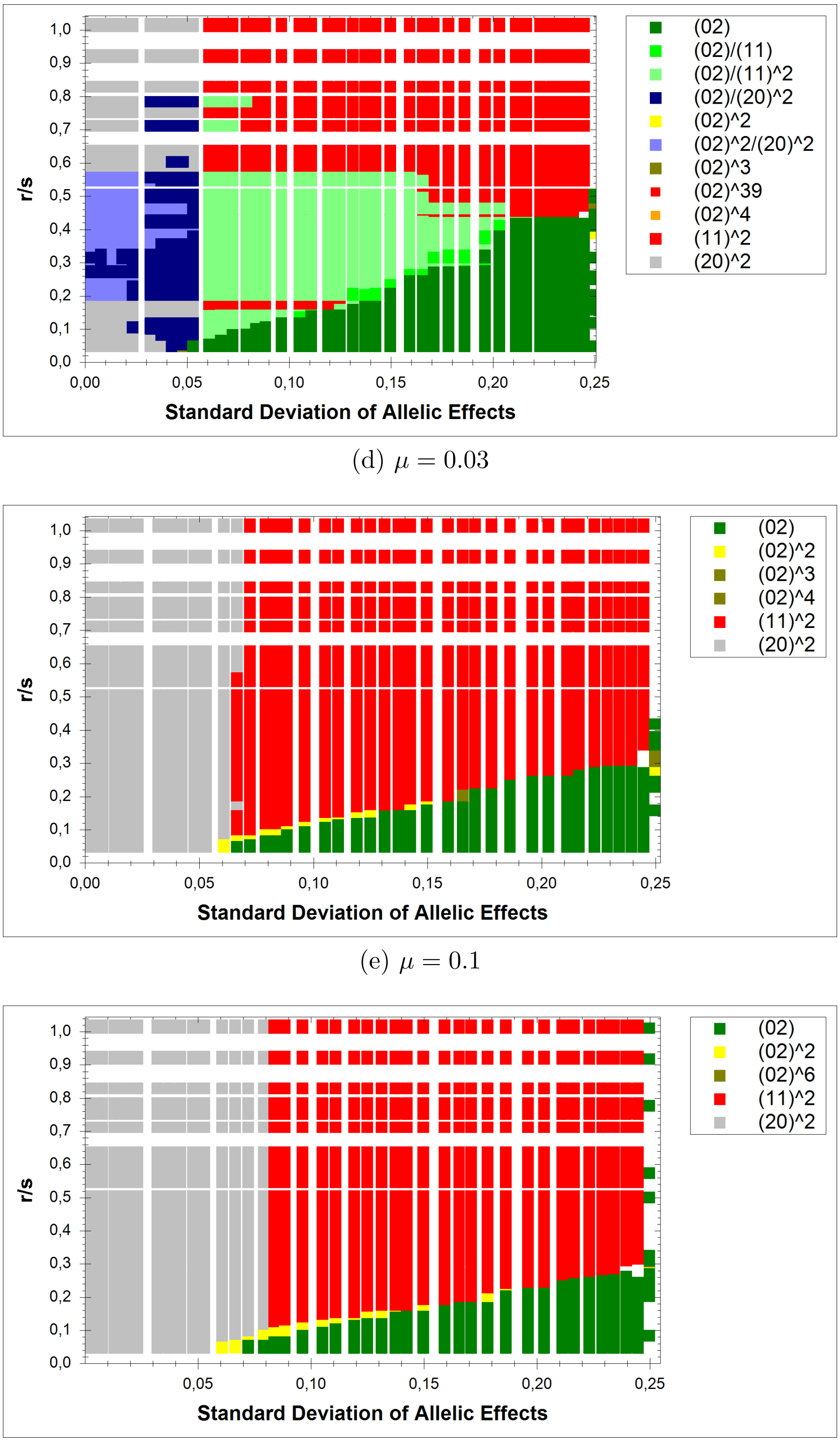


(a) $\mu = 0$

(b) $\mu = 0.001$

(c) $\mu = 0.01$

Figure 4.5: Numerically determined equilibrium structure.



(d) $\mu = 0.03$

(e) $\mu = 0.1$

(f) $\mu = 1$

Figure 4.5: Numerically determined equilibrium structure as a function of the standard deviation of allelic effects and $r/\hat{s}$ for different migration rates. The colors refer to different equilibrium structures. We only depict equilibrium types according to their gametic composition in the meta population and not for each deme separately. An expressions of the form (a, b) refers to an equilibrium type in the meta population. As above, see Figure 4.4, $\hat{}\, n$ denotes the number of equilibria of this type found for the different simulated initial values.

Alleles / Loci / Demes	2 / 3 / 2
Max Iterations	$300,000$
Error / Stagnation Count	10^{-10} / 10
Allele Loss	10^{-4}
Initial Values	40 / $\epsilon_E = 0.25$
Average Allelic Contributions	$(\bar{\gamma}_{\mathbf{1}}, \bar{\gamma}_{\mathbf{2}}) = (0, 0.5)$ / 40 sets
Quadratic Optimum Positions	0.45/0.55
Normalized Selection Coefficients	$\mathscr{U}(0.25, 1, 10)$
Recombination Rates	$\mathscr{U}(0, 0.5, 10)$
Migration Rates	$\mu \in \{0, 0.001, 0.01, 0, 1, 1\}$
Population Regulation	Soft selection

Table 4.5: Parameters and settings for the simulation of a two deme population with three diallelic loci, subject to quadratic stabilizing selection with symmetrically displaced optimum and Deakin migration.

per generation, see the error property for the simulation run in Table 4.3.

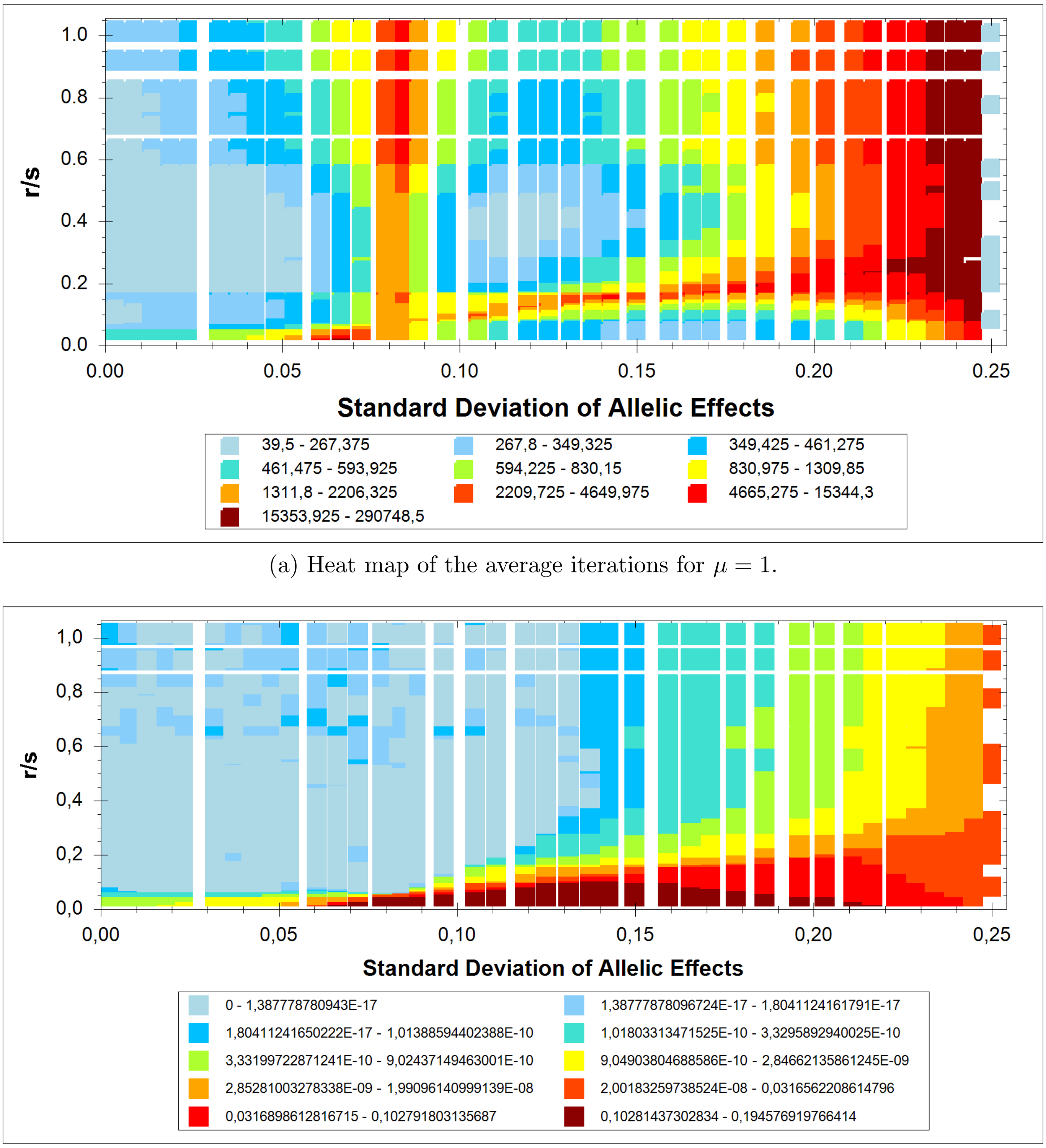
4.2.1 Additional Numerical Investigations for Two Demes

This section is intended to check our prior made assumptions to model migration. Therefore, we ran additional simulations varying single assumptions. We will discuss them, presenting the gathered data and comparing it to our prior analysis.

3 Diallelic Loci For the prior analysis, we restricted ourselves to the case of two diallelic loci. From the work of Bürger and Gimelfarb (1999), and Gimelfarb (1998) we already know, how the genetic variance, linkage disequilibrium, or the polymorphic fraction of the genomes are affected by an increasing number of contributing loci. Thus, we ran another simulation for three diallelic loci and compared the collected results, stated in Table 4.6, to the ones achieved for the case of two loci. Table 4.5 provides an overview on the simulation settings.

Similar to the results attained by Bürger and Gimelfarb (1999), the total genetic variance is smaller for three than for two diallelic loci. This applies to all tested migration rates. The polymorphic fraction of the genome declines even stronger for increasing migration than in the case of two loci. As for the case of two diallelic loci, the average linkage disequilibrium increases for growing migration rate until migration reaches $\mu = 0.01$, then decreases for even stronger migration. This suggests that the reason for this behavior lies in the ratio of migration to selection and recombination and the specific optimum position and is unrelated to the number of loci. As we shall see later on, the optimum position has no influence on this behavior, as we also obtained the same result for a higher shift of the optimum, see Table 4.9.

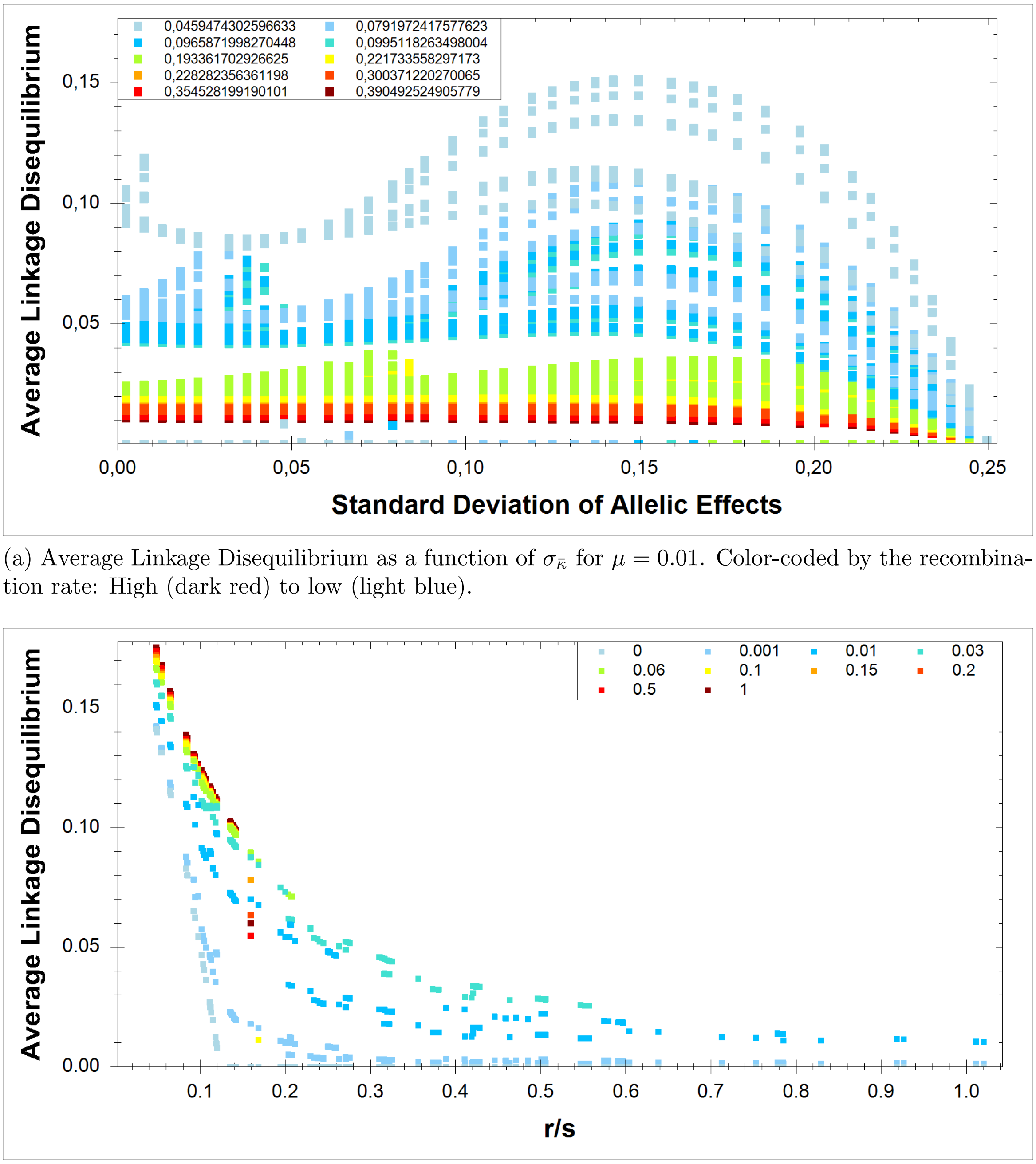
Since increasing the recombination strength decreases linkage equilibrium, cf. Figure 4.7a, the relation to recombination and selection is crucial and must be included to completely analyze the observed behavior.



(a) Heat map of the average iterations for $\mu = 1$.

(b) Heat map of the average linkage disequilibrium for $\mu = 1$

Figure 4.6: Average iterations until equilibrium was reached and average linkage disequilibrium for the Levene model in a two deme population with symmetrical optimum displacement $d = 0.05$. Heat map as a function of the standard deviation of allelic effects and the ratio $r/\tilde{s}$. Dark red refers to the highest values, light blue to the lowest values. Every data point reflects the average value over 40 initial values for one parameter combination $(r, \tilde{s}, \bar{\gamma})$.



(a) Average Linkage Disequilibrium as a function of $\sigma_{\bar{\kappa}}$ for $\mu = 0.01$. Color-coded by the recombination rate: High (dark red) to low (light blue).

(b) Average Linkage Disequilibrium as a function of the ratio $r/\tilde{s}$ for a specific case of allelic effects $\sigma_{\bar{\kappa}} = 0.1496$ for different migration rates (indicated by different colors).

Figure 4.7: Average linkage disequilibrium for two loci. Every data point reflects the average over 40 initial values for one parameter combination $(r, \tilde{s}, \bar{\gamma})$.

μ	Polymorphic Loci				Polymorphic	Linkage	Genetic
	0	1	2	3	Fraction	Disequilibrium	Variance
0	0.0517	0.1861	0.0287	0.7335	0.8146	0.0011	0.0416
0.001	0.0726	0.2071	0.0233	0.697	0.7815	0.0035	0.0407
0.01	0.1596	0.48	0.0149	0.3455	0.5154	0.0067	0.0386
0.1	0.3443	0.5886	0.0274	0.0398	0.2542	0.0027	0.0329
1	0.3915	0.5473	0.0269	0.0343	0.2347	0.0026	0.0312

μ	Q_{ST}	Variance of Gametes among Subpopulations	Mean Fitness	Gametic Variance of Fitness
0	0.4984	0.0393	0.9354	0
0.001	0.3837	0.03	0.9356	0.0001
0.01	0.1722	0.0086	0.932	0.0003
0.1	0.021	0.0005	0.9292	0.0002
1	-	-	0.9258	0.0007

Table 4.6: Resulting measures for 3 diallelic loci listed for the different migration rates. Averages are always taken over 40 initial values and 4000 parameter combinations $(r, \tilde{s}, \bar{\gamma})$.

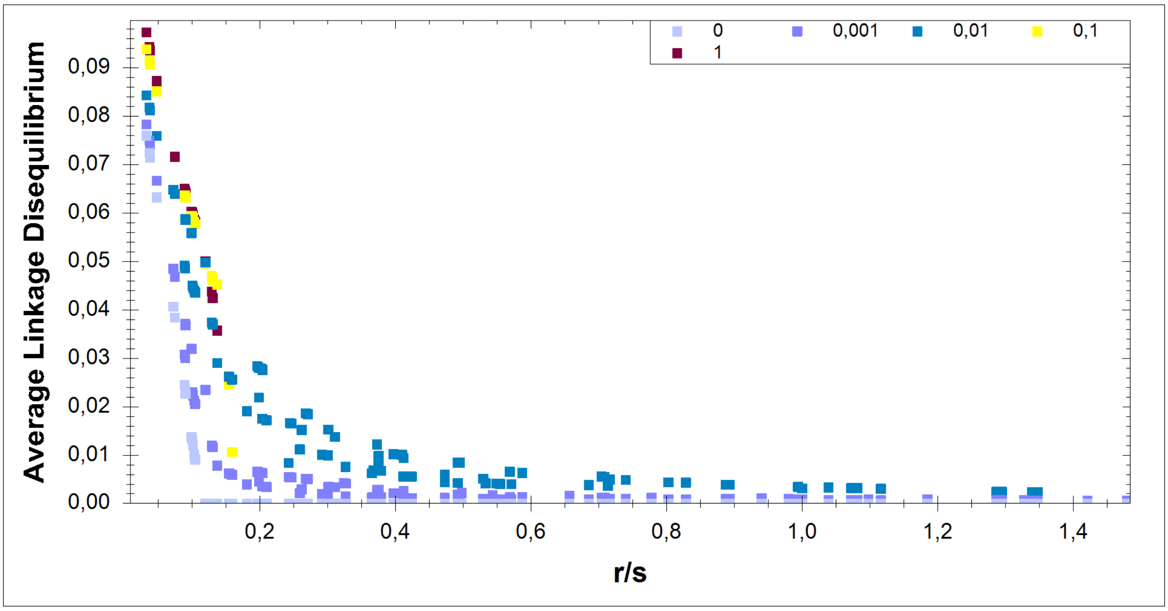


Figure 4.8: Average linkage disequilibrium for three loci as a function of the ratio $r/\tilde{s}$. Plotted for a specific case of allelic effects $\sigma_{\bar{\kappa}} = 0.1637$ and color-coded by different migration rates. Every data point reflects the average over 40 initial values for one parameter combination $(r, \tilde{s}, \bar{\gamma})$.

Alleles / Loci / Demes	2 / 2 / 2
Max Iterations	$300,000$
Error / Stagnation Count	10^{-10} / 10
Allele Loss	10^{-4}
Initial Values	40 / $\epsilon_E = 0.25$
Average Allelic Contributions	$(\bar{\gamma}_\mathbf{1}, \bar{\gamma}_\mathbf{2}) = (0, 0.5)$ / 20 sets
Quadratic Optimum Positions	$d \in \{0, 0.01, 0.03, 0.05, 0.07, 0.1, 0.15, 0.2\}$
Normalized Selection Coefficients	$\mathscr{U}(0.25, 1, 5)$
Recombination Rates	$\mathscr{U}(0, 0.5, 5)$
Migration Rates	$\mathscr{U}(0, 0.1, 5)$
Population Regulation	Soft selection

Table 4.7: Parameters and settings for the simulation of a two deme population with two diallelic loci, exploring multiple symmetrically displaced optima positions for quadratic stabilizing selection.

We restricted the data to a specific set of allelic effects to reduce dimensionality and enable us to analyze the relation of the migration rate, selection and recombination strength to linkage disequilibrium. Averaged over all recombination rates, migration rates and selection coefficients, we located the random genetic setup (allelic effects) in our data, which exhibits the highest linkage disequilibrium in the two and three locus case ($\sigma_{\bar{\kappa}} = 0.1496$ and 0.1637, respectively). Restricted to this genetic setup, we plotted the linkage disequilibrium for different migration rates as a function of the ratio r/s, Figure 4.7 and Figure 4.8. As expected, the linkage disequilibrium declines for stronger recombination relative to selection. If recombination relative to selection is weak enough, e.g. $r/s < 0.16$ for the case $\mu = 1$, the linkage disequilibrium is higher for stronger migration. But for stronger recombination relative to selection, linkage disequilibrium for weaker migration exceeds the value for stronger migration rates. Of course, averaging over all recombination rates and selection coefficients results in lower linkage disequilibrium for stronger migration.

Optimum Position We used a specific symmetrical positioning of the optimum in the two demes so far, i.e., $d = 0.05$. Here, we will explore the impact of this specific choice. Therefore, we first simulated two diallelic loci with symmetrical displaced optimum for 5 randomly chosen migration rates. To reduce the dimensionality of the possible parameter combinations, we only simulated for 20 different allelic contributions, 5 random selection coefficients and 5 random recombination rates. Since, already for a migration rate of $\mu = 0.1$, the data suggested the population to be spatially homogeneous for the case $d = 0.05$, cf. Table 4.4, we restricted the 5 random uniformly distributed migration rates to $\mu \leq 0.1$. Settings for this simulation run are outlined in Table 4.7.

For the interpretation of the results, we have to keep in mind that we restricted the migration rate. For growing disparity of the optima, the polymorphic fraction, genetic variance, gametic variance, mean fitness, and the variance of the gametes

p_O	Polymorphic Loci			Polymorphic	Linkage	Genetic
	0	1	2	Fraction	Disequilibrium	Variance
0.5/ 0.5	0.2449	0.2961	0.459	0.607	0.0405	0.0472
0.49/ 0.51	0.2412	0.2955	0.4633	0.6111	0.0398	0.0476
0.47/ 0.53	0.2171	0.2545	0.5284	0.6557	0.0381	0.0495
0.45/ 0.55	0.1818	0.2492	0.569	0.6936	0.0351	0.0522
0.43/ 0.57	0.1505	0.2503	0.5992	0.7244	0.0324	0.0552
0.4/ 0.6	0.1035	0.2718	0.6247	0.7606	0.0287	0.0601
0.35/ 0.65	0.0263	0.3088	0.6648	0.8193	0.0232	0.0696
0.3/ 0.7	0.002	0.2576	0.7404	0.8692	0.0178	0.0815

p_O	Q_{ST}	Variance of Gametes among Subpopulations	Mean Fitness	Gametic Variance of Fitness
0.5/ 0.5	0.0177	0.0066	0.8672	0.0006
0.49/ 0.51	0.0269	0.0081	0.8724	0.0006
0.47/ 0.53	0.0745	0.0143	0.8831	0.0006
0.45/ 0.55	0.1365	0.0204	0.8937	0.0005
0.43/ 0.57	0.2019	0.0258	0.9039	0.0005
0.4/ 0.6	0.2948	0.0325	0.9176	0.0006
0.35/ 0.65	0.4248	0.0401	0.9353	0.0007
0.3/ 0.7	0.5263	0.0441	0.9473	0.001

Table 4.8: Resulting measures for different symmetrical displaced optima positions, random migration, recombination, and selection. Averages are always taken over 40 initial values and 2500 parameter combinations $(r, \tilde{s}, \bar{\gamma}, \mu)$.

μ	Polymorphic Loci			Polymorphic	Linkage	Genetic
	0	1	2	Fraction	Disequilibrium	Variance
0	0.0349	0.1417	0.8234	0.8943	0.0062	0.063
0.001	0.0378	0.1511	0.8111	0.8867	0.0113	0.0612
0.01	0.0701	0.216	0.7139	0.8219	0.0289	0.0606
0.1	0.203	0.6117	0.1853	0.4912	0.0258	0.0501
0.2	0.2516	0.5821	0.1662	0.4573	0.0256	0.0472
0.5	0.303	0.5369	0.1601	0.4285	0.0258	0.0443
1	0.3308	0.5093	0.1598	0.4145	0.0261	0.0431

μ	Q_{ST}	Variance of Gametes among Subpopulations	Mean Fitness	Gametic Variance of Fitness
0	0.6541	0.0878	0.9439	0
0.001	0.5649	0.0786	0.9443	0
0.01	0.3516	0.0397	0.9378	0.0003
0.1	0.074	0.0037	0.9263	0.0005
0.2	0.0314	0.0016	0.9227	0.0009
0.5	0.0042	0.0002	0.9172	0.0015
1	-	-	0.9135	0.0021

Table 4.9: Resulting measures for higher deviation in optimum positions, i.e., $d = 0.1$ for different migration rates. Averages are always taken over 40 initial values and 4000 parameter combinations $(r, \tilde{s}, \bar{\gamma})$.

among subpopulations increased, while linkage disequilibrium decreased. Clearly, the growing difference in fitness landscapes results in a higher genetic disparity within subpopulations. Note that this is the first case analyzed so far, showing a simultaneous increase of mean fitness and genetic variance. This can be explained by the fact that for higher deviation of the optimum from symmetry in each deme, the fixation of a homozygote becomes more probable, which increases the mean fitness. On the other hand, the disposition of the optima in opposing directions results in different subpopulation compositions.

So far the question remains unsolved, whether the characteristics of the linkage disequilibrium, as found, is specific for the discussed deviation of the optima from symmetry $d = 0.05$. To conclude, whether it is reproducible for a higher deviation, we conducted another simulation run for the case $d = 0.1$. Aside from that, we chose the same parameter settings as in Table 4.3. The results show the same behavior, although the decline for migration rates $\mu \geq 0.1$ is weaker, see Table 4.9.

5 Summary

This work presents a developed software to simulate and analyze migration-selection models in a multilocus population. Using the implemented program, a subdivided population in two demes, exhibiting two diallelic loci under quadratic stabilizing selection, was investigated. Thereby, we addressed the following questions by numerical simulation of the discrete, dynamical system:

- Can migration account for higher genetic variability and and admit a higher fraction of polymorphic equilibria than in the panmictic population?
- How strong or weak must migration be relative to selection, to create results which can be understood as a pertubation of the weak selection limit or the system without migration?

The object-oriented implementation was based on the mathematical model introduced in Section 1.2. This allowed for an arbitrary number of loci and alleles, as well as population subdivision in any number of demes. We considered three different fitness functions (Gaussian, quadratic, linear) to model viability selection, and allowed for recombination neglecting interference and position effects. Two migration models, the Deakin model and the stepping-stone migration model, were considered. Population regulation comprised soft and hard selection.

Based on the review of available simulation tools in Section 2.1, the implementation focused on extensibility and the integration of analytical tools. The object-oriented design of the program, based on the mathematical life cycle stated in (1.10), and the software architecture were presented in Section 3.2.

Implementation of the mathematical model raised algorithmic issues, which were adressed in Section 3.3. These concerned calculation of recombination rates, initial and fitness value construction, and the fast reconstruction of the allelic composition from a gamete index.

A review of results on quadratic stabilizing selection in Section 2.2 motivated the numerical investigation. The equilibrium structure of the diallelic two-locus model with the optimum attained by the double heterozygote is completely understood, cf. Section 2.2.1. This holds even for arbitrary recombination and asymmetrical allelic effects. For a displaced optimum, additional assumptions were made in previous work to achieve analytical results. For example, parameter conditions which ensure existence of a polymorphic equilibrium can be derived, given strong selection relative to recombination.

This motivated the first simulation in Section 4.1. A diallelic two-locus model of a panmictic population under quadratic stabilizing selection. The recombination rate, selection strength, and allelic effects were arbitrary, randomly chosen from uniform distributions. The system was simulated for ten optimum positions, covering the scope from the symmetrical model to directional selection.

As the optimum position changes in steps from the symmetrical model to directional selection, the genetic variance and average linkage disequilibrium were found to decrease. Also, the average polymorphic fraction of the genome is maximized in the symmetrical case. On the other hand, mean fitness increases, reaching its maximum possible value in the case of the optimum at the homozygote.

Numerically determined equilibrium structures showed, how already small deviations of the optimum reduced the areas of existence of polymorphisms drastically.

As was only proven before for the symmetrical case, no more than two stable coexisting equilibria were found. This was observed in the numerical results of all tested optimum positions.

A second simulation of a subdivided population in two demes was performed in Section 4.2, assuming a homogeneous Deakin migration model. Again, quadratic stabilizing selection was assumed, this time with symmetrical displaced optima within the demes. For increasing migration rate, the average polymorphic fraction declined rapidly, while the population tended to quasi homogeneity. Already for a migration rate $\mu = 0.1$, i.e., one tenth of all individuals migrate, the numerically calculated equilibrium structure coincided with the one of the Levene model. We found that the Levene model exhibits the same structure as observed in the symmetric case for a panmictic population.

For an increasing migration rate, the average genetic variance and the average polymorphic fraction of the genome declined.

To check the assumptions of the second simulation run, additional simulations were performed and analyzed in Section 4.2.1. These included a simulation of the same setup for three diallelic loci, resulting in an even faster decline of the average polymorphic fraction of the genome for growing migration rate. Another simulation run tested the setup for eight different symmetrical optimum deviations. Higher disparity of the optima resulted in higher genetic disparity within subpopulations, as measured by Q_{ST} and the variance of gametes among subpopulations.

Bibliography

Bürger, R. (2000). *The Mathematical Theory of Selection, Recombination, and Mutation.* Wiley, Chichester.

Bürger, R. (2009). *Multilocus selection in subdivided populations I. Convergence properties for weak or strong migration. J. Math. Biol.*, 58, pp. 939–978.

Bürger, R. and Gimelfarb, A. (1999). *Genetic Variation Maintained in Multilocus Models of Additive Quantitative Traits Under Stabilizing Selection. Genetics*, 152, pp. 807–820.

Deakin, M. (1966). *Sufficent Conditions for Genetic Polymorphism. The American Naturalist*, 100, pp. 690–692.

Drayton, P.; Albahari, B.; and Neward, T. (2003). *C# in a nutshell.* OReilly Verlag GmbH & Co. KG.

Edelaar, P. and Björklund, M. (2011). *If F_{ST} does not measure neutral genetic differentiation, then comparing it with Q_{ST} is misleading. Or is it? Mol Ecol.*, 20, pp. 1805–1812.

Endler, J. (1986). *Natural selection in the wild.* Princeton University Press, Princeton, New Jersey, U.K.

Ewens, W. (2004). *Mathematical Population Genetics.* Springer Verlag, New York, Inc.

Futuyma, D. (2005). *Evolution.* Sinauer Associates, Inc., Sunderland, Massachusetts, U.S.A.

Gale, J. and Kearsey, M. (1968). *Stable equilibria under stabilising selection in the absence of dominance. Heredity*, 23, pp. 553–561.

Gavrilets, S. and Hastings, A. (1993). *Maintenance of Genetic Variability Under Strong Stabilizing Selection: A Two-Locus Model. Genetics*, 134, pp. 377–386.

Gimelfarb, A. (1998). *Stable Equilibria in Multilocus Genetic Systems: A Statistical Investigation. Theor. Popul. Biol.*, 54, pp. 133–145.

Guillaume, F. and Rougemont, J. (2006). *Nemo: an evolutionary and population genetics programming framework. Bioinformatics*, 22, pp. 2556–2557.

Hardy, O. and Vekemans, X. (1999). *Isolation by distance in a continuous population: reconciliation between spatial autocorrelation analysis and population genetics models. Heredity*, 83, pp. 145–154.

Hastings, A. (1987). *Monotonic Change of the Mean Phenotype in Two-Locus Models. Genetics*, 117, pp. 583–585.

Hudson, R. (2002). *Generating samples under a Wright-Fisher neutral model of genetic variation. Bioinformatics*, 18, pp. 337–338.

Kansy, T. (2010). *Datenbankprogrammierung mit .NET 4.0.* Carl Hanser Verlag Mnchen.

Karlin, S. (1982). *Classifications of Selection-Migration Structures and Conditions for a Protected Polymorphism. Evol. Bio.*, 14, pp. 61–204.

Karlin, S. and Feldman, W. M. (1970). *Linkage and selection: two locus symmetric viability model. Theor. Popul. Biol.*, 1, pp. 39–71.

Kearsey, M. and Gale, J. (1968). *Stable equilibria under stabilising selection in the absence of dominance: an additional note. Heredity*, 23, pp. 617–620.

Kimura, M. and Weiss, G. (1964). *The Stepping Stone Model of Population Structure and the Decrease of Genetic Correlation with Distance. Genetics*, 49, pp. 561–576.

Lambert, B. W.; Terwilliger, J. D.; and Weiss, K. M. (2008). *ForSim: a tool for exploring the genetic architecture of complex traits with controlled truth. Bioinformatics*, 24, pp. 1821–1822.

LaSalle, J. (1976). *The Stability of Dynamical Systems.* Hamilton Press, Berlin, New Jersey, U.S.A.

Nagylaki, T. (1989). *The Maintenance of Genetic Variability in Two-Locus Models of Stabilizing Selection. Genetics*, 122, pp. 235–248.

Nagylaki, T. (1992). *Introduction to Theoretical Population Genetics.* Springer-Verlag.

Nagylaki, T. (1993). *The Evolution of Multilocus Systems Under Weak Selection. Genetics*, 134, pp. 627–647.

O'Fallon, B. (2010). *TreeSimJ: a flexible, forward time population genetics simulator. Bioinformatics*, 26, pp. 2200–2201.

Parreira, B.; Trussard, M.; Sousa, V.; Hudson, R.; and Chikhi, L. (2009). *SPAms: A user-friendly software to simulate population genetics data under complex demographic models. Molecular Ecology Resources*, 9, pp. 749–753.

Ridley, M. (2004). *Evolution.* Blackwell Science Ltd a Blackwell Publishing company.

Sanford, J.; Baumgardner, J.; Brewer, W.; Gibson, P.; and ReMine, W. (2007). *Mendel's Accountant: A New Population Genetics Simulation Tool for Studying Mutation and Natural Selection. SCPE*, 8, pp. 147165.

Spitze, K. (1993). *Population Structure in Daphnia obtusa: Quantitative Genetic and Allozymic Variation. Genetics*, 135, pp. 367–374.

Wright, S. (1935). *Evolution in populations in approximate equilibrium. J. Genetics*, 30, pp. 257–266.

Wright, S. (1943). *Isolation by distance. Genetics*, 28, pp. 114–138.

Curriculum Vitae

Personal Data:

Name	Peter Kepplinger
Date of Birth	27^{th} of August, 1980
Place of Birth	Salzburg, Austria
Nationality	Austria

Education:

09/1990-06/1999	HIB Saalfelden, 5760 Saalfelden
10/2001-04/2012	Diploma Studies in Mathematics, University of Vienna

Subject-Specific Work Experience:

2002-2007	Tutor for Mathematics at team-plus! Lernhilfe-Institut, 1040 Wien, Favoritenstr. 70 / 14
02/2008-06/2009	Technical Writer / Developer at UC4 Senactive Software GmbH, 1040 Wien, Prinz-Eugen-Str. 72
07/2009-02/2010	Application Analyst at UC4 Senactive Software GmbH, 1040 Wien, Prinz-Eugen-Str. 72
09/2010-11/2010	Internship as Analyst for UC4 Senactive Software GmbH, 1040 Wien, Prinz-Eugen-Str. 72

Publications:

Obweger, H.; Schiefer, J; Suntinger, M.; Kepplinger, P.; and Rozsnyai, S.:
”User-oriented rule management for event-based applications”. DEBS 2011: 39-48

Obweger, H.; Schiefer, J; Kepplinger, P.; and Suntinger, M.:
”Discovering Hierarchical Patterns in Event-Based Systems”. IEEE SCC 2010: 329-336