May 2017

Infinite-Dimensional Traits: Estimation of Mean, Covariance, and Selection Gradient of Tribolium Castaneum Growth Curves

Ly Viet Hoang
University of Wisconsin-Milwaukee

Infinite-dimensional Traits: Estimation of Mean, Covariance, and Selection Gradient of Tribolium Castaneum Growth Curves

by

Ly Viet Hoang

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Mathematics

at

The University of Wisconsin - Milwaukee

May 2017

Abstract

Infinite-dimensional Traits: Estimation of Mean, Covariance, and Selection Gradient of Tribolium Castaneum Growth Curves

by

Ly Viet Hoang

The University of Wisconsin - Milwaukee, 2017

Under the Supervision of Professor Jay H. Beder

In evolutionary biology, traits like growth curves, reaction norms or morphological shapes cannot be described by a finite vector of components alone. Instead, continuous functions represent a more useful structure. Such traits are called function-valued or infinite-dimensional traits. Kirkpatrick and Heckmann outlined the first quantitative genetic model for these traits. Beder and Gomulkiewicz extended the theory on the selection gradient and the evolutionary response from finite- to infinite-dimensional traits.

Rigorous methods for the estimation of these quantities were developed throughout the years. In his dissertation, Baur defines estimators for the mean and covariance function, as well as for the selection gradient based on two different assumptions. First, it is assumed that all individuals are independent. The second case considers a sample of independent families of equally related individuals. In this thesis, results of the estimations based on data on Tribolium Castaneum larvae will be stated.

Estimations of the pre-selection mean, the evolutionary response to selection, and the phenotypic covariance function were run for five consecutive generations - once assuming that all larvae are independent and once for independent families of full-siblings. Using the pre-selection mean and the evolutionary response to selection, the mean function among newborns of the successive generation is computed. The selection gradient is not explicitly estimated as it is contained in the computation of the evolutionary response to selection.

The differences in results from using Ornstein-Uhlenbeck and Wiener covariance functions are examined. It becomes evident that the choice of the candidate covariance function heavily impacts the results of the estimation. With respect to this observation, alternative ways to find a suitable candidate covariance function, based on the provided data, are discussed.

TABLE OF CONTENTS

LIST OF FIGURES

Acknowledgements

First and foremost, I want to thank my advisor Professor Jay Beder for his never-ending support. His continuous guidance throughout this last year always encouraged me to give my best. He showed me ways to steadily improve myself, and especially the ideas from discussions we had together made me strive for the best, and molded this thesis to what it is today. Moreover, my appreciation goes towards the additional committee members Professor Gabriella Pinter and Professor Richard Stockbridge, as well as to Tyler Baur, whose dissertation , for his involvement and willingness to help whenever it was needed.

I am grateful to the Department of Mathematical Sciences at the University of Wisconsin - Milwaukee and the Mathematics Department at Ulm University for the chance of being part in this double-degree program. I've gathered irreplaceable experience living in the United States of America. It undoubtedly became clear to me why this program stood its ground for more than twenty years to this date.

Special thanks go to my friends, colleagues and fellow TAs for keeping me company and making every day a joy. Lastly, I would also like to thank my parents, my brother, and my friends back home in Germany for their endless support from across the world.

1. Introduction

Evolutionary biology concerns itself with the change of physical traits between generations. These traits are also called *phenotypes*. In classical evolutionary biology, a *vector-valued* or *finite-dimensional trait* $\boldsymbol{z}$ is a trait that can be represented by a finite vector $(z_1, \ldots, z_n)$, where the z_i is the i^{th} component that contributes to an individual's phenotype. Examples are the crop yield of a plant, the amount of milk produced by a cow, or the height and weight of an individual at some critical age.

However, often traits such as *growth trajectories*, *morphological shapes* like the shape of a wing, or *reaction norms*, i.e. traits that react to environmental variables, as for example the speed of lizards depending on the ambient temperature, are of interest. Therefore it makes more sense to describe those phenotypes by *continuous functions*. Such traits are called *function-valued* traits, denoted by $\{z(t) : t \in T\}$, where T could represent time when describing growth, or angles and distances when talking about shapes, or temperature that influences reaction. Due to continuity of those functions, describing a function-valued trait would take infinitely many measurements, hence the alternative name *infinite-dimensional trait*, and traits like these are treated are modeled by *stochastic processes* rather than *random variables*. In their paper of 1989, Kirkpatrick and Heckmann [12] define a quantitative genetic model for infinite-dimensional traits, which is the basis for all future computation.

The evolutionary change of the mean of traits, whether finite-dimensional or function-valued, is subject to selection between generations. This *evolutionary response to selection* is characterized by the *selection gradient* β. Information on the selection gradient can be obtained by analyzing the *fitness function* W of a trait. In the vector-valued case, the selection gradient β is a finite-

dimensional vector that can be estimated by regressing observed fitness on observed traits. In the framework of *reproducing kernel Hilbert spaces* and *Gaussian processes* the definition of the selection gradient is extended to function-valued traits by Beder and Gomulkiewicz [10] and further computational methods were developed.

This thesis is structured in the following way. Chapter 2 takes a look at finite-dimensional traits and derives the selection gradient for this case. Chapter 3 focuses on reproducing kernel Hilbert spaces and Gaussian processes, as they are important for the estimation of the selection gradient for infinite-dimensional traits. The definition of this special kind of Hilbert space and how Gaussian processes are connected to them will be introduced. In particular, the Gaussian Dichotomy Theorem will be stated, since it gives necessary and sufficient conditions for the equivalence of Gaussian probability measures. This is an important cornerstone for the estimation of the selection gradient. Chapter 4 will make use of this theory, extending the idea of the selection gradient from Chapter 2 to infinite-dimensional traits. Built on the knowledge from the previous chapters, Chapter 5 gives estimators for all functions of interest introduced in Chapter 4, in particular for the estimation of the pre-selection and next-generation mean functions $\bar{z}$ and $\bar{z}'$, as well as for the phenotypic covariance function P. Furthermore, one distinguishes between a sample of all independent observations, and a sample of independent families of equally related organisms. Following this, in Chapter 6, the estimation methods are applied to real-life data on Tribolium Castaneum larvae. Estimates of the pre-selection mean, the evolutionary response to selection which dictates the next-generation mean, and the phenotypic covariance function are computed. Even though it plays an essential role, the selection gradient β is not explicitly estimated, as it is an functional and therefore cannot be illustrated alone, as well as the fact that it is contained in the estimation of the evolutionary response to selection. The results are discussed and alternative approaches are discussed in Chapter 7. Finally, in the last chapter, all results of this thesis will be summarized and suggestions on open questions will be stated.

2. Finite-dimensional Traits

In this chapter, the necessary biological terminology for this thesis and results of classical evolutionary biology are introduced. An observable physical trait is called a *phenotype*. The genetic makeup of an organism on the other hand is called *genotype*. The analysis of *evolutionary changes between generations* has to rely on the observation of phenotypes, as it is in general not possible to observe genotypes.

The *selection gradient* β, which characterizes the evolutionary response to selection, will be defined for finite-dimensional traits. The *Breeder's Equation*, which relates the means of the phenotypes in the parent and offspring generation, as well as the *Robertson-Price Identity*, which puts the *fitness* W of an organism into relation with the selection gradient β, are essential results for the estimation of β.

Remark 2.1.(Conventions) This thesis will follow the conventional notation of evolutionary biologists as close as possible. Like Beder and Gomulkiewicz [9], the following notation is used.

In a mathematical sense, the terms *trait* and *phenotype* are regarded as a random variable and its realization, respectively. Denote column vectors $\boldsymbol{z}$ and matrices $\mathbf{P}$ by boldface characters and the transpose by ^{T}. The multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is denoted by $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

In the context of evolutionary biology, every generation's mean is denoted differently:

$\bar{\boldsymbol{z}}$ = mean of the trait $\boldsymbol{z}$ among newborns of the current generation before selection (pre-selection mean),

$\bar{\boldsymbol{z}}^*$ = mean of the trait $\boldsymbol{z}$ of the current generation (after selection),

$\bar{\boldsymbol{z}}'$ = mean of the trait $\boldsymbol{z}$ among newborns of the following generation

The notation $\bar{\boldsymbol{z}}$, i.e. the mean of a $\boldsymbol{z}$, refers to the expected value of $\boldsymbol{z}$, not the arithmetic mean. The probability density of each generation is indexed by the corresponding mean, e.g. $p_{\bar{\boldsymbol{z}}}$ is the probability density of the trait among newborns of the current generation before selection, i.e. the pre-selection mean.

Furthermore, to establish a genetic model, it is assumed that evolution proceeds in two steps.

(1) Selection determined by the *fitness* (also survivorship) of each individual, and

(2) Inheritance controlled by mating patterns and genetics of the breeding adults (survivors after selection).

Lastly, it is assumed that traits are *autosomally* inherited. This means that any effects of random genetic drift, mutation, epistasis[1] or recombination are neglected.

Let $\boldsymbol{z}$ be a finite-dimensional trait of an individual. We may decompose $\boldsymbol{z}$ as the sum of two uncorrelated random variables (vectors),

$$\boldsymbol{z} = \boldsymbol{g} + \boldsymbol{e}, \tag{2.1}$$

where $\boldsymbol{g}$ represents the *additive-genetic component* of the trait inherited by the parents and $\boldsymbol{e}$ represents *environmental effects*. In a sample, $\boldsymbol{z}_i = \boldsymbol{g}_i + \boldsymbol{e}_i$ denotes the trait of the i^{th} individual. Since all individuals are assumed to live in similar and independent environments, the environmental effects $\boldsymbol{e}_i$ are all independent and identically distributed. The *pre-selection distribution* is assumed to be normal with mean $\bar{\boldsymbol{z}}$ and (phenotypic) covariance matrix $\mathbf{P}$. As described before, $p_{\bar{\boldsymbol{z}}}$ denotes the *pre-selection density*.

In this model, the *phenotypic covariance matrix* $\mathbf{P}$ of a trait can be decomposed into the sum of a *additive-genetic covariance matrix* $\mathbf{G}$ and an *environmental covariance matrix* $\mathbf{E}$,

$$\mathbf{P} = \mathbf{G} + \mathbf{E}, \tag{2.2}$$

[1] Dependence between genes.

each corresponding to the respective component of the trait.

The fitness $W(\boldsymbol{z})$ of an individual with trait $\boldsymbol{z}$ (or just W if the circumstances are clear) is a random variable fulfilling the following assumptions:

(i) $W > 0$

(ii) $\mathrm{Var}_{\bar{\boldsymbol{z}}}[W] < \infty \quad$ for all $\bar{\boldsymbol{z}} \in \mathbb{R}^n$.

We define the *relative fitness* w by the quotient

$$w(\boldsymbol{z}) = \frac{W(\boldsymbol{z})}{\mathbb{E}_{\bar{\boldsymbol{z}}}[W]}, \tag{2.3}$$

and the *post-selection density* $p_{\bar{\boldsymbol{z}}^*}$ is given by

$$p_{\bar{\boldsymbol{z}}^*} = w(\boldsymbol{z}) p_{\bar{\boldsymbol{z}}} = \frac{W(\boldsymbol{z}) p_{\bar{\boldsymbol{z}}}}{\mathbb{E}_{\bar{\boldsymbol{z}}}[W]}. \tag{2.4}$$

Note that the assumptions made on W guarantee that $p_{\bar{\boldsymbol{z}}^*}$ is a positive function and integrates to 1, and therefore a probability density. The pre-selection distribution is assumed to be $N(\bar{\boldsymbol{z}}, \mathbf{P})$. Observe that the post-selection distribution does not necessarily have to be normal.

The *selection differential*, which describes the *within-generation change* in the mean trait, i.e. the changes from pre-selection to post-selection, is

$$\boldsymbol{s} = \bar{\boldsymbol{z}}^* - \bar{\boldsymbol{z}}. \tag{2.5}$$

The *evolutionary response to selection*, or *between-generation change* in the mean trait, i.e. changes between the newborns of successive generations, is defined by

$$\Delta\bar{\boldsymbol{z}} = \bar{\boldsymbol{z}}' - \bar{\boldsymbol{z}}. \tag{2.6}$$

Under the assumption that the trait $\boldsymbol{z}$ before selection is normally distributed with mean $\bar{\boldsymbol{z}}$ and covariance matrix $\mathbf{P}$, the *Breeder's Equation* holds and describes the evolutionary response to

selection as

$$\Delta\bar{\boldsymbol{z}} = \mathbf{G}\mathbf{P}^{-1}\boldsymbol{s}. \tag{2.7}$$

The breeder's equation directly connects the between-generation change to the within-generation change. In this the *selection gradient* $\boldsymbol{\beta}$ is defined by

$$\boldsymbol{\beta} = \mathbf{P}^{-1}\boldsymbol{s}. \tag{2.8}$$

Thus, equation (4.8) can be written as

$$\Delta\bar{\boldsymbol{z}} = \mathbf{G}\boldsymbol{\beta}. \tag{2.9}$$

The i^{th} component of the selection gradient, β_i, describes the force of directional selection on the i^{th} component of the trait. The estimation of $\boldsymbol{\beta}$ relies on knowledge about the post-selection mean $\bar{\boldsymbol{z}}^*$, as it is part of the computation of the selection differential $\boldsymbol{s}$. Unfortunately, $\bar{\boldsymbol{z}}^*$ is in general hard to determine (and thus so is $\boldsymbol{s}$) except for fully artificially conducted selection. Lande extended the model to natural populations and proved that $\boldsymbol{\beta}$ can be estimated without any information on the post-selection distribution.

Theorem 2.2.(Lande's Theorem, [9] Theorem 1) Let $\boldsymbol{z}$ be a trait with pre-selection distribution $N(\boldsymbol{\mu}, \mathbf{P})$, where the phenotypic covariance matrix $\mathbf{P}$ is positive-definite. If the fitness W is *frequency independent*, i.e. W is independent of $\bar{\boldsymbol{z}}$ or any other parameter of the pre-selection distribution, then

$$\boldsymbol{\beta} = \mathbf{P}^{-1}\boldsymbol{s} = \nabla_{\bar{\boldsymbol{z}}} \log\left(\mathbb{E}_{\bar{\boldsymbol{z}}} W\right) \tag{2.10}$$

where $\nabla_{\bar{\boldsymbol{z}}} = \left(\frac{\partial}{\partial \bar{z_1}}, \ldots, \frac{\partial}{\partial \bar{z_n}}\right)$ is the vector gradient operator at $\bar{\boldsymbol{z}}$.

Note that Lande's theorem presumes that differentiating under the integral sign ([9] Regularity Condition) is allowed. It follows that the evolutionary response to selection can then be computed by

$$\Delta\bar{\boldsymbol{z}} = \mathbf{G}\nabla_{\bar{\boldsymbol{z}}} \log\left(\mathbb{E}_{\bar{\boldsymbol{z}}} W\right). \tag{2.11}$$

In addition, the selection gradient $\boldsymbol{\beta}$ can also be viewed as the vector of *regression coefficients* of a partial regression of the relative fitness w on the trait $\boldsymbol{z}$. In the following denote $\mathbb{E} = \mathbb{E}_{\bar{\boldsymbol{z}}}$. Consider $\hat{w}$ to be the *best linear predictor* of w based on $\boldsymbol{z}$. Then $\hat{w}$ fulfills

(i) $\hat{w} = \beta_0 + \boldsymbol{\beta}^T \boldsymbol{z} = \beta_0 + \sum_{j=1}^{n} \beta_j z_j$,

(ii) $\hat{w}$ minimizes $\mathbb{E}\left[(w - \hat{w})^2\right]$ over all $\hat{w}'$ that satisfy (i).

To solve for β_0 and $\boldsymbol{\beta}$, note that $\hat{w}$ is the orthogonal projection of w into the vector space of random variables spanned by $1, z_1, \ldots, z_n$ (1 is the degenerate random variable at 1) with respect to the inner product $(x, y) = \mathbb{E}[xy]$. It can be easily verified that $w - \hat{w}$ is orthogonal to $1, z_1, \ldots, z_n$, i.e.

$$(w - \hat{w}, 1) = 0, \tag{2.12}$$

$$(w - \hat{w}, z_i) = 0 \quad \text{for all } i = 1, \ldots, n. \tag{2.13}$$

Hence, following equation (2.12) and property (i) of $\hat{w}$

$$\mathbb{E}[w] = \mathbb{E}[\hat{w}] = \beta_0 + \sum_{j=1}^{n} \beta_j \mathbb{E}[z_j] \tag{2.14}$$

and from the equations (2.13) one concludes

$$\mathbb{E}[z_i w] = \mathbb{E}[z_i \hat{w}] = \beta_0 \mathbb{E}[z_i] + \sum_{j=1}^{n} \beta_j \mathbb{E}[z_i z_j] \quad \text{for all } i = 1, \ldots, n. \tag{2.15}$$

Consequently, β_0 and $\boldsymbol{\beta}$ are the solution of the following system of equations

$$\begin{bmatrix} 1 & \mathbb{E}[z_1] & \mathbb{E}[z_2] & \cdots & \mathbb{E}[z_n] \\ \mathbb{E}[z_1] & \mathbb{E}[z_1^2] & \mathbb{E}[z_1 z_2] & \cdots & \mathbb{E}[z_1 z_n] \\ \mathbb{E}[z_2] & \mathbb{E}[z_2 z_1] & \ddots & \cdots & \mathbb{E}[z_2 z_n] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbb{E}[z_n] & \mathbb{E}[z_n z_1] & \cdots & \cdots & \mathbb{E}[z_n^2] \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} = \begin{bmatrix} \mathbb{E}[w] \\ \mathbb{E}[z_1 w] \\ \mathbb{E}[z_2 w] \\ \vdots \\ \mathbb{E}[z_n w] \end{bmatrix} \tag{2.16}$$

Following the same chain of thoughts, but regressing w on $\boldsymbol{z}-\bar{\boldsymbol{z}}$ instead, the best linear predictor of w on $\boldsymbol{z}-\bar{\boldsymbol{z}}$ satisfies

(i) $\hat{w} = \beta_0 + \boldsymbol{\beta}^T(\boldsymbol{z}-\bar{\boldsymbol{z}}) = \beta_0 + \sum_{j=1}^{n} \beta_j(z_j - \bar{z_j})$,

(ii) $\hat{w}$ minimizes $\mathbb{E}\left[(w-\hat{w})^2\right]$ over all $\hat{w}'$ that satisfy (i).

Note, that $\bar{z_i}$ is the mean of the i^{th} component of the trait $\boldsymbol{z}$. $\hat{w}$ is the orthogonal projection of w into the vector space of random variables with basis $1, z_1-\bar{z_1}, \ldots, z_n-\bar{z_n}$ with respect to the inner product $(x,y) = \mathbb{E}[xy]$. Similar to equations (2.14) and (2.15), it follows that

$$\mathbb{E}\left[w\right] = \mathbb{E}\left[\hat{w}\right] = \beta_0 + \sum_{j=1}^{n} \beta_j \underbrace{\mathbb{E}\left[z_j - \bar{z_j}\right]}_{=0} = \beta_0, \tag{2.17}$$

$$\mathbb{E}\left[(z_i-\bar{z_i})w\right] = \mathbb{E}\left[(z_i-\bar{z_i})\hat{w}\right] = \beta_0 \underbrace{\mathbb{E}[z_i-\bar{z_i}]}_{=0} + \sum_{j=1}^{n} \beta_j \mathbb{E}\left[(z_i-\bar{z_i})(z_j-\bar{z_j})\right]$$

$$\mathrm{Cov}(z_i, w) = \sum_{j=1}^{n} \beta_j \mathrm{Cov}(z_i, z_j) \quad \text{for all } i = 1, \ldots, n, \tag{2.18}$$

i.e. β_0 and $\boldsymbol{\beta}$ fulfill

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \mathrm{Cov}(z_1,z_1) & \cdots & \mathrm{Cov}(z_1,z_n) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \mathrm{Cov}(z_n,z_1) & \cdots & \mathrm{Cov}(z_n,z_n) \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} = \begin{bmatrix} \mathbb{E}[w] \\ \mathrm{Cov}(z_1,w) \\ \vdots \\ \mathrm{Cov}(z_n,w) \end{bmatrix}$$

$$\begin{bmatrix} 1 & \mathbf{0}^T \\ 0 & \mathbf{P} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbb{E}[w] \\ \mathrm{Cov}(\boldsymbol{z}, w) \end{bmatrix}. \tag{2.19}$$

Separating this, it holds that

$$\beta_0 = \mathbb{E}[w] \quad \text{and} \quad \mathbf{P}\boldsymbol{\beta} = \mathrm{Cov}(\boldsymbol{z}, w). \tag{2.20}$$

Consequently, by using the *Robertson-Price Identity*, which states that the selection differential

s is equal to the covariance between the trait $\boldsymbol{z}$ and its fitness w,

$$\mathbf{P}\boldsymbol{\beta} = s. \tag{2.21}$$

3. Reproducing Kernel Hilbert Spaces

3.1. Reproducing Kernel Hilbert Spaces

The theory of *reproducing kernel Hilbert spaces (RKHS)* appears in many fields of Mathematics, for example in complex analysis, harmonic analysis and quantum mechanics. Reproducing kernels and the *reproducing property* were first introduced by Zaremba[1] in 1907 in his work on harmonic functions, and later Bergman[2] discovered the reproducing property of kernels built by orthogonal systems of harmonic and analytic functions. Bergman and Aronszjn[3] eventually systematically developed this subject.

Especially in the recent past, reproducing kernel Hilbert spaces have become more and more important in Probability and Mathematical Statistics. In this chapter, the theory of reproducing kernel Hilbert spaces is introduced, as they are closely related to stochastic processes. In particular, the Gaussian dichotomy theorem is stated and its importance in the estimation of the selection gradient portrayed. A more detailed look on this matter is presented in Berlinet and Thomas-Agnan's book on RKHS [3].

Definition 3.1.(Hilbert Space) A real or complex vector space $\mathcal{H}$ is called *Hilbert space*

[1]Stanislaw Zaremba, 1863 -1942, Polish mathematician and engineer.
[2]Stefan Bergman, 1895 - 1977, Polish American mathematician.
[3]Nachman Aronszjn, 1907 - 1980, Polish American mathematician.

defined by the *inner product*

$$\begin{aligned} \langle \cdot, \cdot \rangle_{\mathcal{H}} : \mathcal{H} \times \mathcal{H} &\longrightarrow \mathbb{R} \quad \text{or} \quad \mathbb{C} \\ (\varphi, \psi) &\longmapsto \langle \varphi, \psi \rangle_{\mathcal{H}} \ , \end{aligned} \tag{3.1}$$

if it is complete with respect to the norm induced by the inner product

$$\|\varphi\|_{\mathcal{H}} = \sqrt{\langle \varphi, \varphi \rangle_{\mathcal{H}}} \ , \tag{3.2}$$

i.e. every Cauchy sequence in $\mathcal{H}$ converges.

Definition 3.2.(Reproducing Kernel) For an arbitrary set T and a Hilbert space $\mathcal{H}$ of functions on T, the function K on $T \times T$,

$$\begin{aligned} K : T \times T &\longrightarrow \mathbb{R} \quad \text{or} \quad \mathbb{C} \\ (s, t) &\longmapsto K(s, t) \end{aligned} \tag{3.3}$$

is called *reproducing kernel* of the Hilbert space $\mathcal{H}$ if and only if it satisfies

(i)

$$K_t = K(\cdot, t) \in \mathcal{H} \quad \text{for all } t \in T, \tag{3.4}$$

(ii)

$$\langle f, K_t \rangle_{\mathcal{H}} = f(t) \quad \text{for all } f \in \mathcal{H} \text{ and } t \in T. \tag{3.5}$$

The Hilbert space $\mathcal{H}$ is then called *reproducing kernel Hilbert space*, which is often also denoted by $\mathcal{H}(K, T)$ or $\mathcal{H}(K)$.

Remark 3.3. Equation (3.5) defines the so-called *reproducing property*, since any function $f \in \mathcal{H}$ can be evaluated, or reproduced, at a point $t \in T$ by the inner product of f with K_t.

Furthermore, from properties (i) and (ii)

$$K(s,t) = \langle K(\cdot,t), K_s\rangle_{\mathcal{H}} = \langle K_t, K_s\rangle_{\mathcal{H}} \tag{3.6}$$

follows immediately.

Example 3.4.([3] Example 1, pp. 4-5, 7-8) Let $\mathcal{H}$ be a finite dimensional complex vector space of functions on T with basis $(f_1, \ldots, f_n)$. Since this is a basis of $\mathcal{H}$ it is possible to represent any vector in $\mathcal{H}$ as a linear combination of the basis vectors $f_1, \ldots, f_n$. Consider vectors $v, u \in \mathcal{H}$. Then

$$v = \sum_{i=1}^{n} \lambda_i f_i, \qquad u = \sum_{j=1}^{n} \mu_j f_j$$

where the $\lambda_i, \mu_j \in \mathbb{C}, i, j = 1, \ldots, n$ are complex coefficients. The inner product $\langle v, u\rangle_{\mathcal{H}}$ is completely determined by values

$$g_{ij} = \langle f_i, f_j\rangle_{\mathcal{H}} \quad \text{for } i, j = 1, \ldots, n \tag{3.7}$$

since

$$\langle v, u\rangle_{\mathcal{H}} = \langle \sum_{i=1}^{n} \lambda_i f_i, \sum_{j=1}^{n} \mu_j f_j\rangle_{\mathcal{H}} = \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \overline{\mu}_j \langle f_i, f_j\rangle_{\mathcal{H}} \overset{(3.7)}{=} \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \overline{\mu}_j g_{ij}. \tag{3.8}$$

Note that the matrix $\boldsymbol{G} = [g_{ij}]_{i,j=1,\ldots,n}$ is called Gram matrix of the basis.

Since $\mathcal{H}$ is finite dimensional and equipped with the norm $\|\varphi\|_{\mathcal{H}} = \sqrt{\langle \varphi, \varphi\rangle_{\mathcal{H}}}$, it is complete with respect to that norm, and $\mathcal{H}$ is a Hilbert space.

Further developing our example of $\mathcal{H}$ to a finite dimensional Hilbert space of functions on T, consider an orthonormal basis $(e_1, \ldots, e_n)$ in $\mathcal{H}$. Define a function K on $T \times T$ by

$$K(s,t) = \sum_{i=1}^{n} e_i(s)\overline{e}_i(t) \quad \text{for all } s, t \in T. \tag{3.9}$$

Then for any $t \in T$

$$K(\cdot, t) = \sum_{i=1}^{n} e_i(\cdot)\overline{e}_i(t) \tag{3.10}$$

is a function in $\mathcal{H}$. Additionally, for any function $\varphi \in \mathcal{H}$,

$$\varphi(\cdot) = \sum_{i=1}^{n} \lambda_i e_i(\cdot), \tag{3.11}$$

the following holds:

$$\begin{aligned}\langle \varphi(\cdot), K_t \rangle &= \Big\langle \sum_{i=1}^{n} \lambda_i e_i(\cdot), \sum_{j=1}^{n} \overline{e}_j(t) e_j(\cdot) \Big\rangle = \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \overline{\overline{e}}_i(t) \underbrace{\langle e_i(\cdot), e_j(\cdot)\rangle}_{=\begin{cases}1, & \text{if } i=j\\ 0, & \text{otherwise.}\end{cases}} \\ &= \sum_{i=1}^{n} \lambda_i e_i(t) = \varphi(t).\end{aligned} \tag{3.12}$$

Thus, both properties (i) and (ii) of Definition 3.2 are satisfied and K is a reproducing kernel, making $\mathcal{H}$ a reproducing kernel Hilbert space.

Example 3.5.(Cont'd; [3] Theorem 14, p. 32) An extension of Example 3.4 is given in [3] Theorem 14. Let $\mathcal{H}$ be a separable Hilbert space of complex-valued functions on T, i.e. $\mathcal{H}$ has countable dimension. Let K be the reproducing kernel. Then for any complete orthonormal system $(e_i)_{i\in\mathbb{N}}$

$$K(s,t) = \sum_{i=1}^{\infty} e_i(s)\overline{e}_i(t) \quad \text{for all } s,t \in T, \tag{3.13}$$

and

$$K(\cdot, t) = \sum_{i=1}^{\infty} e_i(\cdot)\overline{e}_i(t) \quad \text{for all } t \in T. \tag{3.14}$$

Conversely if (3.14) holds for orthonormal system $(e_i)_{i\in\mathbb{N}}$ then this system is complete and $\mathcal{H}$ is separable. The proof of this theorem is stated in [3].

Even though all results of this section are given for complex functions, henceforth we will only consider $\mathcal{H}$ to be Hilbert space of real-valued functions on an arbitrary domain T and denote by $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ the associated inner product.

3.2. Stochastic Processes

Before stating the Gaussian Dichotomy Theorem, which plays a central role in estimating the selection gradient of function-valued traits, it is important to define the necessary foundation on which this theory is constructed. In this section, we will define stochastic processes, in particular Gaussian processes, and establish how they are connected to reproducing kernel Hilbert spaces.

Definition 3.6.(Stochastic Process) A *stochastic process* $\{Z_t : t \in T\}$ on the probability space $(\Omega, \mathcal{A}, \mathrm{P})$ is a family of real-valued random variables on Ω. We say, the stochastic process $\{Z_t : t \in T\}$ is of *p-th order* if

$$\mathbb{E}_{\mathrm{P}}|Z_t|^p < \infty \quad \text{for all } t \in T, \tag{3.15}$$

where the expected value with respect to the probability measure P is defined by

$$\mathbb{E}_{\mathrm{P}}[\cdot] = \int_{\Omega} \cdot \, \mathrm{dP}. \tag{3.16}$$

Remark 3.7. A stochastic process $\{Z_t : t \in T\}$ is called *Gaussian process* if and only if for any finite set of indices $t_1, \ldots, t_n \in T$, the random vector $(Z_{t_1}, \ldots, Z_{t_n})$ is multivariate Gaussian, i.e. any finite linear combination of the random variables $Z_{t_1}, \ldots, Z_{t_n}$ has a normal distribution. The process $\{Z_t : t \in T\}$ is additionally called a *zero mean Gaussian process* if $\mathbb{E}[Z_t] = 0$ for all $t \in T$.

Definition 3.8.(Covariance Kernel) Let T be an arbitrary set and K be a real-valued function on $T \times T$. The function K is called *covariance kernel* if it is

(i) Symmetric:

$$K(s,t) = K(t,s) \quad \text{for all } s,t \in T \tag{3.17}$$

(ii) Positive definite: For all finite sets of indices $t_1, \ldots, t_n \in T$ and for all vectors $\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{x} =$

$[x_1, \ldots, x_n]$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j K(t_i, t_j) \geq 0 \quad (= 0 \Leftrightarrow \boldsymbol{x} = \mathbf{0}). \tag{3.18}$$

Remark 3.9. For a stochastic process of *second order* $\{Z_t : t \in T\}$, the mean function $\mu(\cdot)$ and the covariance function $K(\cdot, \cdot)$ exist and are defined by

$$\mu(t) = \mathbb{E}_{\mathrm{P}}[Z_t] \tag{3.19}$$

$$K(s,t) = Cov(Z_s, Z_t) = \mathbb{E}_{\mathrm{P}}[(Z_s - \mu(s))(Z_t - \mu(t))]. \tag{3.20}$$

Let K be the covariance function of a second order stochastic process $\{Z_t : t \in T\}$. Then K fulfills both conditions that are defining a covariance kernel. K is symmetric and positive definite,

$$Cov(Z_s, Z_t) = Cov(Z_t, Z_s) \quad \text{for all } s, t \in T, \tag{3.21}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j Cov(Z_{t_i}, Z_{t_j}) = Var\left(\sum_{i=1}^{n} a_i Z_{t_i}\right) \geq 0, \tag{3.22}$$

and it follows that K is a covariance kernel. Furthermore, any second order stochastic process is contained in the function space $L^2(\Omega, \mathcal{A}, \mathrm{P})$.

Definition 3.10.(Gaussian Space) Define the space of stochastic processes spanned by $\{Z_t : t \in T\}$ by

$$V = \left\{ X \in L^2(\Omega, \mathcal{A}, \mathrm{P}) : X = \sum_{i=1}^{n} \lambda_i Z_{t_i}, n \in \mathbb{N}, \lambda_1, \ldots, \lambda_n \in \mathbb{R}, t_1, \ldots, t_n \in T \right\}. \tag{3.23}$$

Clearly V is contained in $L^2(\Omega, \mathcal{A}, \mathrm{P})$ and the closure of V, denoted by $H = \overline{V} \in L^2(\Omega, \mathcal{A}, \mathrm{P})$, is called the *Hilbert space spanned by the process* $\{Z_t : t \in T\}$. H is the smallest Hilbert space in $L^2(\Omega, \mathcal{A}, \mathrm{P})$ that contains $\{Z_t : t \in T\}$. If the stochastic process $\{Z_t : t \in T\}$ is Gaussian, then H is called the *Gaussian space associated to* $\{Z_t : t \in T\}$. Furthermore denote by $\mathcal{B}(H) = \overline{\mathcal{B}(Z_t : t \in T)}$ the σ-algebra generated by the process $\{Z_t : t \in T\}$ and P-negligible sets.

A key property of second order stochastic processes and reproducing kernel Hilbert spaces is stated in the following.

Proposition 3.11.([16] Proposition 3.2, [1] Proposition 2.1.1) Let $\{Z_t : t \in T\}$ be a zero mean Gaussian process on the probability space $(\Omega, \mathcal{A}, \mathrm{P})$ with covariance function K and let H be the Gaussian space associated to the process. There exists a unique reproducing kernel Hilbert space with kernel K consisting of functions defined on T. This reproducing kernel Hilbert space $\mathcal{H}(K)$ and the Gaussian space H are isomorphic via the isometry $\Lambda : H \to \mathcal{H}(K)$ defined by

$$\Lambda(Y)[t] = \mathbb{E}_{\mathrm{P}}[YZ_t] \quad \text{for all } Y \in H. \tag{3.24}$$

This map is called the *Loève map*. Note that the Loève map has the key property

$$\Lambda(Z_s) = K_s, \tag{3.25}$$

where $K_s = K(\cdot, s)$ is a section of the covariance kernel K at $s \in T$. This is true not only for zero mean Gaussian processes, but also for any second order stochastic process.

3.3. The Gaussian Dichotomy Theorem

The Gaussian Dichotomy Theorem plays a central role in estimating the mean and covariance function of Gaussian processes.

Definition 3.12.(Singular and equivalent measures, [18] Definition 6.7) Let μ and ν be measures on the same measurable space $(\Omega, \mathcal{A})$.

We say μ is *absolutely continuous with respect to* ν, $\mu \ll \nu$, if

$$\nu(N) = 0 \Longrightarrow \mu(N) = 0, \tag{3.26}$$

that is, every *nullset* $N \in \mathcal{A}$ of ν is a nullset of μ as well. The measures μ and ν are called

equivalent if and only if they are absolute continuous with respect to each other:

$$\mu(N) = 0 \iff \nu(N) = 0, \tag{3.27}$$

in other words, μ and ν have the same nullsets. The equivalence of two measures is denoted $\mu \approx \nu$.

Let $A \in \mathcal{A}$ be a set. The measure μ is *concentrated on* A if

$$\mu(E) = \mu(A \cap E) \quad \text{for all } E \in \mathcal{A}, \tag{3.28}$$

equivalently

$$A \cap E = \emptyset \implies \mu(E) = 0. \tag{3.29}$$

Two measures μ and ν are *(mutually) singular* if there exists a pair of disjoints subsets $A, B \subset \Omega$ such that μ is concentrated on A and ν is concentrated on B. We write $\mu \perp \nu$.

Theorem 3.13.(The Gaussian Dichotomy Theorem Part 1, [16] Proposition 8.1, [1] Theorem 2.2.1) Let $\{Z_t : t \in T\}$ be a Gaussian process defined on $(\Omega, \mathcal{A}, \mathrm{P})$ with mean zero and covariance function K. Let H be the Gaussian space associated to the process. Furthermore, assume $\mathcal{A} = \mathcal{B}(H)$.

Then any probability measure Q on $(\Omega, \mathcal{A})$ such that $\{Z_t : t \in T\}$ is a Gaussian process with covariance function K, is either singular or equivalent to P. For Q and P to be equivalent, it is necessary and sufficient that there exists a random variable $Y \in H$ such that

$$\mathbb{E}_Q[Z_t] = \mathbb{E}_{\mathrm{P}}[Y Z_t] \quad \text{for all } t \in T. \tag{3.30}$$

Equivalently, it is necessary and sufficient that the mean function $\mu_Q(\cdot) = \mathbb{E}_Q[Z_\cdot]$ of the process $\{Z_t : t \in T\}$ belongs to the reproducing kernel Hilbert space $\mathcal{H}(K)$.

If these conditions are fulfilled, the Radon-Nikodym derivative of Q with respect to P on

$(\Omega, \mathcal{A}, \mathrm{P})$ is given by

$$\frac{\mathrm{d}Q}{\mathrm{dP}} = \exp\{Y - \frac{1}{2}\mathbb{E}_{\mathrm{P}}\left[Y^2\right]\}. \tag{3.31}$$

Conversely, for any $Y \in H$, the process $\{Z_t : t \in T\}$ is Gaussian with respect to the probability measure Q on $(\Omega, \mathcal{A})$ defined by Y, i.e.

$$\mathrm{d}Q = \exp\{Y - \frac{1}{2}\mathbb{E}_{\mathrm{P}}\left[Y^2\right]\}\mathrm{dP} \tag{3.32}$$

and $Q \approx P$. The mean function of $\{Z_t : t \in T\}$ is then given by $\mu_Q(\cdot) = \mathbb{E}_Q\left[Z_\cdot\right] = \mathbb{E}_{\mathrm{P}}\left[YZ_\cdot\right]$ and the covariance function is K.

As the full second part of the Gaussian Dichotomy Theorem does not play a major role in the estimations of interest, only a shortened version will be stated. The full version can be found in the sources cited.

Theorem 3.14.(The Gaussian Dichotomy Theorem Part 2, [16] Proposition 8.6, [1] Theorem 2.2.1) Let $\{Z_t : t \in T\}$ be a zero mean Gaussian process on the probability space $(\Omega, \mathcal{A}, \mathrm{P})$ where $\mathcal{A} = \mathcal{B}(H)$. Denote by K_P and H the covariance function as well as the Gaussian space associated to this process.

Let Q be a another probability measure on the same space which endows $\{Z_t : t \in T\}$ with a zero mean Gaussian distribution and let K_Q be the covariance function with respect to Q. Then the probability measures P and Q are either *singular* or *equivalent*. For Q and P to be equivalent, it is necessary and sufficient that

$$K_Q(s,t) - K_{\mathrm{P}}(s,t) = \sum_k a_k g_k(s) g_k(t) \quad \text{for all } s,t \in T. \tag{3.33}$$

where $\{g_k\}$ is an orthonormal system in the reproducing kernel Hilbert space $\mathcal{H}(K_{\mathrm{P}})$

3.4. Sieve Estimation

In his thesis [1], Baur illustrates how the Gaussian Dichotomy Theorem is used to estimate covariance functions and the importance of *sieve estimators*. At this point, only the results shall be mentioned.

Let $\{Z_t : t \in T\}$ be a stochastic process on the measure space $(\Omega, \mathcal{A})$. Consider the space $\mathscr{Q}$ of probability measures such that

(1) the process $\{Z_t : t \in T\}$ is a zero mean Gaussian process under every probability measure $Q \in \mathscr{Q}$,

(2) all probability measures $Q \in \mathscr{Q}$ are equivalent,

(3) the true probability measure belongs to $\mathscr{Q}$

Under these conditions, for any arbitrary $\mathrm{P} \in \mathscr{Q}$, $\{Z_t : t \in T\}$ is a zero mean Gaussian process with covariance function K_{P} and associated Gaussian space H_{P}. Denote $\mathcal{H}_{\mathrm{P}}$ the reproducing kernel Hilbert space with kernel K_{P} and $\Lambda_{\mathrm{P}} : H_{\mathrm{P}} \longrightarrow \mathcal{H}_{\mathrm{P}}$ the associated Loève map.

By the Gaussian Dichotomy theorem, it follows for each $Q \in \mathscr{Q}$ that there exists a countable orthonormal sequence $\{g_k\}$ in the RKHS $\mathcal{H}_{\mathrm{P}}$ and a sequence $\{a_k\}$, with $\sum_k a_k^2 < \infty$ and $\inf_k a_k > -1$, such that

$$K_Q = K_{\mathrm{P}} + \sum_k a_k g_k \otimes g_k, \tag{3.34}$$

where the operation $f \otimes g$ defines a function on $T \times T$ with $(f \otimes g)(s,t) = f(s)g(t)$. Thus, equation (3.34) essentially has the form

$$\begin{aligned} K_Q(s,t) &= K_{\mathrm{P}}(s,t) + \sum_k a_k g_k \otimes g_k(s,t) \\ &= K_{\mathrm{P}}(s,t) + \sum_k a_k g_k(s) g_k(t). \end{aligned} \tag{3.35}$$

The sequences $\{g_k\}$ and $\{a_k\}$ both depend on the measure $Q \in \mathscr{Q}$. Following equation (3.34),

define the set of covariance functions

$$\mathcal{K} = \Big\{ K_Q = K_{\mathrm{P}} + \sum_k a_k g_k \otimes g_k : \{g_k\} \subset \mathcal{H}_{\mathrm{P}} \text{ countable and orthonormal,} \\ \{a_k\} \text{ with } \sum_k a_k^2 < \infty \text{ and } \inf_k a_k > -1 \Big\}. \tag{3.36}$$

Furthermore, there is a countable orthonormal sequence $\{U_k\}$ in the Gaussian space H_{P} and a sequence $\{\lambda_k\}$, with $\{-\lambda_k\}$ fulfilling the same conditions as $\{a_k\}$, such that the Radon-Nikodym density of Q with respect to P is given by

$$\frac{\mathrm{d}Q}{\mathrm{dP}} = \exp\Big\{ \frac{1}{2} \sum_k \big(\lambda_k U_k^2 + ln(1-\lambda_k) \big) \Big\}, \tag{3.37}$$

where $(1-\lambda_k)(1+a_k) = 1$ and the random variables U_k are the solution of $g_k = \Lambda_{\mathrm{P}} U_k$.

Definition 3.15.(Sieve Estimator) Let $\mathscr{Q} = \{P_\theta : \theta \in \Theta\}$ be a *dominated* family of distributions, that means, every measure P_θ is absolute continuous with respect to some measure μ. A *sieve* in the parameter space Θ is a collection $\{\mathcal{S}_d\}$ of subsets of Θ indexed by a so-called *sieve parameter* d with the properties

(i) $d' > d$ implies that $\mathcal{S}_{d'} \supset \mathcal{S}_d$,

(ii) the union over all subsets $\mathcal{S}_d$ is dense in Θ,

(iii) the likelihood can be maximized at $\hat{\theta}_d$ for each $\mathcal{S}_d$ for some sample of size n.

The estimator $\hat{\theta}_d$ over each $\mathcal{S}_d$ is called *sieve estimator* of θ with sieve parameter d.

Finding a sieve in $\mathcal{K}$ is rather difficult, as both the sequence of orthonormal functions $\{g_k\}$ as well as the sequence $\{a_k\}$ are variable, making $\mathcal{K}$ very large. Therefore, by fixing a complete orthonormal sequence $\{g_k\}$, a subset $\mathcal{K}_0 \subset \mathcal{K}$ is considered,

$$\mathcal{K}_0 = \Big\{ K_Q = K_{\mathrm{P}} + \sum_k a_k g_k \otimes g_k : \{a_k\} \text{ with } \sum_k a_k^2 < \infty \text{ and } \inf_k a_k > -1 \Big\}. \tag{3.38}$$

Finding a sieve estimator in $\mathcal{K}_0$ requires finding a sieve estimator $\hat{\boldsymbol{a}}$ in

$$\ell_c^2 = \Big\{ \boldsymbol{a} = \{a_k\} : \sum_k a_k^2 < \infty \text{ and } \inf_k a_k > -1 \Big\}. \tag{3.39}$$

The collection $\{\mathcal{S}_d : d \in \mathbb{N}\}$ with

$$\mathcal{S}_d = \{\boldsymbol{a} = \{a_k\} \in \ell_c^2 : a_k = 0 \text{ for all } k > d\} \tag{3.40}$$

is a sieve in ℓ_c^2. The sieve estimator $\hat{\boldsymbol{a}} = \hat{\boldsymbol{a}}_{n,d} = \{a_k\}_{k\in\mathbb{N}}$ is given by

$$\hat{a}_k = \begin{cases} S_k^2 - 1 & , \text{ for } k \le d \\ 0 & , \text{ otherwise} \end{cases}, \tag{3.41}$$

where $S_k^2 = \frac{1}{n}\sum_{i=1}^{n} U_{ki}^2$. As a result, the sieve estimator for K_Q in $\mathcal{K}_0$ has the form

$$\widehat{K_Q} = K_{\mathrm{P}} + \sum_k \hat{a}_k g_k \otimes g_k. \tag{3.42}$$

4. Infinite-dimensional Traits

4.1. The Quantitative Genetic Model for Infinite-dimensional Traits

Analogous to chapter 2, the selection gradient will be defined for function-valued traits. Notations follow the conventions specified in Remark 2.1 for most part. A function-valued trait is a *stochastic process* $\{z(t) : t \in T\}$, often also $\{z_t : t \in T\}$, where T is for example a set of time, temperatures, or other environmental variables. For ease of notation, function-valued traits are often simply denoted $z(t)$. The phenotype of an individual is a realization of $\{z(t) : t \in T\}$, i.e. a *sample path*. The *pre-selection mean* is denoted $\bar{z}(t)$. The *post-selection mean* is given by $\bar{z}^*(t)$ and $\bar{z}'(t)$ is the mean function of the trait among newborns of the next generation. The distribution before selection, after selection and the distribution of the offspring generation are $\mathrm{P}_{\bar{z}}$, $\mathrm{P}_{\bar{z}^*}$ and $\mathrm{P}_{\bar{z}'}$ respectively.

Similar as for finite-dimensional traits, the infinite-dimensional model decomposes a function-valued trait $z(t)$ into a sum

$$z(t) = g(t) + e(t) \ , \tag{4.1}$$

where $\{g(t) : t \in T\}$ is a stochastic process describing the *additive-genetic factor* in the trait, i.e. inheritance from the parents, and the process $\{e(t) : t \in T\}$ represents *environmental effects*. It is assumed that there is no correlation between environmental and genetic factors, and therefore the *phenotypic covariance function* $P(s,t)$ for $s,t \in T$ breaks down as

$$P(s,t) = G(s,t) + E(s,t) \ , \tag{4.2}$$

where $G(s,t)$ is the *additive-genetic covariance function* and $E(s,t)$ is the *environmental covariance function*.

The *fitness* $W(z)$, or just W, of a function-valued trait is assumed to be a function that satisfies the following properties:

(i) W is a positive function

(ii) $\mathbb{E}_{\bar{z}}[W] < \infty$ and $\mathbb{E}_{\bar{z}}[z(t)W] < \infty$ for all $t \in T$.

The fitness might depend on $\bar{z}(t)$ or other parameters of the pre-selection distribution $\mathrm{P}_{\bar{z}}$. If W is independent of those, we say W is *frequency-independent*. The *relative fitness*

$$w = \frac{W}{\mathbb{E}_{\bar{z}}[W]} \tag{4.3}$$

defines the *post-selection distribution* by

$$\mathrm{dP}_{\bar{z}^*} = w\mathrm{dP}_{\bar{z}} = \frac{W}{\mathbb{E}_{\bar{z}}[W]}\mathrm{dP}_{\bar{z}}. \tag{4.4}$$

Note that $\mathrm{P}_{\bar{z}^*}$ is a probability measure and absolutely continuous with respect to the pre-selection distribution $\mathrm{P}_{\bar{z}}$.

Similar as for vector-valued traits, the *selection differential* s is defined as a function on T,

$$s(t) = \bar{z}^*(t) - \bar{z}(t)\ , \tag{4.5}$$

and represents the change in the mean function of the trait within a generation, i.e. the change between pre-selection mean and post-selection mean. The *evolutionary response to selection* $\Delta\bar{z}$, that is the change of the mean among newborns of successive generations, is

$$\Delta\bar{z}(t) = \bar{z}'(t) - \bar{z}(t). \tag{4.6}$$

In the following it is assumed that

(i) the trait $z(t)$ is Gaussian with mean $\bar{z}(t)$ and phenotypic covariance function $P(s,t)$, $s,t \in T$. In particular, $\bar{z}$ belongs to the reproducing kernel Hilbert space $\mathcal{H}(P,T)$.

(ii) The fitness $W > 0$ almost surely with respect to the probability measure $\mathrm{P}_{\bar{z}}$ and belongs to $L^2(\Omega, \mathcal{A}, \mathrm{P}_{\bar{z}})$ for all $\bar{z} \in \mathcal{H}(P,T)$. That is,

$$Var_{\bar{z}}(W) < \infty \quad \text{for all } \bar{z} \in \mathcal{H}(P,T). \tag{4.7}$$

Since it is hard to gather information on the distribution or the newborns of the next generation, in particular on $\bar{z}'$, no inference can be made on the evolutionary response to selection, $\Delta\bar{z}$. Kirkpatrick and Heckmann state that the *Breeder's Equation* is applicable for infinite-dimensional traits, thus

$$\Delta\bar{z}(t) = \mathcal{G}\mathcal{P}^{-1}s(t), \tag{4.8}$$

with $\mathcal{G}$ and $\mathcal{P}$ being *integral operators* with kernel $G(s,t)$ and $P(s,t)$, respectively. Integral operators will be defined in the next subsection on restrictions on T and the covariance functions. Lastly, the *selection gradient* is then defined by

$$\beta(t) = \mathcal{P}^{-1}s(t). \tag{4.9}$$

The Breeder's equation is dependent on the selection differential, which is in general hard to compute. Information on the post-selection mean $\bar{z}^*$ is not observable in the field, except for in artificially controlled breeding environments. The Robertson-Price Identity, stated later, will be a useful tool to overcome this obstacle.

Remark 4.1. The selection gradient β given in equation (4.9) is only defined if the selection differential s belongs to the range of $\mathcal{P}$. To give a proper definition of the selection gradient for the case $s \notin \mathcal{P}$, an extension $\bar{\mathcal{P}}$ of the integral operator $\mathcal{P}$ can be found. Technically this means, the selection gradient β is the solution of the equation

$$\bar{\mathcal{P}}\beta = s \tag{4.10}$$

(see [1] Section 3.2.2).

Similar to Lande's Theorem for finite-dimensional traits, see Theorem 2.2, it is possible to define the selection gradient β as the *functional gradient* of $\log(\mathbb{E}_{\bar{z}}W)$ at $\bar{z}$. A detailed view on this can be found in [9], Section 4.3.

4.2. Restrictions on T and the Covariance Functions

Beder and Gomulkiewicz [9], Section 3.3, give restrictions on the set T and the covariance function P.

(i) The set T is assumed to belong to a measure space with σ-algebra $\mathcal{T}$ and measure μ. Furthermore, μ is a σ-finite measure, i.e. every $A \in \mathcal{A}$ is the countable union of sets $A_n \in \mathcal{A}, n \in \mathbb{N}$, with finite measure. This means for all $A \in \mathcal{A}$ there exists a sequence $(A_n)_{n\in\mathbb{N}} \subset \mathcal{A}$ with $\mu(A_n) < \infty$ for all $n \in \mathbb{N}$, such that $A = \bigcup_{n\in\mathbb{N}} A_n$.

(ii) P is a measurable covariance kernel on $T \times T$ with finite trace, i.e.

$$\int_T P(t,t)\mathrm{d}\mu(t) < \infty. \tag{4.11}$$

(iii) The zero function on T is the only μ-negligible function in the reproducing kernel Hilbert space $\mathcal{H}(P,T)$.

A simple example of T and P that satisfy assumptions (i), (ii) and (iii) is given by any interval $T = [a,b]$ on the real line and continuous function P. Further results following the conditions above can be found in [9], Section 3.3.

Definition 4.2.(Integral Operator) Let f be a square-integrable function on T. The *integral operator* $\mathcal{P}$ is defined by

$$\mathcal{P}f(s) = \int_T P(s,t)f(t)\mathrm{d}\mu(t). \tag{4.12}$$

$P(s,t)$ is called the *kernel* of the integral operator $\mathcal{P}$.

Proposition 4.3.(Robertson-Price Identity for Function-valued Traits) Under the assumptions made above, the selection differential $s(t)$ is given by

$$s(t) = \mathrm{Cov}_{\bar{z}}(z(t), w). \tag{4.13}$$

Proof. Let all expected values be computed with respect to $\mathrm{P}_{\bar{z}}$. It holds

$$\mathrm{Cov}(z(t), w) = \mathbb{E}[z(t)w] - \mathbb{E}[z(t)]\mathbb{E}[w]. \tag{4.14}$$

Since the relative fitness w is defined by $w = \frac{W}{\mathbb{E}[W]}$, its expected value is $\mathbb{E}[w] = \mathbb{E}\left[\frac{W}{\mathbb{E}[W]}\right] = 1$. Moreover, by convention, $\mathbb{E}[z(t)]$ is denoted by $\bar{z}(t)$.

Using the relationship between the pre-selection distribution $\mathrm{P}_{\bar{z}}$ and post-selection distribution $\mathrm{P}_{\bar{z}^*}$ given by equation (4.4), compute

$$\mathbb{E}[z(t)w] = \int z(t)w\mathrm{dP}_{\bar{z}} = \int z(t)\mathrm{dP}_{\bar{z}^*} = \bar{z}^*(t). \tag{4.15}$$

Hence, the covariance of $z(t)$ and w is

$$\mathrm{Cov}(z(t), w) = \bar{z}^*(t) - \bar{z}(t) = s(t). \tag{4.16}$$

□

Proposition 4.4.([2] Proposition 2.2, [1] Proposition 3.2.2) Let P_0 and P be equivalent probability measures with covariance kernels P_0 and P, and corresponding integral operators $\mathcal{P}_\prime, \mathcal{P}$, respectively. Let $\{g_k\}$ be an orthonormal sequence in the reproducing kernel Hilbert space $\mathcal{H}(P_0)$ and let $\{\gamma_k\}$ be a sequence of functions such that $\mathcal{P}_0\gamma_k = g_k$ for all k. Then

$$(\gamma_k, g_k) = \delta_{jk} = \begin{cases} 1 & , \text{ if } j = k, \\ 0 & , \text{ if } j \neq k \end{cases}, \text{ (Kronecker } \delta), \tag{4.17}$$

where $(\cdot, \cdot)$ is the inner product in $L^2(T)$, i.e. $(f, g) = \int_T f(t)g(t)\mathrm{d}t$.

Furthermore, if the covariance function P and the selection gradient β are of the form

$$P = P_0 + \sum_k a_k g_k \otimes g_k \tag{4.18}$$

$$\beta = \sum_k b_k \gamma_k, \tag{4.19}$$

as well as, if the selection differential has the expansion $s = \sum_k c_k g_k$, then

$$\bar{\mathcal{P}}\beta = s \tag{4.20}$$

if and only if the coefficients a_k, b_k, c_k satisfy

$$b_k = \frac{c_k}{a_k + 1}. \tag{4.21}$$

5. Estimating the Selection Gradient β for Infinite-dimensional Traits

This chapter deals with the actual estimation of the selection gradient β for function-traits, in particular estimators for the phenotypic covariance function P, the mean functions among the newborns of the parent generation $\bar{z}$ and the offspring generation $\bar{z}'$, respectively, as well as all necessary quantities of the estimation will be defined. A rigorous derivation is disclosed in [1], so that, here, this thesis will only focus on its results.

5.1. Independent Case

Assume that all observed individuals are bred independently under identical conditions. Let $(\Omega, \mathcal{A})$ be a measurable space and $\{z(t) : t \in T\}$ a stochastic process on $(\Omega, \mathcal{A})$. Denote by $\mathrm{P}_{\bar{z}}$ the distribution which endows $\{z(t) : t \in T\}$ with a Gaussian distribution with mean $\bar{z}$ and covariance function $P_{\bar{z}}$. Likewise P and P_0 endow the process with a zero mean Gaussian distribution and covariance functions P and P_0, respectively.

As specified in the previous chapter, the selection gradient β is the solution of equation (4.10), i.e.

$$\bar{\mathcal{P}}\beta = s \tag{5.1}$$

where $\bar{\mathcal{P}}$ is the extension of the integral operator $\mathcal{P}$ whose kernel is the covariance function P. For ease of notation, this extension will simply be denoted $\mathcal{P}$.

Let $\{g_k\}$ be a complete orthonormal sequence in $\mathcal{H}(P_0)$ and let $\{\gamma_k\}$ be a sequence satisfying $\mathcal{P}_0\gamma_k = g_k$. Furthermore, it is assumed, that the covariance function P, the selection gradient β and the selection differential s satisfy the prerequisites of Proposition 4.4, that is to say

$$P = P_0 + \sum_k a_k g_k \otimes g_k, \tag{5.2}$$

$$\beta = \sum_k b_k \gamma_k, \tag{5.3}$$

$$s = \sum_k c_k g_k. \tag{5.4}$$

Then, by the same proposition, estimates $\hat{P}$ and $\hat{s}$ are determined by the estimation of the coefficients a_k and c_k, denoted $\hat{a}_k$ and $\hat{c}_k$, from which it is possible to estimate

$$\hat{b}_k = \frac{\hat{c}_k}{\hat{a}_k + 1}. \tag{5.5}$$

5.1.1. Estimating the Phenotypic Covariance Function P

The estimate for the phenotypic covariance function is given by

$$\hat{P} = P_0 + \sum_{k=1}^{d} \hat{a}_k g_k \otimes g_k. \tag{5.6}$$

The candidate covariance function P_0 is simply chosen before simulation. As it turns out later, the choice of this candidate significantly influences the resulting estimates. $\{g_k\}$ is an orthonormal sequence in $\mathcal{H}(P_0)$ achieved by *Gram-Schmidt orthonormalization* of the sections $P_{0t_i} = P_0(\cdot, t_i)$, i.e. every function g_k has the form

$$g_k = \sum_i q_{ki} P_{0t_i}, \tag{5.7}$$

where q_{ki} are the coefficients of the orthonormalization (inner products). The estimator for the coefficients a_k follows from the Gaussian Dichotomy theorem and restricted maximum likelihood

estimation,

$$\hat{a}_k = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}(U_k^{(i)})^2 - 1 & , \text{ if } k \leq d, \\ 0 & , \text{ otherwise} \end{cases}. \tag{5.8}$$

Here, the $\{U_k\}$ are a sequence of random variables in the Gaussian space H_{P_0}, such that $\Lambda_{P_0} U_k = g_k$. The computation of U_k is straightforward. Since the Loève map Λ_{P_0} is an isomorphism and g_k is constructed by Gram-Schmidt, U_k has the simple form

$$U_k = \Lambda_{\mathrm{P}_0}^{-1} g_k = \Lambda_{\mathrm{P}_0}^{-1} \sum_i q_{ki} P_{0t_i} = \sum_i q_{ki} \Lambda_{\mathrm{P}_0}^{-1} P_{0t_i} = \sum_i q_{ki} z(t_i). \tag{5.9}$$

5.1.2. Estimating the Selection Differential s

To determine the estimates $\hat{c}_k$, observations for the selection differential s are needed, which is in general not possible, except for individuals in artificially controlled breeding environments. Fortunately, the fitness function W of a trait is well observable, and a function-valued version of the Robertson-Price identity is given in Proposition 4.3, which enables the estimation of the coefficients c_k. Note that $\{z(t) : t \in T\}$ has an expansion ([2] p.7) of the form

$$z(t) = \sum_k U_k g_k(t), \tag{5.10}$$

where $\{U_k\}$ and $\{g_k\}$ are specified as above. By the Robertson-Price identity and the linearity of the covariance, it follows that

$$s(t) = \mathrm{Cov}(z(t), w) = \sum_k \mathrm{Cov}(U_k, w) g_k(t), \tag{5.11}$$

and for a sample of size n the estimate of c_k is

$$\hat{c}_k = \widehat{\mathrm{Cov}}(U_k, w) = \frac{1}{n}\sum_{i=1}^{n}(\hat{w}_i - \bar{w})(U_k^{(i)} - \bar{U}_k), \tag{5.12}$$

where

$$\hat{w}_i = \frac{W_i}{\frac{1}{n}\sum_{j=1}^{n} W_j}, \tag{5.13}$$

and $U_k^{(i)}$ is the i^{th} realization of U_k.

5.1.3. Estimating the Selection Gradient β

As a result, estimates $\hat{b}_k$ are computed according to equation (5.5) and thus the estimate for the selection gradient is

$$\hat{\beta} = \sum_{k=1}^{d} \hat{b}_k \gamma_k. \tag{5.14}$$

Recall that the functions γ_k satisfy $\mathcal{P}_0 \gamma_k = g_k$. In Sections 2.1 and 2.2 of [2] the explicit form of those functions, depending on the method used to derive the g_k, is given. Above, the g_k were constructed by Gram-Schmidt orthonormalization, see equation (5.7), that is

$$g_k = \sum_i q_{ki} P_{0t_i}. \tag{5.15}$$

By defining the γ_k by

$$\gamma_k = \sum_i q_{ki} \delta_{t_i}, \tag{5.16}$$

where δ_{t_i} is the *Dirac-δ* function, which has the property

$$\mathcal{P}_0 \delta_t = P_{0t}, \tag{5.17}$$

it follows that

$$\mathcal{P}_0 \gamma_k = \mathcal{P}_0 \sum_i q_{ki} \delta_{t_i} = \sum_i q_{ki} \mathcal{P}_0 \delta_{t_i} = \sum_i q_{ki} P_{0t_i} = g_k. \tag{5.18}$$

5.1.4. Estimating the Pre-selection Mean $\bar{z}$

The above estimators only hold if the Gaussian process $\{Z(t) : t \in T\}$ under the true probability measure P has mean zero. In general this is not the case, and new estimators for $\{a_k\}$, and

additionally for the mean function $\bar{z}$ and covariance function P, have to be defined. An in-depth derivation of those can be read in [1], Section 4.1.

Like Section 3.4 on sieve estimation, under the light of the Gaussian Dichotomy Theorem, consider the collection $\mathscr{P}$ of probability measures on the measure space $(\Omega, \mathcal{A})$ such that

1. the process $\{Z_t : t \in T\}$ is a zero mean Gaussian process under every probability measure $P \in \mathscr{P}$,
2. all probability measures $P \in \mathscr{P}$ are equivalent,
3. the mean function $\bar{z}$ belongs to the reproducing kernel Hilbert space for all $P \in \mathscr{P}$.

Recall the following notation. Let $\mathrm{P}_{\bar{z}}$ be the measure that endows $\{Z(t) : t \in T\}$ with a Gaussian distribution with mean $\bar{z}$ ($\bar{z} \neq 0$) and covariance function P. Furthermore, the spaces ℓ^2 and ℓ_c^2 are defined by

$$\ell^2 = \{\boldsymbol{a} = \{a_k\} : \sum_k a_k < \infty\} \tag{5.19}$$

$$\ell_c^2 = \{\boldsymbol{a} = \{a_k\} : \sum_k a_k < \infty, \inf_k a_k > -1\}. \tag{5.20}$$

We wish to find a joint estimator $(\hat{\bar{z}}, \hat{P})$ for the mean and covariance function of the function-valued trait $\{z(t) : t \in T\}$. Using the Gaussian Dichotomy Theorem and Proposition 3.11, the mean function $\bar{z}$ can be written in the following way. Denote by $\mathbb{E}_{\mathrm{P}}$ the expected value with respect to the distribution P. The same holds for any other operation. Recall, P is the probability measure that endows $\{z(t) : t \in T\}$ with a zero mean Gaussian distribution and covariance function $P(s,t)$.

$$\begin{aligned}\bar{z}(t) = \mathbb{E}_{\mathrm{P}_{\bar{z}}}[Z(t)] &\overset{(1)}{=} \mathbb{E}_{\mathrm{P}}[YZ(t)] \overset{(2)}{=} \Lambda_P(Y)(t) \\ &\overset{(3)}{=} \sum_k \theta_k \Lambda_P(U_k)(t) \overset{(4)}{=} \sum_k \theta_k (1 + a_k) g_k,\end{aligned} \tag{5.21}$$

where (1) follows from the Gaussian Dichotomy Theorem, see equation (3.30), as both measures

$\mathrm{P}_{\bar{z}}$ and P are elements of $\mathscr{P}$ and therefore equivalent, and furthermore $\{Z(t) : t \in T\}$ is a zero mean Gaussian process under P. Proposition 3.11 yields equality (2) and equality (3) is a result of the expansion $Y = \sum_k \theta_k U_k$ for some $\boldsymbol{\theta} = \{\theta_k\} \in \ell^2$. The last part, equality (4), is derived from $\Lambda_P(U_k) = (1 + a_k)g_k$ ([1], p. 33). Defining $\mu_k = \theta_k(1 + a_k)$ gives

$$\bar{z} = \sum_k \mu_k g_k. \tag{5.22}$$

Define the parameter space of $\mathscr{P}$ by

$$\mathcal{K} = \Big\{(\bar{z}, \mathrm{P}) : \bar{z} = \sum_k \mu_k g_k, \boldsymbol{\mu} \in \ell^2, P = P_0 + \sum_k a_k g_k \otimes g_k, \boldsymbol{a} \in \ell_c^2, \\ \{g_k\} \subset \mathcal{H}(P_0) \text{ countable and orthonormal}\Big\}. \tag{5.23}$$

It is hard to find a sieve in this very large space, but by fixing a complete orthonormal system $\{g_k\}$ it is possible to find a sieve in the subset

$$\mathcal{K}' = \Big\{(\bar{z}, \mathrm{P}) : \bar{z} = \sum_k \mu_k g_k, \boldsymbol{\mu} \in \ell^2, P = P_0 + \sum_k a_k g_k \otimes g_k, \boldsymbol{a} \in \ell_c^2\Big\} \tag{5.24}$$

Consider the collection

$$\mathcal{S}_d = \{(\boldsymbol{\mu}, \boldsymbol{a}) \in \ell^2 \times \ell_c^2 : \mu_k = a_k = 0 \text{ for all } k > d\} \tag{5.25}$$

which is a sieve of $\mathcal{K}'$. By finding sieve estimators $\hat{\boldsymbol{\mu}}$ and $\tilde{\boldsymbol{a}}$, one can compute the estimator for $(\bar{z}, \bar{P})$. In [1] it is shown that under the probability measure $\mathrm{P}_{\bar{z}}$, which endows the process $\{z(t) : t \in T\}$ with a Gaussian$(\bar{z}, P)$ distribution, $\{U_k\}$ is a sequence of independent normally distributed random variables in the associated Gaussian space. In particular, $U_k \sim N(\mu_k, 1+a_k)$. The unbiased restricted maximum likelihood estimators $\hat{\boldsymbol{\mu}}$ and $\tilde{\boldsymbol{a}}$ are derived in [1], Lemma 4.1.1

and Theorem 4.1.2, pp. 34-37:

$$\hat{\mu}_k = \begin{cases} \bar{U}_k & \text{, if } k \leq d, \\ 0 & \text{, otherwise} \end{cases} \text{, and} \tag{5.26}$$

$$\tilde{a}_k = \begin{cases} \frac{1}{n-1} \sum_{i=1}^{n} (U_k^{(i)} - \bar{U}_k)^2 - 1 & \text{, if } k \leq d, \\ 0 & \text{, otherwise} \end{cases} . \tag{5.27}$$

Consequently, the joint estimator of $(\bar{z}, P)$ is given by $(\hat{\bar{z}}, \hat{P})$, where

$$\hat{\bar{z}} = \sum_{k=1}^{d} \hat{\mu}_k g_k, \text{ and} \tag{5.28}$$

$$\hat{P} = P_0 + \sum_{k=1}^{d} \tilde{a}_k g_k \otimes g_k. \tag{5.29}$$

The estimation of the selection gradient is analogous to before. Calculate the sequence $\tilde{\boldsymbol{b}}$ with the same estimator $\hat{\boldsymbol{c}}$, defined in equation (5.12), and the new estimator $\tilde{\boldsymbol{a}}$ according to equation (5.5), and hence

$$\tilde{\beta} = \sum_{k=1}^{d} \tilde{b}_k \gamma_k. \tag{5.30}$$

5.1.5. Estimating the Next-generation Mean $\bar{z}'$

The next-generation mean $\bar{z}'$ is determined by the evolutionary response to selection s, specifically the selection gradient β. By the Breeder's Equation for function-valued traits, an estimate of the evolutionary response to selection can be computed as

$$\widehat{\Delta \bar{z}} = \mathcal{G} \hat{\beta}. \tag{5.31}$$

The estimator for β is given in the previous section, and the computation of $\widehat{\Delta\bar{z}}$ becomes straightforward. It holds

$$\widehat{\Delta\bar{z}} = \mathcal{G}\hat{\beta} = \mathcal{G}\sum_{k=1}^{d}\tilde{b}_k\gamma_k = \sum_{k=1}^{d}\tilde{b}\mathcal{G}\gamma_k, \tag{5.32}$$

where

$$\mathcal{G}\gamma_k = \mathcal{G}\sum_i q_{ki}\delta_{t_i} = \sum_i q_{ki}G_{t_i}. \tag{5.33}$$

For the additive-genetic covariance function G a candidate is chosen. In the estimations later, the same candidate as for the phenotypic covariance is used.

Lastly, the estimator for the next-generation follows the definition of $\Delta\bar{z}$, i.e.

$$\hat{\bar{z}}' = \hat{\bar{z}} + \widehat{\Delta\bar{z}}. \tag{5.34}$$

5.2. Dependent Case

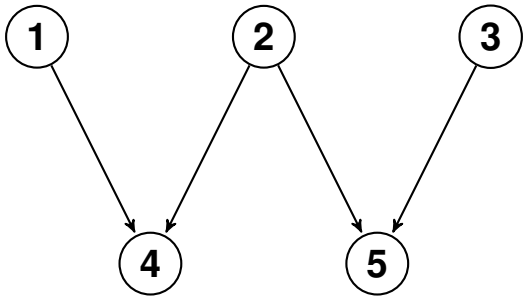
Contrary to the previous section, the observations of the function-valued trait $\{z(t) : t \in T\}$ are not independent anymore. It is assumed that the sample consists of independent families of individuals. In each family, the individuals are related to each other in some form (full siblings, half-siblings), determined the by the *relationship matrix* $\boldsymbol{A} = [A_{ij}]_{ij}$. The matrix entries A_{ij} specify the relationship between the organisms i and j of a sample. By [11], the covariance between a family full-siblings is

$$\text{Cov(full-sibling, full-sibling)} = \text{Cov(parent, offspring)} = \frac{1}{2}\boldsymbol{G}, \tag{5.35}$$

where $\boldsymbol{G}$ is the additive-genetic covariance matrix. Analogously for half-siblings, it holds

$$\text{Cov(half-sibling, half-sibling)} = \frac{1}{4}\boldsymbol{G}. \tag{5.36}$$

To illustrate, what an actual relationship matrix looks like, consider the following example from [15] p. 757:



Organism 4 is the offspring of 1 and 2, and 5 is the offspring of 2 and 3. The relationship matrix of this sample is given by

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 & 1 & \frac{1}{4} \\ 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & 1 \end{bmatrix}. \tag{5.37}$$

Let $(\Omega, \mathcal{A})$ be a measurable space and $\{z(t) : t \in T\}$ a stochastic process on $(\Omega, \mathcal{A})$. Consider a sample $z^{(1)}(t), \ldots, z^{(n)}(t)$ of the infinite-dimensional trait $z(t)$. Define

$$\begin{aligned} \boldsymbol{z}(t) &= [z^{(1)}(t), \ldots, z^{(n)}(t)]^T, \\ \boldsymbol{g}(t) &= [g^{(1)}(t), \ldots, g^{(n)}(t)]^T, \\ \boldsymbol{e}(t) &= [e^{(1)}(t), \ldots, e^{(n)}(t)]^T, \end{aligned} \tag{5.38}$$

of which each $z^{(i)}(t)$ is decomposed into the sum

$$z^{(i)}(t) = g^{(i)}(t) + e^{(i)}(t) \quad \text{for all } i = 1, \ldots, n, \tag{5.39}$$

where $g^{(i)}(t)$ is the additive genetic process and $e^{(i)}(t)$ the environmental process of the i^{th}

organism. It holds that

$$\mathrm{Cov}(\boldsymbol{g}(s), \boldsymbol{g}(t)) = \boldsymbol{A}G(s,t) \text{ for all } s,t \in T, \tag{5.40}$$

where $\boldsymbol{A}$ is the relationship matrix of the sample and $G(s,t)$ is the additive-genetic covariance function. Since it is assumed that all organisms are raised independently under the same conditions $e^{(1)}(t), \ldots, e^{(n)}(t)$ are independent, and it follows

$$\mathrm{Cov}(\boldsymbol{e}(s), \boldsymbol{e}(t)) = \boldsymbol{I}_n E(s,t) \text{ for all } s,t \in T, \tag{5.41}$$

where $\boldsymbol{I}_n$ is the identity matrix in $\mathbb{R}^{n\times n}$ and $E(s,t)$ is the environmental covariance function.

Note that, if all organisms have the same relation to each other, the relationship matrix $\boldsymbol{A}$ has the form

$$\boldsymbol{A} = \begin{bmatrix} 1 & a & \cdots & a \\ a & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a \\ a & \cdots & a & 1 \end{bmatrix} = (1-a)\boldsymbol{I}_n + a\boldsymbol{J}_n, \tag{5.42}$$

where a is the *relationship coefficient* ($a = \frac{1}{2}$ for full-siblings, $a = \frac{1}{4}$ for half-siblings), and $\boldsymbol{J}_n$ is the $n \times n$ matrix of 1's. It follows that, equation (5.40) reduces to

$$\mathrm{Cov}(\boldsymbol{g}(s), \boldsymbol{g}(t) = \left((1-a)\boldsymbol{I}_n + a\boldsymbol{J}_n\right) G(s,t) = \boldsymbol{I}_n G(s,t) + (\boldsymbol{J}_n - \boldsymbol{I}_n) a G(s,t). \tag{5.43}$$

Assuming, that there is no correlation between environmental and genetic factors, we conclude that

$$\begin{aligned} \mathrm{Cov}(\boldsymbol{z}(s), \boldsymbol{z}(t)) &= \mathrm{Cov}(\boldsymbol{g}(s), \boldsymbol{g}(t)) + \mathrm{Cov}(\boldsymbol{e}(s), \boldsymbol{e}(t)) \\ &= \boldsymbol{I}_n G(s,t) + (\boldsymbol{J}_n - \boldsymbol{I}_n) a G(s,t) + \boldsymbol{I}_n E(s,t) \\ &= \boldsymbol{I}_n (G(s,t) + E(s,t)) + (\boldsymbol{J}_n - \boldsymbol{I}_n) a G(s,t). \end{aligned} \tag{5.44}$$

From the independent case, it is known, that the phenotypic covariance function decomposes as the sum $P(s,t) = G(s,t) + E(s,t)$ and one can define the covariance function $\Psi(s,t) := aG(s,t)$, which leads to

$$\mathrm{Cov}(\boldsymbol{z}(s), \boldsymbol{z}(t)) = \boldsymbol{I}_n P(s,t) + (\boldsymbol{J}_n - \boldsymbol{I}_n)\Psi(s,t) = \boldsymbol{I}_n(P(s,t) - \Psi(s,t)) + \boldsymbol{J}_n\Psi(s,t). \tag{5.45}$$

As the exact derivation of the estimates for the covariance function P and the selection gradient β relies on a lot of technicalities, this thesis will skip the details at this point. The construction of the estimates of interest can be found in [1], Chapter 5. In the following, only the rough ideas that lead to the estimates will be outlined.

Define the matrix $\boldsymbol{P}(s,t)$ of covariance functions,

$$\boldsymbol{P}(s,t) = \mathrm{Cov}(\boldsymbol{z}(s), \boldsymbol{z}(t)) = \boldsymbol{I}_n(P(s,t) - \Psi(s,t)) + \boldsymbol{J}_n\Psi(s,t), \tag{5.46}$$

where $\boldsymbol{z}(t) = [z^{(1)}(t), \ldots, z^{(n)}(t)]^T$ is a vector of function-valued traits of equally related organisms. Under the measure $\mathbb{P}_0$ on the measure space $\big(\Omega, \mathcal{A} = \sigma(z_i(t), \forall i, t \in T)\big)$, the stochastic processes $\{\boldsymbol{z}(t), t \in T\}$ has mean zero and the covariance is given by

$$\boldsymbol{P}_0(s,t) = \boldsymbol{I}_n(P_0(s,t) - \Psi_0(s,t)) + \boldsymbol{J}_n\Psi_0(s,t) \tag{5.47}$$

One can diagonalize the matrix $\boldsymbol{P}(s,t)$ by

$$\boldsymbol{D}(s,t) = \boldsymbol{V}^{-1}\boldsymbol{P}(s,t)\boldsymbol{V}, \tag{5.48}$$

where the columns of $\boldsymbol{V}$ are orthonormal eigenfunctions of $\boldsymbol{P}(s,t)$, and the elements of the diagonal matrix $\boldsymbol{D}$ are the eigenvalues

$$D_{11}(s,t) = P(s,t) + (n-1)\Psi(s,t), \tag{5.49}$$

$$D_{jj}(s,t) = P(s,t) - \Psi(s,t) \quad \text{for all } j = 2, \ldots, n. \tag{5.50}$$

Transform $\boldsymbol{z}(t)$ to

$$\boldsymbol{y}(t) = V^{-1}\boldsymbol{z}(t). \tag{5.51}$$

It is easy to show

$$\mathrm{Cov}(\boldsymbol{y}(s), \boldsymbol{y}(t)) = \boldsymbol{D}(s,t). \tag{5.52}$$

Thus the covariance function for the process $\{y_1(t), t \in T\}$ is $P + (n-1)\Psi$ and the processes $\{y_1(t), t \in T\}, i = 2, \ldots, n$, have the covariance function $P - \Psi$. This leads to the consideration of two separate reproducing kernel Hilbert spaces, $\mathcal{H}(P_0 + (n-1)\Psi_0)$ and $\mathcal{H}(P_0 - \Psi_0)$, which are both subspaces of $\mathcal{H}(P_0)$ ([1] Lemma 5.1.3). Using the Gaussian Dichotomy Theorem, it is possible to expand $P + (n-1)\Psi$ and $P - \Psi$ to

$$P + (n-1)\Psi = P_0 + (n-1)\Psi_0 + \sum_k a_k^+ f_k^+ \otimes f_k^+, \text{ and} \tag{5.53}$$

$$P - \Psi = P_0 - \Psi_0 + \sum_k a_k^- f_k^- \otimes f_k^-, \tag{5.54}$$

where $\{f_k^+\}, \{f_k^-\}$ are complete orthonormal systems in $\mathcal{H}(P_0+(n-1)\Psi_0)$ and $\mathcal{H}(P_0-\Psi_0)$, respectively, and $\{a_k^+\}, \{a_k^-\}$ are sequences in ℓ_c^2 (square-summable and $\inf_k a_k > -1$). By computing estimates $\{\hat{a}_k^+\}, \{\hat{a}_k^-\}$, the estimates $\widehat{P + (n-1)\Psi}$ and $\widehat{P - \Psi}$ follow. The detailed estimates for $\{a_k^+\}, \{a_k^-\}$ are explained in [1] Chapter 5.

Under the same conditions as before, that is, the sample consists of independent families of equally related organisms, an estimator for the additive-genetic covariance matrix can be constructed. Consider $m \in \mathbb{N}$ families with relationship coefficient a of sizes $n_1, \ldots, n_m$, and $N = \sum_{j=1}^{m} n_j$. Then,

$$n_j\Psi = P + (n_j - 1)\Psi - (P - \Psi) \quad \text{for all } j = 1, \ldots, n. \tag{5.55}$$

Summing over all $j = 1, \ldots, m$ leads to

$$\begin{aligned}
\sum_{j=1}^{m} n_j \Psi &= \sum_{j=1}^{m} \Big(P + (n_j - 1)\Psi - (P - \Psi) \Big) \\
N\Psi &= \sum_{j=1}^{m} \Big(P + (n_j - 1)\Psi \Big) - m(P - \Psi) \\
\Psi &= \frac{1}{N} \sum_{j=1}^{m} \Big(P + (n_j - 1)\Psi \Big) - \frac{m}{N}(P - \Psi). \qquad (5.56)
\end{aligned}$$

Using the estimators of $P + (n_j - 1)\Psi$ and $P - \Psi$, specified before, it follows that

$$\hat{\Psi} = \frac{1}{N} \sum_{j=1}^{m} \widehat{(P + (n_j - 1)\Psi)} - \frac{m}{N} \widehat{(P - \Psi)}, \qquad (5.57)$$

and hence by the definition of Ψ, i.e. $\Psi = aG$,

$$\hat{G} = \frac{\hat{\Psi}}{a} = \frac{1}{aN} \sum_{j=1}^{m} \widehat{(P + (n_j - 1)\Psi)} - \frac{m}{aN} \widehat{(P - \Psi)}. \qquad (5.58)$$

When trying to estimate the selection gradient β in the independent case, estimates of the sequence $\{b_k\}$, where $b_k = \frac{c_k}{a_k + 1}$, are needed. For the dependent case, there is no estimator for the coefficients $\{c_k\}$ since the distribution of the fitness function W is unknown. A workaround, is to use the estimator of the independent case, see equations (5.12) and (5.13). The estimate for β is then given in section 5.1.3. The estimation of the evolutionary response to selection $\Delta\bar{z}$ follows the Breeder's equation for infinite-dimensional traits, see section 5.1.5 Lastly, to estimate the mean function of a trait among the newborns of the offspring generation $\bar{z}'$, an estimate of the parent generation's mean function $\bar{z}$ is needed, which is not provided. Thus for the estimation in this thesis the estimated mean function $\hat{\bar{z}}_{\text{ind}}$ from the independent case is used, and we get

$$\hat{\bar{z}}' = \widehat{\Delta\bar{z}} + \hat{\bar{z}}_{\text{ind}}. \qquad (5.59)$$

6. Estimates Based on Tribolium Castaneum Data

In this section, the results from estimations using live data on *Tribolium Castaneum* larvae growth curves are presented. With the theory and methods described in the previous chapters, estimates for the mean function of the larvae before selection (parent generation) and of the their offspring (offspring generation), as well as estimates for the covariance functions are computed. The MATLAB code for the estimations is based on a MATLAB script by Tyler Baur [1] with a few modifications, see Appendix A.

Data of the Tribolium larvae for generations 0,1,2,3, and 4 were provided by Carter and Irwin. Before processing the data in MATLAB, it had to be cleaned using Microsoft Excel. Individuals with incomplete data were removed, for example in case of missing information on the weight, the sire, or dam of the individual. The biggest subset of individuals with measurements at the same age was chosen. Due to sample size restrictions, it was decided to include observations with slightly different ages at measurements, e.g. all observations of larvae where the first measurement was conducted at age 1 and 2 (age in days) were included in the estimation process and a weighted average was assigned as the new age at first measurement.

The estimations of the pre-selection mean, the evolutionary response to selection, and the phenotypic covariance function were run using the *Ornstein-Uhlenbeck* covariance function and the *Wiener* covariance function as candidate covariance functions P_0. Furthermore, all weight data on the larvae was *log-transformed* prior to estimation. Lastly, the observations are once assumed to be independent, and once seen as a collection of independent families of full siblings.

The difference in estimation of those two cases are described in Chapter 5. Note that using the pre-selection mean and the evolutionary response to selection, the mean function among newborns of the successive generation is computed according to equation (5.34). The selection gradient is not explicitly estimated as it is contained in the computation of the evolutionary response to selection. Furthermore, it is a functional, and illustrating it by itself is not possible. One needs to apply it to some function which happens to be the case for the evolutionary response to selection.

To compute estimates of the functions of interest, it is essential to find estimates for the sequences of coefficients $\{a_k\}, \{b_k\}, \{c_k\}$, and $\{\mu_k\}$. The sequence $\{a_k\}$ belongs to the space ℓ_c^2, see equation 5.20, and $\{\mu_k\}$ is a sequence in ℓ^2 (equation (5.19)). Computing estimates is straightforward using equation (5.5), which requires knowledge about the sequence $\{c_k\}$. By the Robertson-Price identity $c_k = \mathrm{Cov}(U_k, w)$. To determine the fitness W of a trait $z(t)$, assume that $W = \nu(X)$ for some $X \in H_P$, i.e. X has the form $X = \sum_k \lambda_i z(t_k)$. Two useful forms of X are

$$X = Z(t^*), \tag{6.1}$$

for some specific $t^* \in T$, or

$$X = \int_T Z(t) f(t) \mathrm{d}\mu(t), \tag{6.2}$$

where f is a square-integrable function on T. In practice, we will compute X as the sum $X = \sum_k Z(t_k) f(t_k)$, and use *directional selection* for the fitness, that is

$$W = \exp(X). \tag{6.3}$$

For more detail on other forms of fitness functions, see [10], Section 6.

The MATLAB script written for these tests gives more detail on the computation, and can be found in Appendix B. Only a shortened version for Generation is given here to prevent repetition and an unnecessary long thesis. Running the estimations and generating the plots for all other generations is straightforward and follows the same commands as for Generation

0. Due to different data for every generation, the ranges in the import commands need to be adjusted to fit the Excel spreadsheet containing the data. Also, note that, due to the fact that estimations were run for 5 generations, each with the Ornstein-Uhlenbeck covariance function and Wiener covariance function as a candidate covariance, each run resulting in multiple estimates, a considerable amount of plots was generated. To keep this thesis at a reasonable length, only a small selection of the results are shown, mainly focusing on generations 0 and 1.

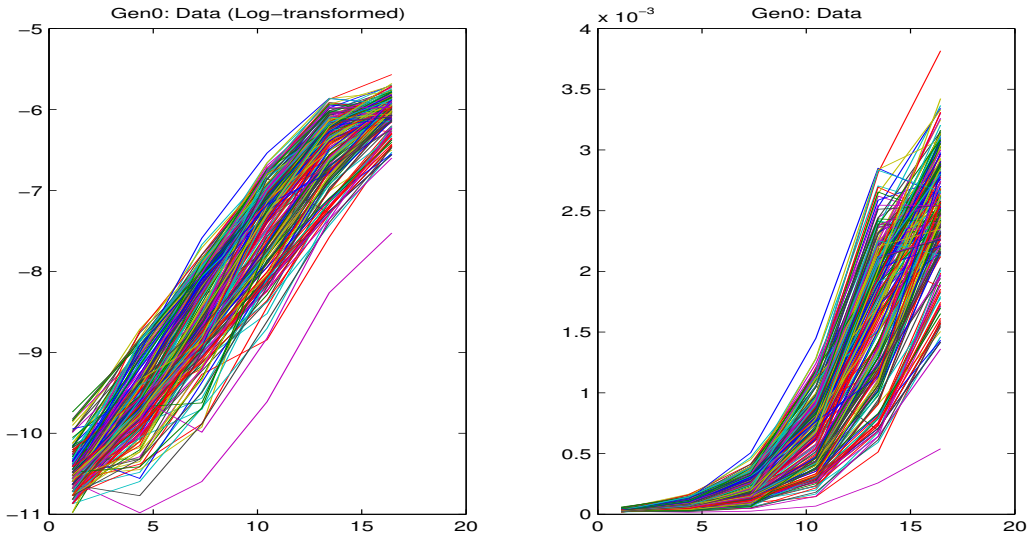


Figure 6.1.: Generation 0 data, log-transformed on the right

Figure 6.1 illustrates the weight of 224 organisms in Generation 0, measured on 6 different times in their larval period. The average larval period lasts about 15 to 17 days. Since dealing with very small numbers (the weight of the beetle larvae lies in the region of a thousandth of a gram) is difficult, a log-transformation makes sense to spread out the data to a wider range of values. Note that these plots do not show functions. The given data only provides point measurements, the lines are just connecting those points to illustrate growth.

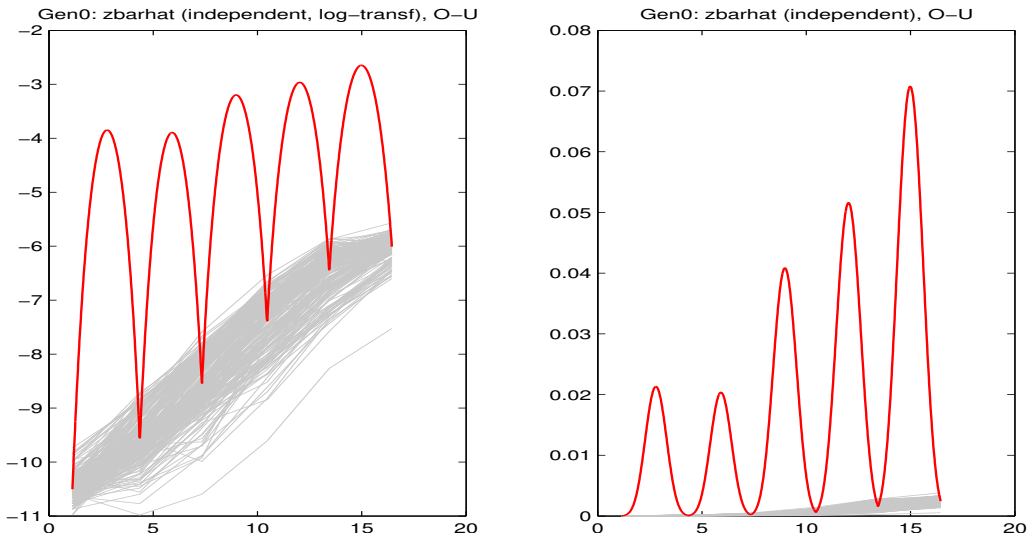


Figure 6.2.: Gen0: Mean function among organisms of the current generation $\hat{\bar{z}}$ using Ornstein-Uhlenbeck covariance function, based on log-transformed data on the left, re-transformed on the right

6.1. Ornstein-Uhlenbeck Covariance Function

The first estimations based on generation 0 use an Ornstein-Uhlenbeck covariance function as the candidate covariance function P_0, i.e.

$$P_0(s,t) = \exp(-|s-t|) \quad s,t \in T. \tag{6.4}$$

Starting with the assumption that all organisms are independent, the red graph in Figure 6.2 shows the mean function among the larvae of generation 0. Left picture shows the estimated mean function based on the log-transformed date, whereas the right refers to untransformed data. The light gray areas, indicate the lines the growth data. Comparing the estimated mean function of the offspring generation $\hat{\bar{z}}'$ with the data on generation 1, the same phenomenon is seen (Figure 6.3). The estimated phenotypic covariance function $\hat{P}$ for generation 0 is given in Figure 6.4. The bumps, which were observable in the previous figures, appear again.

For the dependent case, there exists no estimate for the mean function of the current generation $\bar{z}$. In the test script the mean function of the independent case was used instead. The estimated

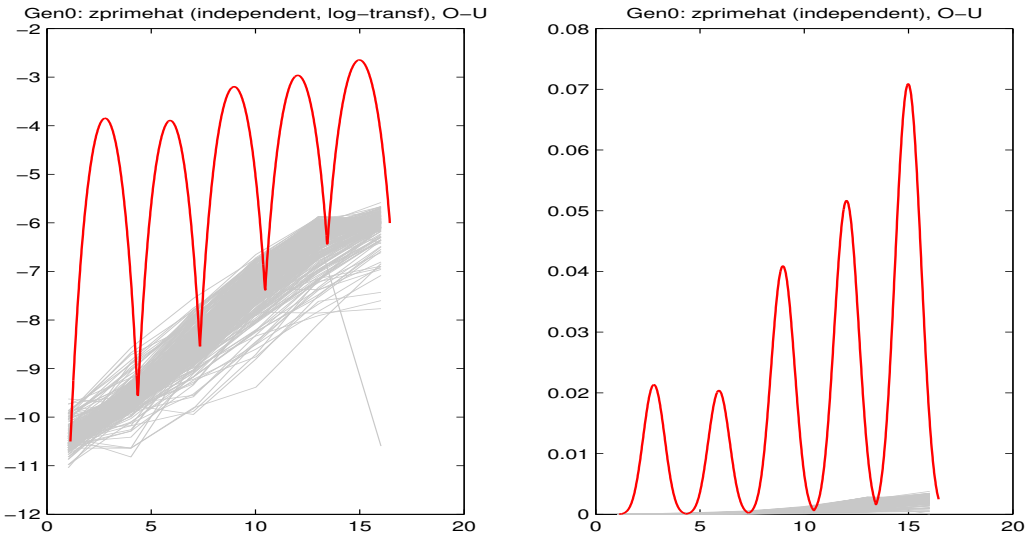


Figure 6.3.: Gen0: Mean function among organisms of the offspring generation $\hat{\bar{z}}'$ using Ornstein-Uhlenbeck covariance function, based on log-transformed data on the left, re-transformed on the right

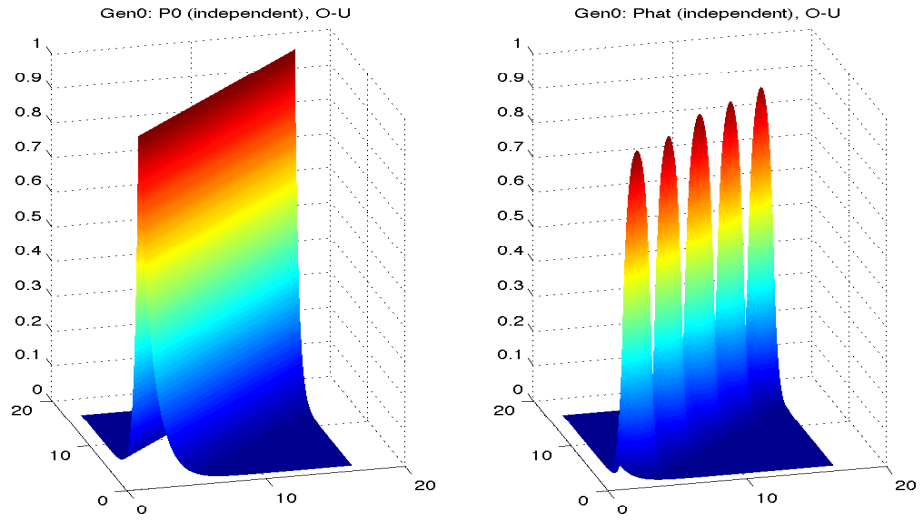


Figure 6.4.: Gen0: Ornstein-Uhlenbeck candidate covariance function P_0 and the estimateed phenotypic covariance function $\hat{P}$

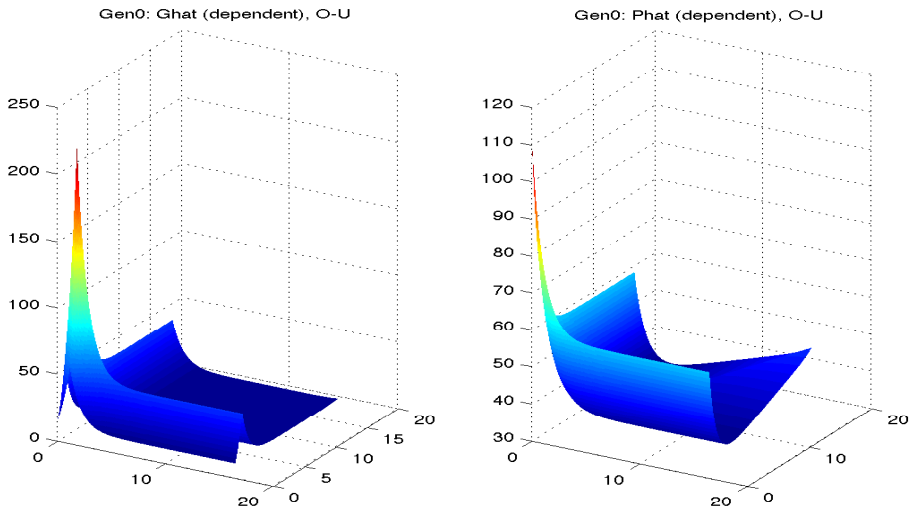


Figure 6.5.: Gen0: $\hat{G}$ and $\hat{P}$ in the dependent case using Ornstein-Uhlenbeck covariance function

mean function among the newborns of the next generation $\bar{z}'$ looks almost identical to the mean of the independent case in Figure 6.3.

Figure 6.5 depicts the estimates for the additive-genetic covariance function G and the phenotypic covariance function P, respectively.

6.2. Wiener Covariance Function

Using the Wiener covariance function

$$P_0(s,t) = \min(s,t) \tag{6.5}$$

different estimates are observed. Figure 6.6 shows $\hat{\bar{z}}$, the estimate for the pre-selection mean of Generation 0, and Figure 6.7 gives the estimated next-generation mean $\hat{\bar{z}}'$ based on Generation 0. Figure 6.8 compares the Wiener covariance function to the estimate of the phenotypic covariance function and the same comparison is given in Figure 6.9 for the dependent case. As mentioned before, there is no estimate for $\bar{z}$ and the next-generation mean $\bar{z}'$ is computed using the estimates from the independent case. The generated plots show almost identical results for the dependent

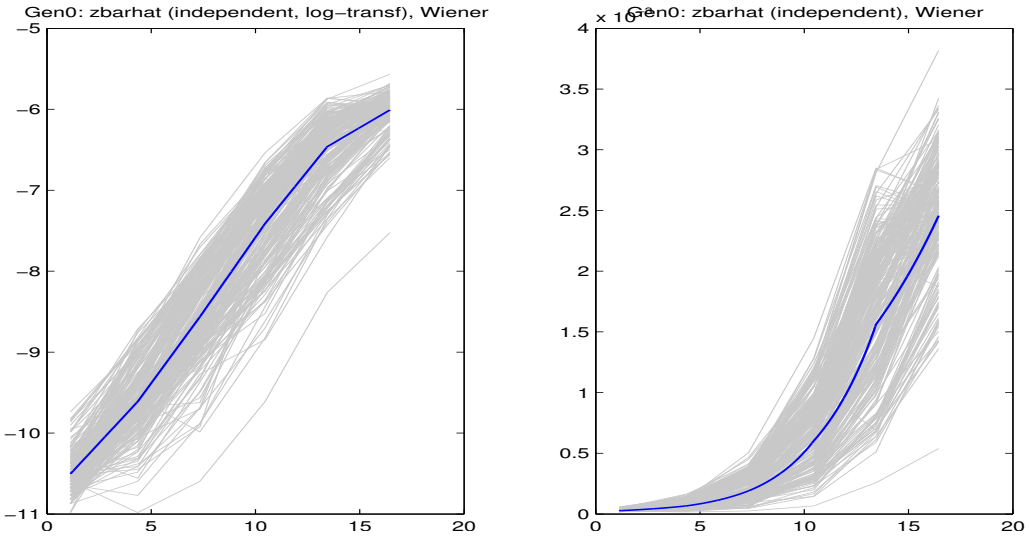


Figure 6.6.: Gen0: Mean function among organisms of the current generation $\hat{\bar{z}}$ using Wiener covariance function, based on log-transformed data on the left, re-transformed on the right

case and are therefore not shown here.

6.3. Comparison between Ornstein-Uhlenbeck and Wiener

Clearly using the Ornstein-Uhlenbeck covariance function as stated before gives a bad estimate for the pre-selection mean as well as for the next-generation mean. The observable local minima are located at the points in time of the actual measurements, and represent the closest $\hat{\bar{z}}$ gets to the data, something one would expect. At least for the time points at which data is given the estimate should be the most precise. But the mean weight for every time in-between is highly overestimated. The same phenomenon is observable for the estimated covariance function. High spikes are observable. In-between those spikes, the spatial minima are located at the time points of the given data. It seems like the candidate covariance function can be adjusted for those time points only. Missing data in-between those times results in the spikes.

Using the Wiener covariance function as the candidate, better estimates for the mean functions are achieved. The mean functions fit the data. At $t \approx 13$ a sharp bend is observable. A reason for

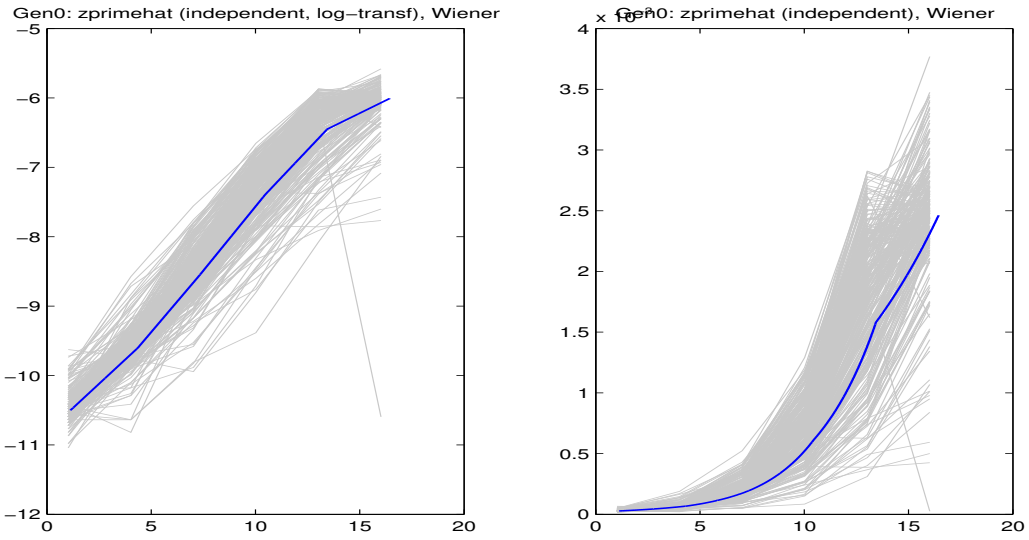


Figure 6.7.: Gen0: Mean function among organisms of the offspring generation $\hat{\bar{z}}'$ using Wiener covariance function, based on log-transformed data on the left, re-transformed on the right

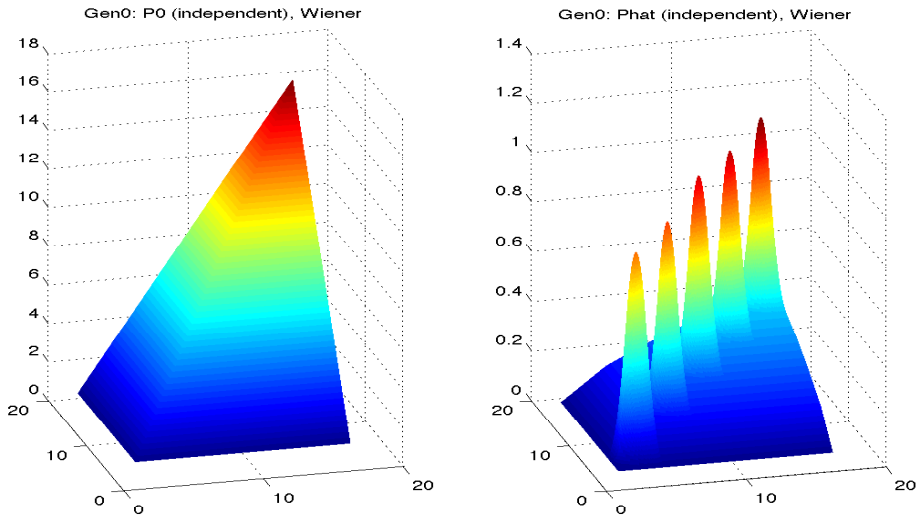


Figure 6.8.: Gen0: Wiener candidate covariance function P_0 and the estimated phenotypic covariance function $\hat{P}$

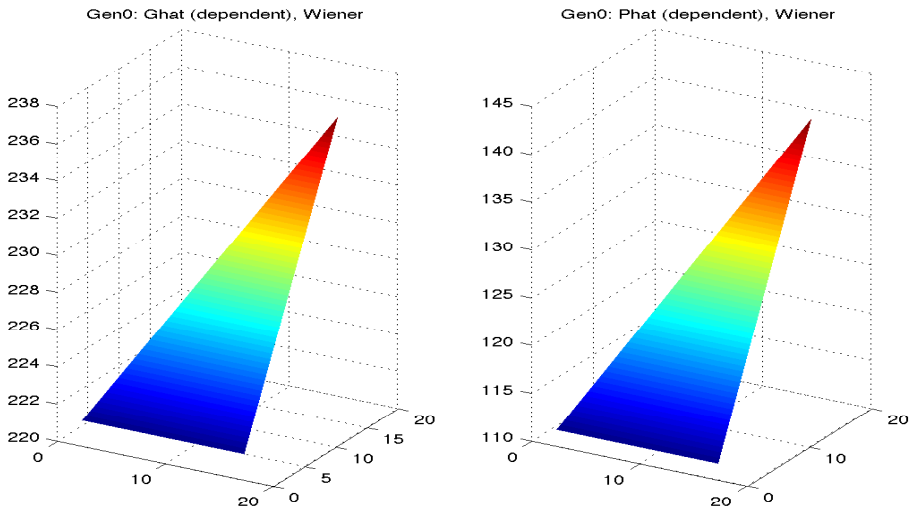


Figure 6.9.: Gen0: $\hat{G}$ and $\hat{P}$ in the dependent case using Wiener covariance function

this can be the fact that some organisms get lighter towards the end of their larval period. Due to the time adjustments made in the beginning to achieve a bigger sample size, the decrease in weight for those organisms appears earlier in time. Using basis functions generated from the Wiener covariance function might capture this and impact the estimated mean functions more directly. One would expect the estimated covariance function to have a decent form, but unfortunately $\hat{P}$ shows spikes in the Wiener-case too.

Figures 6.10 and 6.11 show a comparison of the results for $\hat{\hat{z}}$ and $\hat{P}$ between the two candidate covariance functions based on the date of Generation 0. The same comparison for all generations is depicted in figures 6.12 and 6.13. From the results of these first tests, it is clear that the estimation is highly dependent of the choice of the candidate covariance function used. The Ornstein-Uhlenbeck covariance functions gives bad results in general, whereas from the Wiener covariance function acceptable estimates for the mean functions are achieved, but still the estimated covariance seems not right. A question that arises immediately is how to choose the right candidate covariance function.

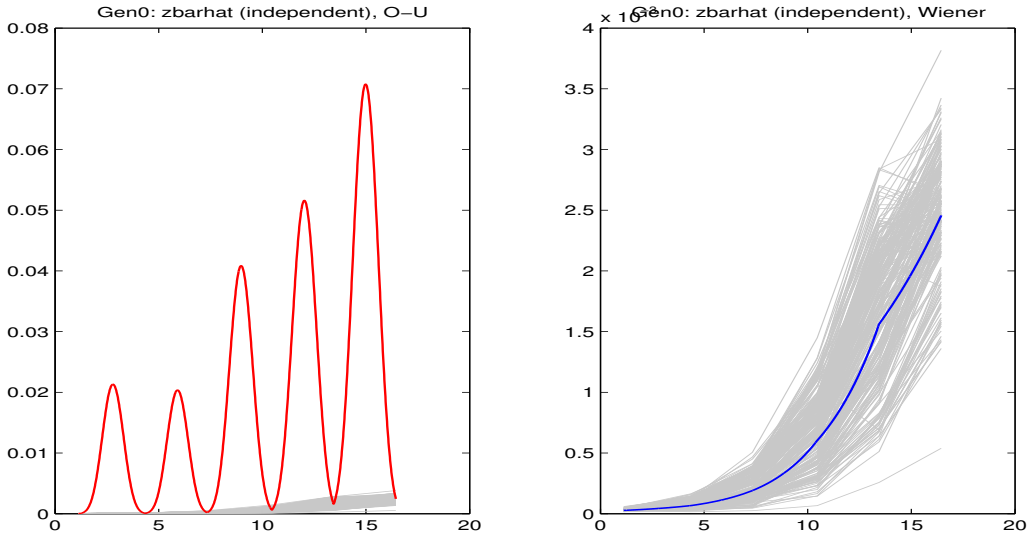


Figure 6.10.: Gen0: Comparison of estimated pre-selection mean $\hat{\bar{z}}$, Ornstein-Uhlenbeck and Wiener

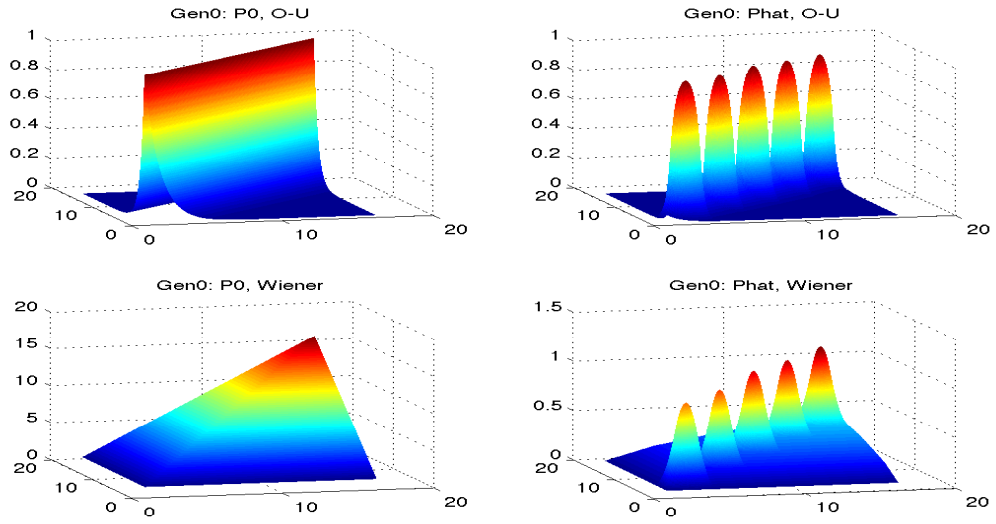


Figure 6.11.: Gen0: Comparison of estimated phenotypic covariance function $\hat{P}$, upper row Ornstein-Uhlenbeck, lower row Wiener

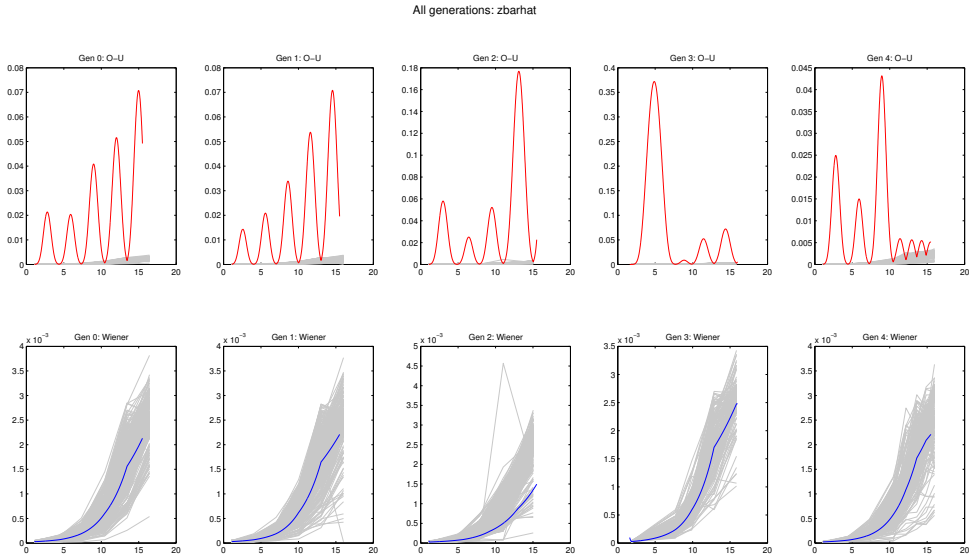


Figure 6.12.: All generations: $\hat{\bar{z}}$, upper row Ornstein-Uhlenbeck, lower row Wiener

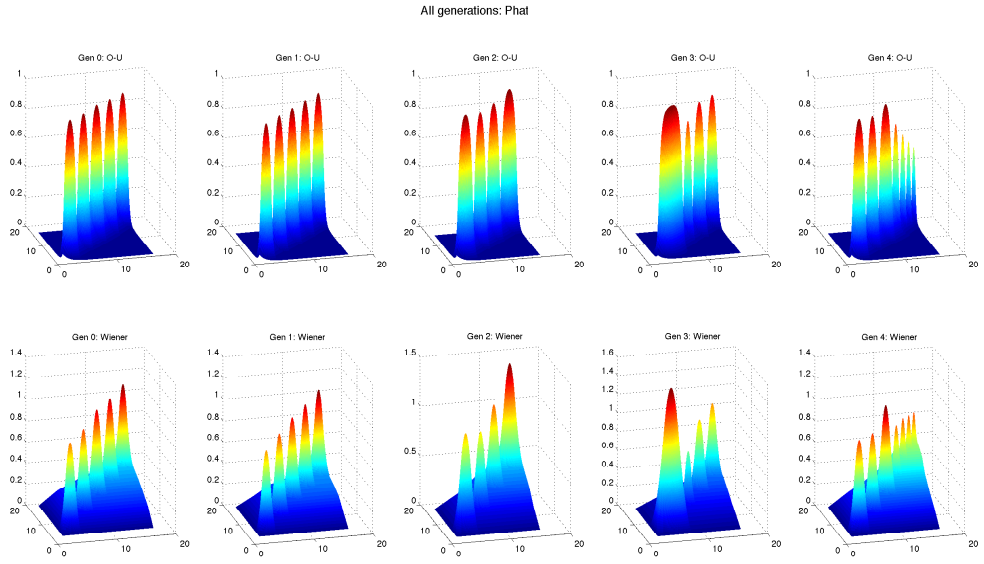


Figure 6.13.: All generations: $\hat{P}$, upper row Ornstein-Uhlenbeck, lower row Wiener

7. Alternative Candidate Covariance Function

The choice of a proper candidate covariance is crucial for the estimation of the mean functions $\bar{z}, \bar{z}'$ and the phenotypic covariance function P. In the following, further choices are discussed.

7.1. Carter and Irwin

A natural choice for the candidate covariance function is the estimated additive-genetic covariance function provided by Carter and Irwin. In their papers [5], [7], [6], the covariance function G is estimated by fitting their covariance data of the *Tribolium castaneum* larvae to a random multiple regression model of the form

$$G \sim s + t + st \tag{7.1}$$

with regression coefficients $\beta_0, \beta_1, \beta_2$ and β_3. In other words the additive covariance function is of the form

$$G(s,t) = \beta_0 + \beta_1 s + \beta_2 t + \beta_3 st. \tag{7.2}$$

A more detailed view on the estimation of the coefficients can be read in Carter's and Irwin's description [6]. Note that these β_i have nothing to do with the selection gradient β, this thesis is just following the notation of Carter and Irwin. Their resulting polynomial for G is

$$G(s,t) = 0.0449 - 0.00297s - 0.00297t + 0.000208st. \tag{7.3}$$

Using this function in our estimation is problematic as matrices $\boldsymbol{G} = [G(t_i, t_j)]_{i,j=1,\dots,d}$ are only positive semi-definite, whereas positive definite matrices are needed. One has to investigate the positive-definiteness of matrices generated by functions of the form (7.4).

Any covariance kernel has to fulfill the conditions of Definition 3.8, that is it has to be symmetric and positive definite. It is easy to show, that symmetry holds if and only if $\beta_1 = \beta_2$, thus by changing the notation of the β_i, G has the form

$$G(s,t) = \beta_0 + \beta_1(s+t) + \beta_2 st. \tag{7.4}$$

The goal is to find conditions on β_0, β_1, and β_2 such that a function of the form (7.4) is positive definite and to check whether the candidate given by Carter and Irwin fulfills those conditions or not. G is positive definite if and only if for all $d \in \mathbb{N}$ and time points $t_1, \dots, t_d$ the matrix

$$\boldsymbol{G} = [G(t_i, t_j)]_{i,j=1,\dots,d} = \begin{bmatrix} G(t_1,t_1) & \cdots & G(t_1,t_d) \\ \vdots & \ddots & \vdots \\ G(t_d,t_1) & \cdots & G(t_d,t_d) \end{bmatrix} \tag{7.5}$$

is positive definite, i.e. for any vector $\boldsymbol{x} = [x_1, \dots, x_d]^T \in \mathbb{R}$

$$\boldsymbol{x}^T \boldsymbol{G} \boldsymbol{x} \geq 0 \quad (= 0 \Leftrightarrow \boldsymbol{x} = \boldsymbol{0}). \tag{7.6}$$

Using simply matrix algebra, the matrix $\boldsymbol{G}$ can be written as

$$\begin{aligned} \boldsymbol{G} &= [\beta_0 + \beta_1(t_i + t_j) + \beta_2 t_i t_j]_{i,j=1}^n \\ &= \beta_0 \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} + \beta_1 \left(\begin{bmatrix} t_1 & \cdots & t_d \\ \vdots & \cdots & \vdots \\ t_1 & \cdots & t_d \end{bmatrix} + \begin{bmatrix} t_1 & \cdots & t_1 \\ \vdots & \cdots & \vdots \\ t_d & \cdots & t_d \end{bmatrix} \right) + \beta_2 \begin{bmatrix} t_1 t_1 & \cdots & t_1 t_d \\ \vdots & \ddots & \vdots \\ t_d t_1 & \cdots & t_d t_d \end{bmatrix} \\ &= \beta_0 \boldsymbol{1}\, \boldsymbol{1}^T + \beta_1(\boldsymbol{1}\, \boldsymbol{t}^T + \boldsymbol{t}\, \boldsymbol{1}^T) + \beta_2 \boldsymbol{t}\, \boldsymbol{t}^T \end{aligned} \tag{7.7}$$

where $\mathbf{1} = [1, \ldots, 1]^T \in \mathbb{R}^d$ and $\boldsymbol{t} = [t_1, \ldots, t_d]^T$. Following equation (7.7) it holds that

$$\boldsymbol{x}^T \left[G(t_i, t_j)\right]_{i,j=1}^{d} \boldsymbol{x} = \beta_0 \boldsymbol{x}^T \mathbf{1}\ \mathbf{1}^T \boldsymbol{x} + \beta_1 (\boldsymbol{x}^T \mathbf{1}\ \boldsymbol{t}^T \boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{t}\ \mathbf{1}^T \boldsymbol{x}) + \beta_2 \boldsymbol{x}^T \boldsymbol{t}\ \boldsymbol{t}^T \boldsymbol{x}. \tag{7.8}$$

Let

$$\begin{aligned} \boldsymbol{x}^T \mathbf{1} = \mathbf{1}^T \boldsymbol{x} = \sum_{i=1}^{d} x_i = u \in \mathbb{R}, \\ \boldsymbol{x}^T \boldsymbol{t} = \boldsymbol{t}^T \boldsymbol{x} = \sum_{i=1}^{d} t_i x_i = v \in \mathbb{R}. \end{aligned} \tag{7.9}$$

Then the condition for positivity can be stated as follows

$$\begin{aligned} \boldsymbol{x}^T \boldsymbol{G} \boldsymbol{x} &= \beta_0 u^2 + 2\beta_1 uv + \beta_3 v^2 \\ &= u^2 \left(\beta_0 + 2\beta_1 \frac{v}{u} + \beta_3 \left(\frac{v}{u} \right)^2 \right) \geq 0 \end{aligned} \tag{7.10}$$

for all $d \in \mathbb{N}$, time points $\boldsymbol{t} = [t_1, \ldots, t_d]^T$, and $\boldsymbol{x} = [x_1, \ldots, x_d]^T \in \mathbb{R}$. Examine the quadratic polynomial $P\left(\frac{v}{u}\right)$ in $\frac{v}{u} \in \mathbb{R}$,

$$P\left(\frac{v}{u}\right) = \beta_0 + 2\beta_1 \frac{v}{u} + \beta_3 \left(\frac{v}{u} \right)^2, \tag{7.11}$$

and consider following cases:

(a) $\underline{\beta_2 = 0}$: $P\left(\frac{v}{u}\right)$ reduces to a linear function in $\frac{v}{u}$,

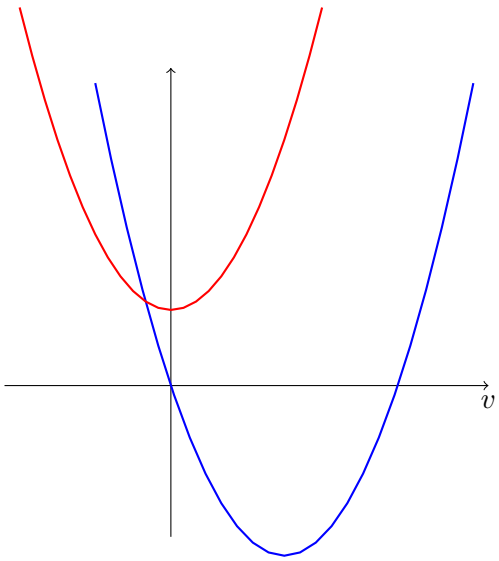
$$P\left(\frac{v}{u}\right) = \beta_0 + 2\beta_1 \frac{v}{u}, \tag{7.12}$$

and can therefore not be greater than or equal to 0 for all $\frac{v}{u} \in \mathbb{R}$.

(b) $\underline{\beta_2 < 0}$: $P\left(\frac{v}{u}\right)$ is a concave quadratic function, i.e. its graph is a parabola with opening facing downwards, and hence $P\left(\frac{v}{u}\right) \ngeq 0$ for all $\frac{v}{u} \in \mathbb{R}$.

(c) $\underline{\beta_2 > 0}$: $P\left(\frac{v}{u}\right)$ is a convex quadratic polynomial in $\frac{v}{u} \in \mathbb{R}$. It is greater than 0 for all $\frac{v}{u} \in \mathbb{R}$

if and only if it has no roots, i.e. if its discriminant is less than 0.



The discriminant of $P\left(\frac{v}{u}\right)$ is easily computed and has the form

$$\text{Discr}_P = (2\beta_1)^2 - 4\beta_0\beta_1 = 4(\beta_1^2 - \beta_0\beta_2). \tag{7.13}$$

This is less than 0 if and only if

$$\beta_1^2 - \beta_0\beta_2 < 0. \tag{7.14}$$

Going back to the additive genetic covariance function given by Carter and Irwin

$$G(s,t) = 0.0449 - 0.00297s - 0.00297t + 0.000208st, \tag{7.15}$$

we have

$$\beta_0 = 0.0449, \beta_1 - 0.00297, \beta_2 = 0.000208. \tag{7.16}$$

This function fulfills the conditions specified since

$$\beta_1^2 - \beta_0\beta_3 = -5.183 \times 10^{-7} \leq 0. \tag{7.17}$$

Unfortunately, this condition is not sufficient to guarantee that the generated matrices are positive definite. With the help of the computer algebra software Maple, one can easily compute the determinant of matrices $\boldsymbol{G} = [G(t_i, t_j)]_{i,j=1,\dots,d}$ for general β_i and time points $t_1, \dots, t_d$. Examining the matrices generated by $d > 2$ time points, their determinant is equal to 0 for all $beta_i, t_i$. Thus any matrix generated by more than two time points can only be positive semidefinite. A more rigorous but easy proof, explaining the results from Maple, follows.

Consider the matrix-vector product of $\boldsymbol{G}$ with an arbitrary vector $\boldsymbol{x} \in \mathbb{R}^d$ and set this equal to $\mathbf{0}$,

$$\begin{aligned}\boldsymbol{G}\boldsymbol{x} &= \beta_0 \mathbf{1}\ \mathbf{1}^T\boldsymbol{x} + \beta_1(\mathbf{1}\ \boldsymbol{t}^T\boldsymbol{x} + \boldsymbol{t}\ \mathbf{1}^T\boldsymbol{x}) + \beta_2 \boldsymbol{t}\ \boldsymbol{t}^T\boldsymbol{x} \\ &= \left(\beta_0 \sum_{i=1}^{d} x_i\right)\mathbf{1} + \left(\beta_1 \sum_{i=1}^{d} t_i x_i\right)\mathbf{1} + \left(\beta_1 \sum_{i=1}^{d} x_i\right)\boldsymbol{t} + \left(\beta_2 \sum_{i=1}^{d} t_i x_i\right)\boldsymbol{t} = \mathbf{0}, \end{aligned} \tag{7.18}$$

where $\mathbf{0}$ is the vector of 0's in $\mathbb{R}^d$. This is a linear combination of the linearly independent vectors $\mathbf{1}$ and $\boldsymbol{t}$ which is only equal to the zero-vector if and only if the respective coefficients are all equal to 0. That is

$$\begin{cases} \beta_0 \sum\limits_{i=1}^{d} x_i + \beta_1 \sum\limits_{i=1}^{d} t_i x_i &= 0 \\ \beta_1 \sum\limits_{i=1}^{d} x_i + \beta_2 \sum\limits_{i=1}^{d} t_i x_i &= 0 \end{cases} \tag{7.19}$$

which is equivalent to

$$\begin{bmatrix} \beta_0 & \beta_1 \\ \beta_1 & \beta_2 \end{bmatrix} \begin{bmatrix} \sum\limits_{i=1}^{d} x_i \\ \sum\limits_{i=1}^{d} t_i x_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{7.20}$$

This has a solution if the matrix consisting of the β_i is regular, i.e. its determinant is non-zero and the inverse matrix exists. The determinant of the matrix consisting of the β_i is $\beta_1^2 - \beta_0\beta_2$. Initially a condition for β_0, β_1, and β_2 was found, see equation (7.14), which is exactly $\beta_1^2 - \beta_0\beta_2 < 0$.

Thus this matrix is invertible, and as a result,

$$\begin{bmatrix} \sum_{i=1}^{d} x_i \\ \sum_{i=1}^{d} t_i x_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{7.21}$$

This is a system of two equations an $d \in \mathbb{N}$ unknowns x_i. For $d > 2$ there are infinitely many non-trivial solutions $\boldsymbol{x} \in \mathbb{R}^d$ such that $\boldsymbol{x}^T \boldsymbol{G} \boldsymbol{x} = 0$. Therefore the matrix $\boldsymbol{G}$ and its corresponding function G are only positive semidefinite for $d > 2$ time points $t_1, \ldots, t_d$.

In general, data provides more than two points in time, and thus Carter and Irwin's covariance function cannot be used.. An alternative way to find a proper candidate covariance function must be found.

7.2. Fitting Orthonormal Functions to the Data

Kirkpatrick, Lofsvold and Bulmer published a paper [14] described a method to estimate the additive-genetic covariance function $\mathcal{G}$.. This method involves fitting *orthogonal functions* to the observed $n \times n$ *sample covariance matrix* $\hat{\boldsymbol{G}}$.

Remark 7.1.(Notation) In the following, $\mathcal{G}$ will denote the continuous covariance function (not an integral operator as before), whereas boldface characters like $\hat{\boldsymbol{G}}$ or $\boldsymbol{t}$ will be used for matrices and column vectors, respectively. This notation is chosen as this is the notation of the paper. Also, it is easier to distinguish $\hat{\mathcal{G}}$ from $\hat{\boldsymbol{G}}$ than $\hat{G}$ from $\hat{\boldsymbol{G}}$.

Consider a data set of observations of a infinite-dimensional trait z of n individuals and measurements at d time points for each individual (in matrix form) $[z_{ij}]_{ij} = [z_i(t_j)]_{ij},\ i = 1, \ldots, n,\ j = 1, \ldots, n$. Denote the time points by $t_1, \ldots, t_d$.

The sample covariance matrix $\hat{\boldsymbol{G}}$ is computed, where the i, j-th entry is the sample covariance

of the trait at time t_i and t_j, i.e.

$$\hat{G}_{ij} = \frac{1}{n}\sum_{k=1}^{n}\Big(z_k(t_i) - \bar{z}(t_i)\Big)\Big(z_k(t_j) - \bar{z}(t_j)\Big). \tag{7.22}$$

This matrix gives direct estimates for the covariance function $\mathcal{G}$ at d^2 points as $\hat{\boldsymbol{G}}_{ij} = \hat{\mathcal{G}}(t_i, t_j)$. To get a full continuous estimate $\hat{\mathcal{G}}$, smooth curves are fitted to the data. The paper approaches this using orthonormal polynomials, in particular *normalized Legendre polynomials.*

Definition 7.2.(Orthonormal functions) A pair of functions ϕ_i and ϕ_j is *orthogonal* and *normalized* on an interval $[a, b]$ if

$$\int_a^b \phi_i(x)\phi_j(x)\mathrm{d}x = 0 \quad \text{and} \quad \int_a^b \phi_i^2(x)\mathrm{d}x = 1. \tag{7.23}$$

Let $(\phi_i), i = 0, 1, 2, \ldots,$ defined on the interval $[a, b]$, be a complete orthonormal basis. Then, for all $s, t \in T$, the additive-genetic covariance function $\mathcal{G}$ can be written as the linear combination of the orthonormal functions (ϕ_i)

$$\mathcal{G}(s, t) = \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} c_{kl}\phi_k(s^*)\phi_l(t^*) \tag{7.24}$$

where

$$t^* = a + \frac{b-a}{t_{\max} - t_{\min}}(t - t_{\min}), \tag{7.25}$$

with $t_{\min}$ and $t_{\max}$ being the smallest and largest data time points, and the c_{kl} are the coefficients of the linear combination. To find an estimate $\hat{\mathcal{G}}$ for the additive-genetic covariance function, it is essential to estimate the coefficients c_{kl}.

From equation (7.24), it is clear that, for the time points $t_0, \ldots, t_n$ given by the data it holds

$$\mathcal{G}(t_i, t_j) = \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} c_{kl}\phi_k(t_i^*)\phi_l(t_j^*) \quad \text{for all } i, j = 0, \ldots, d. \tag{7.26}$$

As mentioned before, the entries $\hat{G}_{ij}$ of the sample covariance matrix $\hat{\boldsymbol{G}}$ are direct estimates of

$\mathcal{G}(t_i, t_j)$, that means $mathcalG(t_i, t_j) = \hat{G}_{ij}$, and it follows that for the estimate $\hat{\mathcal{G}}$

$$\begin{aligned}\hat{G}_{ij} = \hat{\mathcal{G}}(t_i, t_j) &= \sum_{k=0}^{d}\sum_{l=0}^{d} \hat{c}_{kl}\phi_k(t_i^*)\phi_l(t_j^*) \\ &= \begin{bmatrix}\phi_0(t_i^*), \ldots, \phi_d(t_i^*)\end{bmatrix} \underbrace{\begin{bmatrix}\hat{c}_{00} & \cdots & \hat{c}_{0d} \\ \vdots & & \vdots \\ \hat{c}_{d0} & \cdots & \hat{c}_{dd}\end{bmatrix}}_{\hat{\boldsymbol{C}}} \begin{bmatrix}\phi_0(t_j^*) \\ \vdots \\ \phi_d(t_j^*)\end{bmatrix}\end{aligned} \tag{7.27}$$

for all $i, j = 0, \ldots, d$. Thus the $d \times d$ sample covariance matrix $\hat{\boldsymbol{G}}$ takes the form

$$\hat{\boldsymbol{G}} = \boldsymbol{\Phi}\hat{\boldsymbol{C}}\boldsymbol{\Phi}^T \tag{7.28}$$

where $\boldsymbol{\Phi}$ is the matrix defined by $\boldsymbol{\Phi}_{ij} = \phi_j(t_i)$ for all $i, j = 0, \ldots, d$, i.e.

$$\boldsymbol{\Phi} = \begin{bmatrix}\phi_0(t_0^*) & \cdots & \phi_d(t_0^*) \\ \vdots & & \vdots \\ \phi_0(t_d^*) & \cdots & \phi_d(t_d^*)\end{bmatrix}. \tag{7.29}$$

Kirkpatrick also calls $\hat{\boldsymbol{C}}$ the *coefficient matrix* (in his paper this matrix is sub-scripted by $\hat{\boldsymbol{G}}$ to show its dependence on the sample covariance matrix). The previous implies that an estimator for the coefficient matrix is then given by

$$\hat{\boldsymbol{C}} = \boldsymbol{\Phi}^{-1}\hat{\boldsymbol{G}}[\boldsymbol{\Phi}^T]^{-1}. \tag{7.30}$$

Consequently, for any $s, t \in T$, an estimator for the additive-genetic covariance function obtained

$$\hat{\mathcal{G}}(s, t) = \sum_{k=0}^{d}\sum_{l=0}^{d} \hat{c}_{kl}\phi_k(s^*)\phi_l(t^*). \tag{7.31}$$

In their paper, Kirkpatrick, Lofsvold and Bulmer use *normalized Legendre polynomials* as the basis of orthonormal functions.

Definition 7.3.(Normalized Legendre polynomials) The j-th *normalized Legendre polynomial* ϕ_j is defined by

$$\phi_j(x) = \frac{1}{2^j}\sqrt{j+\frac{1}{2}} \cdot \sum_{m=0}^{\lfloor j/2 \rfloor} (-1)^m \binom{j}{m} \binom{2j-2m}{j} x^{j-2m}, \; j = 0, 1, \ldots \tag{7.32}$$

where $\lfloor \cdot \rfloor$ is the floor function, that rounds values down to the nearest integer. The normalized Legendre polynomials are defined on the interval $[-1, 1]$.

The choice of the family of orthonormal polynomials influences the interpolation of the covariance matrix $\hat{\boldsymbol{G}}$, except for the time points at which data was sampled ($\mathcal{G}(t_i, t_j) = \hat{\boldsymbol{G}}_{ij}$). The differences in the interpolation by using different orthonormal polynomials is minimized by the number of time points taken during the observation. It is also mentioned that the choice of fitting orthonormal functions over other methods like for example *splines* is due to the analytical benefits of the coefficients derived from using this method.

The method constructed above is named the *full estimate of* $\mathcal{G}$ by the authors of [14] as the number of orthonormal functions fitted to the sample covariance matrix $\hat{\boldsymbol{G}}$ is equal to the number of time points given by the data. An approach to find a *reduced estimate of* $\mathcal{G}$ is specified as well, where a smaller set of orthonormal functions $\phi_0, \ldots, \phi_k, k < d$, is fitted to $\hat{\boldsymbol{G}}$. The resulting reduced estimate $\tilde{\mathcal{G}}$ needs to be tested for goodness of fit. An approximate χ^2 test statistic is developed. This thesis will not go into detail here, a detailed explanation and examples can be read in the previously mentioned paper [14].

There are possible drawbacks to the method. Pletcher and Geyer [17] state that fitting orthonormal polynomials to the scattered data, i.e. the sample covariances, does not automatically conclude in a positive definite function. Furthermore, using polynomials of high degree results in high fluctuations in the estimated covariance function, as those polynomials are very "wiggly".

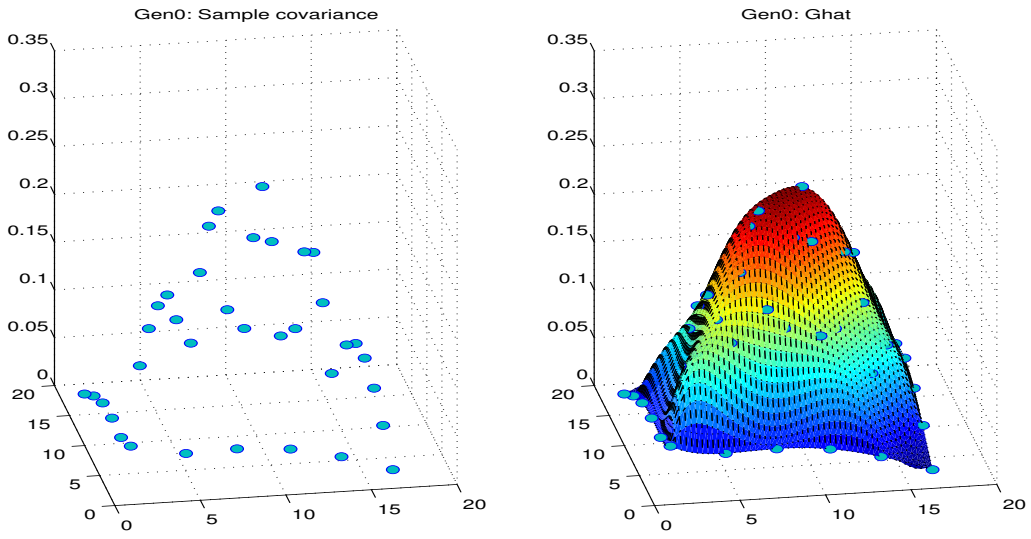


Figure 7.1.: Gen0: Sample covariances and estimated covariance function from fitting Legendre polynomials

7.3. Estimates of the Additive-genetic Covariance Function Using Legendre Polynomials

The data on Tribolium Castaneum larvae is used again to estimate the additive-genetic covariance function G, which is then used as a candidate covariance function to estimate the pre-selection mean function, next-generation, and the phenotypic covariance function. Again, for illustrative purposes, only results based on Generation 0 are presented. The MATLAB script and the implementation of all additional functions are given in the Appendix C. The test script only states the commands for the estimation based on Generation 0. Reproducing the estimation based on all other estimations is straightforward, only the ranges of the import lines need to be adjusted to the corresponding Excel file.

Figure 7.1 shows a scatterplot estimated covariances $\hat{G}_{ij} = \widehat{\text{Cov}(z(t_i), z(t_j))}$ for the data at the given time points $t_1, \ldots, t_d$. Next to that, the estimated covariance function from fitting normalized Legendre polynomials is illustrated. As mentioned before, this estimate is not guaranteed to be positive definite. For the estimation of the mean functions and phenotypic covariance function, a candidate that generates a positive definite matrix, from the time points $t_1, \ldots, t_d$

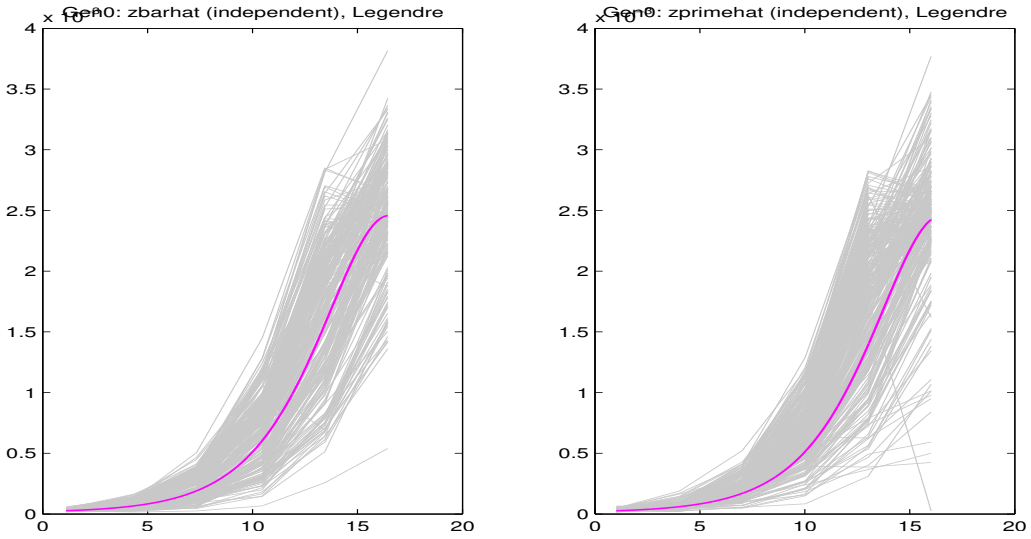


Figure 7.2.: Gen0: Estimated pre-selection mean $\hat{\bar{z}}$ and next-generation mean $\hat{\bar{z}}'$ from fitting Legendre polynomials

given by the data, is needed. The estimated covariance function from this method generates the sample covariance matrix using the time points from the data, which is positive definite. Thus the estimate can be used as a candidate covariance function. Note, that this does not mean that the matrices generated by this estimate are in general positive definite. The graphs in Figure 7.2 are the results of the estimation of the pre-selection mean $\bar{z}$ and the next-generation mean $\bar{z}'$. These estimates fit the data of Generation 0 and 1 very well. Also, the estimates achieved by using polynomials as basis functions are smooth. There are no kinks observable.

The estimate for the phenotypic covariance function P in the independent case is almost identical to the function shown in Figure 7.1, and therefore not explicitly shown anymore. For the dependent case, $\hat{P}$ is illustrated in Figure

The estimate $\bar{z}$ for all generations is shown in Figure 7.4.

On another note, as mentioned before, using polynomials of high degree, which are very wiggly, produces estimates with strong fluctuation. In Generation 4 more data time points are given which means higher degree Legendre polynomials are used in the estimation. The resulting estimate is depicted in Figure 7.5.

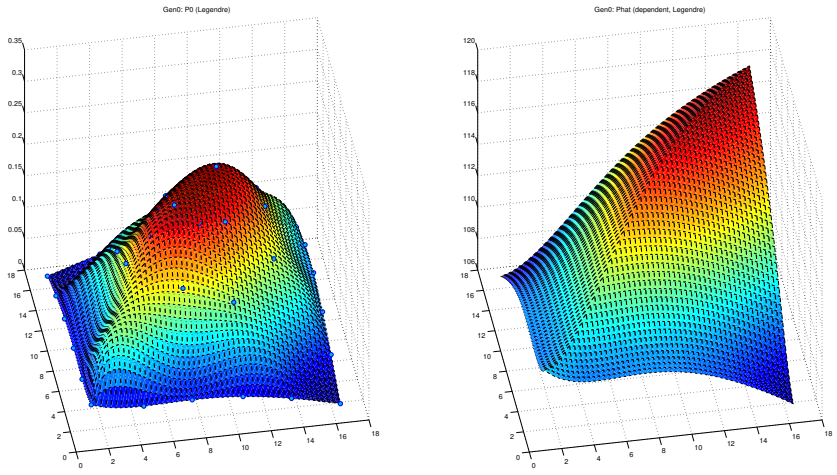


Figure 7.3.: Gen0: Estimated phenotypic covariance function (Legendre, dependent)

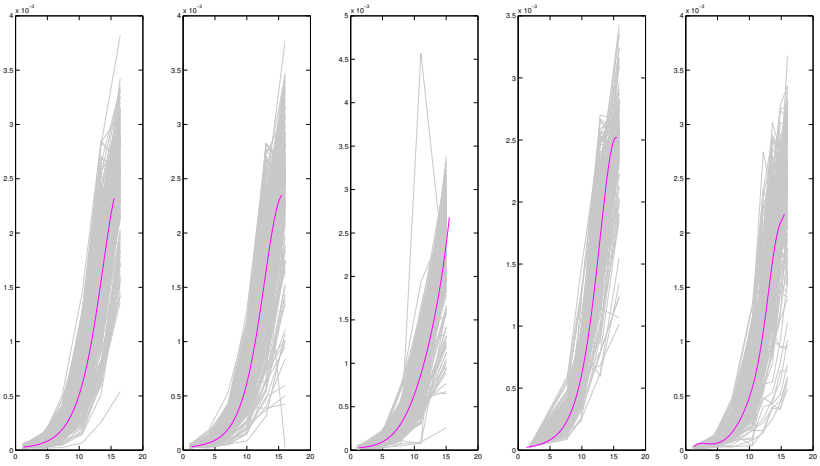


Figure 7.4.: Gen0 to Gen4 (left to right): Estimated pre-selection mean $\hat{\bar{z}}$ and next-generation mean $\hat{\bar{z}}'$ from fitting Legendre polynomials

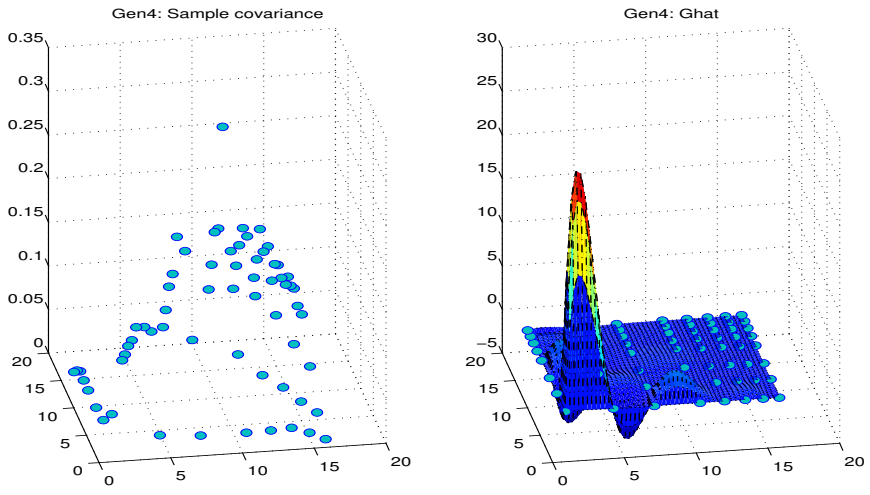


Figure 7.5.: Gen4: Scattered sample covariances and estimated covariance function from fitting Legendre polynomials

7.4. Adjusted Ornstein-Uhlenbeck and Wiener Covariance Function

Due to the mentioned disadvantage from fitting orthonormal polynomials to the data, going back to Ornstein-Uhlenbeck and Wiener candidate covariance functions is considered again. From observing the data in the previous section, it became clear that the original candidate covariance functions in Chapter 6 were badly chosen and did not fit the data, see Figure 7.6. Undoubtedly, the values of the Ornstein-Uhlenbeck covariance function are to high on the diagonal. By adjusting parameters, one can change the surface to better fit the data, i.e lower the center and widen the body. Analogously, the original Wiener covariance used before produces values that are too high compared to the data.

In a sense, the approach described fits a covariance function to the data. Obviously a preliminary examination of the covariances between the data is needed. Note that the parameters chosen in the following are only for illustration and by no means the perfect choice. Parameter

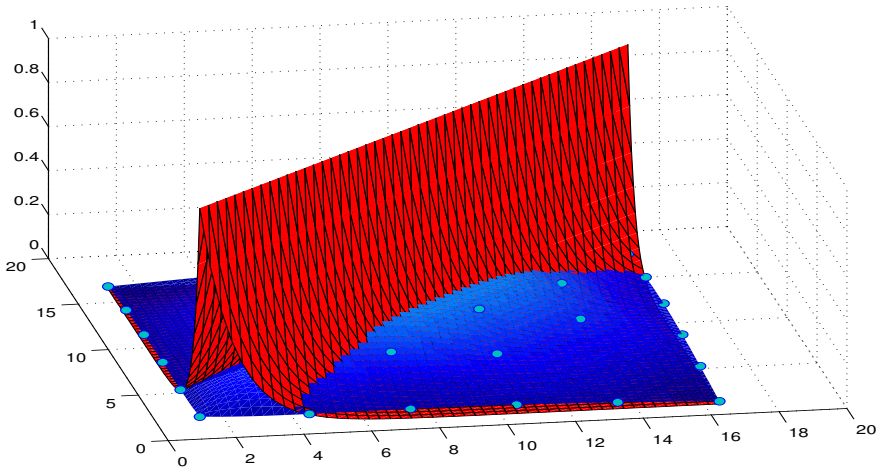


Figure 7.6.: Gen0: Comparison between data (Legendre fitted) and Ornstein-Uhlenbeck covariance function

estimation is a whole new question by itself. We use the following candidate covariance functions

$$P_0^{\mathrm{OU1}}(s,t) = \exp(-0.25|s-t|) \tag{7.33}$$

$$P_0^{\mathrm{OU2}}(s,t) = 0.25\exp(-0.1|s-t|) \tag{7.34}$$

$$P_0^{\mathrm{W}}(s,t) = 10^{-2}\min(s,t), \tag{7.35}$$

the first two being adjustments on the Ornstein-Uhlenbeck covariance function, whereas the last is a downscaled Wiener covariance function. Examining the resulting estimated phenotypic covariance function, there are noticable improvements observable, see Figure 7.7. The first plot, shows a slight reduction in the height of the spikes, still the adjustments made seem to be insufficient. This makes sense, as only the width of the original Ornstein-Uhlenbeck was changed. The second adjustment additionally lowers the center, which results in a far better estimated covariance function, "better" in a sense that there are almost no spikes anymore. One can still see small peaks in the estimate, hinting at a possibly insufficient adjustment. In the case of the Wiener candidate covariance function the spikes previously witness completely disappear. It seems like the adjustments made here cause a significant improvement. Looking at the estimates

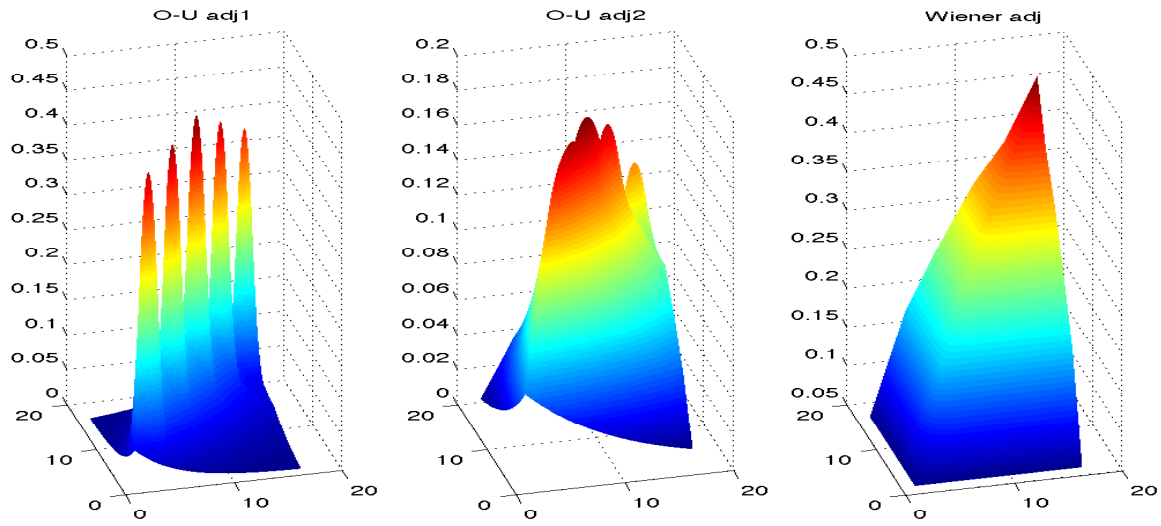


Figure 7.7.: Gen0: Estimated phenotypic covariance function $\hat{P}$

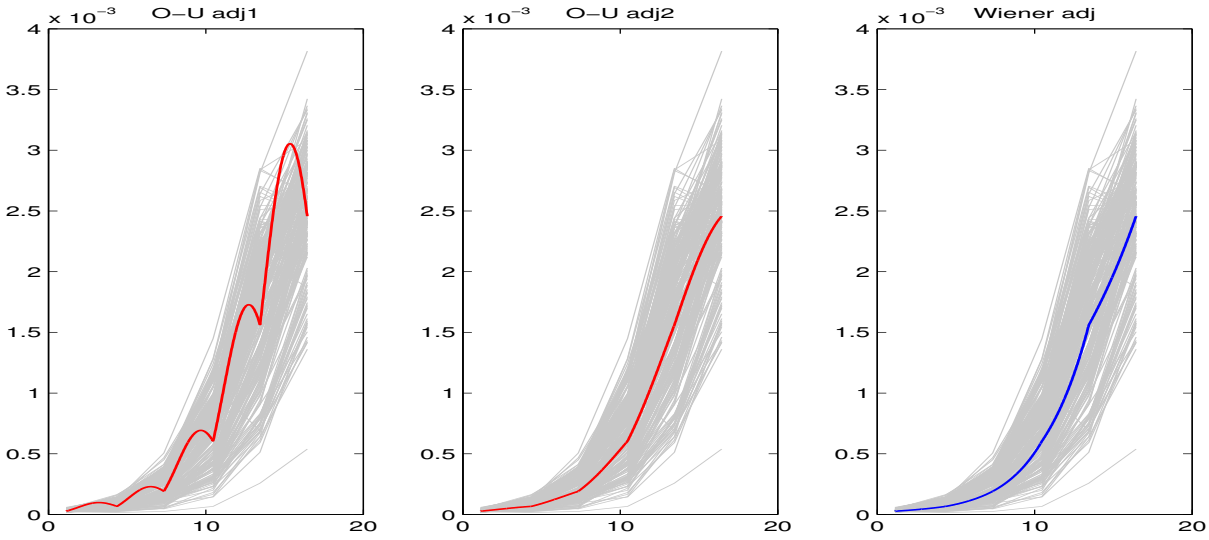


Figure 7.8.: Gen0: Estimated pre-selection mean function $\hat{\hat{z}}$

for the pre-selection mean $\hat{\bar{z}}$, the benefits from fitting the covariance functions to the data are even clearer. In the Figure 7.8 the $\hat{\bar{z}}$ corresponding to the adjustments are shown. The first plot shows the estimate using the first adjustment on the Ornstein-Uhlenbeck covariance function. The fit to the growth data is better, the bumps are less striking but still noticeable. For the second adjustment of the Ornstein-Uhlenbeck covariance function, $\hat{\bar{z}}$ is almost a smooth curve. Closely examining it, one can still see minor kinks at the time points of the given data. This confirms the previous speculation that the fit achieved by the adjustments could be better. Yet, as mentioned before, finding the best fit involves parameter estimation. Lastly, the resulting estimated pre-selection mean using the downscaled Wiener covariance function is no different from the one achieved before in chapter 6, as that was already a decent estimate. The kink described before is still observable.

8. Discussion

In evolutionary biology, infinite-dimensional (function-valued) traits are a much more complex structure, making the estimation of quantities of interest difficult when using methods from classical quantitative genetics which deals with finite-dimensional (vector-valued) traits only. In presence of reproducing kernel Hilbert spaces and the assumption that infinite-dimensional traits are Gaussian, the quantitative genetic model is extended. The Gaussian Dichotomy Theorem as well as the Breeder's Equation and Robertson-Price Identity for infinite-dimensional traits enable the computation of the phenotypic covariance function P, pre-selection mean function $\bar{z}$, and next-generation mean function $\bar{z}'$, in particular. Those estimates are proven to be asymptotically unbiased and weakly consistent [1].

Using the sieve estimators of all the necessary quantities, the estimation of $P, \bar{z}$, and $\bar{z}'$ is tested based on real-life data on Tribolium Castaneum larvae of five generations. Estimations were run for all generations using the Ornstein-Uhlenbeck covariance function and the Wiener covariance function as candidates P_0. Easily one can see that the results of the estimations were highly dependent of the choice of the candidate covariance function.

A method which fits orthonormal Legendre polynomials to the data was introduced. The estimated covariance function is an interpolation of scattered sample covariance data points. Using this smooth function as a candidate covariance function in the estimation of the phenotypic covariance function and the mean functions, results in estimates that fit the data well. As stated before, this method does not guarantee that the estimate is automatically positive definite. Furthermore, using many data points which involves polynomials of higher degree, which are very wiggly, results in an estimate covariance function with significant fluctuation.

To avoid this, fitting a known covariance function, such as the Ornstein-Uhlenbeck and Wiener covariance used before, to the data can be considered. This requires a preliminary examination of the sample covariances at the given time points of the data. Finding the right parameters to adjust the Ornstein-Uhlenbeck and Wiener covariance function, involves parameter estimation. This opens up a whole new world of possibilities. Still, guessing parameters such that the covariance functions roughly fit the data, and using those as candidate covariance functions in the estimations, prove to be a significant improvement.

Although it is possible to compare the estimates among each other, a judgment on whether an estimate is the best is hard to pass, as there is no benchmark to measure the estimate with. One can only compare the estimated functions to the underlying data.

When assuming that the data is clustered in independent families of equally-related organisms, for example independent families of full-siblings, the absence of certain estimators, make it difficult to give an analysis comparable to the extent of the independent case. Still the phenotypic covariance function for different candidate covariance functions could be estimated, but again the results are conditional on the choice of the candidate function, and a best estimate cannot be determined.

Equally important, *Principal Component Analysis* is an alternative view on genetic covariance functions. Covariance functions can be decomposed as the sum of eigenvalues λ_k and eigenfunctions φ_k [20], i.e.

$$G(s,t) = \sum_k \lambda_k \varphi_k(s)\varphi_k(t), \tag{8.1}$$

where the eigenfunction φ_k represents a direction of genetic variation and the eigenvalue λ_k corresponds to the extent of variation of that direction. In practice, one only examines the principal components that make up the most of variation, that is to say only the eigenvalues, and their eigenfunctions that amount to e.g. 95% of the variation are considered. Principal component analysis also opens up new possibilities for the estimation of covariance functions.

Bibliography

[1] Baur, Tyler: ESTIMATING THE SELECTION GRADIENT OF A FUNCTION-VALUED TRAIT. (2016)

[2] Beder, Jay: ESTIMATING THE SELECTION GRADIENT OF FUNCTION-VALUED TRAITS.

[3] Berlinet, Alain ; Thomas-Agnan, Christine: *REPRODUCING KERNEL HILBERT SPACES IN PROBABILITY AND STATISTICS*. Springer, 2004

[4] Bürger, R: *THE MATHEMATICAL THEORY OF SELECTION, RECOMBINATION, AND MUTATION*. Wiley, 2000

[5] Carter, Patrick A. ; Irwin, Kristen: CONSTRAINTS ON THE EVOLUTION OF FUNCTION-VALUED TRAITS: A STUDY OF GROWTH IN TRIBOLIUM CASTANEUM. (2013)

[6] Carter, Patrick A. ; Irwin, Kristen: G-FUNCTION FROM TRIBOLIUM CASTANEUM GROWTH CURVES. (2013)

[7] Carter, Patrick A. ; Irwin, Kristen: ARTIFICIAL SELECTION ON LARVAL GROWTH CURVES IN TRIBOLIUM: CORRELATED RESPONSES AND CONSTRAINTS. (2014)

[8] Gaydos, Travis ; Heckmann, Nancy ; Kirkpatrick, Mark ; Stinchcombe, John ; Schmitt, Johanna ; Kingsolver, Joel ; Marron, J: VISUALIZING GENETIC CONSTRAINTS. (2012)

[9] Gomulkiewicz, Richard ; Beder, Jay H.: THE SELECTION GRADIENT OF AN INFINITE-DIMENSIONAL TRAIT. In: *SIAM Journal on Applied Mathematics* 56 (1996), S. 509–523

[10] Gomulkiewicz, Richard ; Beder, Jay H.: COMPUTING THE SELECTION GRADIENT AND EVOLUTIONARY RESPONSE OF AN INFINTE-DIMENSIONAL TRAIT. In: *Journal of Mathematical Biology* 36 (1998), S. 299–319

[11] Heckmann, Nancy: FUNCTIONAL DATA ANALYSIS IN EVOLUTIONARY BIOLOGY. (2003)

[12] Kirkpatrick, Mark ; Heckmann, Nancy: A QUANTITATIVE GENETIC MODEL FOR GROWTH, SHAPE, REACTION NORMS, AND OTHER INFINITE-DIMENSIONAL CHARACTERS. (1989)

[13] Kirkpatrick, Mark ; Lofsvold, David: MEASURING SELECTION AND CONSTRAINT IN THE EVOLUTION OF GROWTH. (1992)

[14] Kirkpatrick, Mark ; Lofsvold, David ; Bulmer, Michael: ANALYSIS OF THE INHERITANCE, SELECTION AND EVOLUTION OF GROWTH TRAJECTORIES. (1990)

[15] Lynch, Michael ; Walsh, Bruce: *GENETICS AND ANALYSIS OF QUANTITATIVE TRAITS*. SINAUER, 1998

[16] Neveu, Jacques: *PROCESSUS ALÉATOIRES GAUSSIENS*. Seminaire de mathematiques superieures, Presses de l'Universite de Montreal, 1968

[17] Pletcher, Scott D. ; Geyer, Charles J.: THE GENETIC ANALYSIS OF AGE-DEPENDENT TRAITS: MODELING THE CHARACTER PROCESS. (1999)

[18] Rudin, Walter: *REAL AND COMPLEX ANALYSIS*. McGraw-Hill, 1987

[19] Rudin, Walter: *FUNCTIONAL ANALYSIS*. McGraw-Hill, 1991

[20] Stinchcombe, John R. ; Kirkpatrick, Mark: GENETICS AND EVOLUTION OF FUNCTION-VALUED TRAITS: UNDERSTANDING ENVIRONMENTALLY RESPONSIVE PHENOTYPES. (2012)

Appendix

A. Estimation of the Selection Gradient

```
function [ estimates, zprime_hat,zprime_hat_vec,P_hat,varargout] =
    selectgrad(Z,W,d,n,t,related,covtype,varargin)
% Estimates the selection gradient of a sample of organisms:
% The sample should either consist of unrelated organisms, or
    independent
% families of equally-related organisms with the same relationship
% for all families
%
% Written by Tyler Baur, 2016. Modified by Ly Viet Hoang, 2017.
%
%%%%%% Input %%%%%
% n = vector of family sizes (num_fams x 1 or 1 x num_fams if
%           related='related' or scalar if related='unrelated')
% t = grid of time points ( M x 1 or 1 x M)
% P0 = candidate covariance function (anonymous function)
% d = sieve parameter = o(min(n)) (anonymous function)
% Z = data (matrix ****(in rows)**** or cell, each cell should
    contain a
%           family with individuals stored in the rows of a matrix
% related = 'related' or 'unrelated'
% varargin: if related='unrelated', varargin{1}=P0, varargin{2}=G
%           if related='related', varargin{1}=G0, varargin{2}=E0,
%                       varargin{3}=num_fams, and varargin{4}=reln
% reln = relationship coefficient (scalar)
% num_fams = number of families (scalar)
%
% Output
% estimates = struct containing estimates of a,b,c (and mu if
%               related='unrelated')
% zprime_hat = estimated mean of the trait among newborns in the next
%                       generation (anonymous function)
% zprime_hat_vec = zprime_hat evaluated at t (M x 1)
% P_hat = estimated phenotypic covariance function (anonymous
    function)
% if related='unrelated', then
```

```
32 %       varargout{1} = zbar_hat = estimated mean function (anonymous
33 %           function)
34 %       varargout{2] = zbar_hat_vec = zbar_hat evaluated at t (M x 1)
35 % if related='related', then
36 %       varargout{1} = G_hat = estimated genetic covariance function
37 %           (anonymous function)
38 % Sample Commands
39 % If related='unrelated'
40 %  [ estimates, zprime_hat,zprime_hat_vec,P_hat,zbar_hat,zbar_hat_vec
       ] =
41 %                                    selectgrad(Z,W,d,n,t,'unrelated',
       P0,G)
42 % If related='related'
43 %  [ estimates, zprime_hat,zprime_hat_vec,P_hat,G_hat] =
44 %                              selectgrad(Z,W,d,n,t,'related',G0,E0,
       num_fams,reln)
45 %
46
47 %% Input check
48 if (size(t,1)>1 && size(t,2)>1)
49     error('t must be a vector')
50 else
51     if size(t,2)>1
52         t=t';
53     end
54 end
55 if (size(W,1)>1 && size(W,2)>1)
56     error('W must be a vector')
57 else
58     if size(W,2)>1
59         W=W';
60     end
61 end
62
63 %%%%% End Input check %%%%%
64
65 %% Cases: unrelated = independent, related =dependent
66 switch related
67     case 'unrelated'
68         P0=varargin{1};
69         G=varargin{2};
70         G_t=cell(length(t),1);
71         Z=Z';
72         for i=1:length(t)
73             G_t{i}=@(s)G(t(i),s);
```

```
74          end
75          [T,S]=meshgrid(t,t);
76
77          switch covtype
78              % Candidate covariance function is a known cov fn (e.g.
                    Wiener, Ornstein-Uhlenbeck, etc.)
79              case 'covfn'
80                  P0_matrix=P0(T,S);
81              % Candidate covariance function of the form G(s,t)=b0+b1
                    *(s+t)+b3*s*t.
82              % Computational issues (not pos. def.). This is just a
                    workaround and not a good solution in a mathematical
                    sense!
83              case 'regression'
84                  P0_matrix=P0(T,S);
85                  [V,D]=eig(P0_matrix);
86                  [tmp,~]=find(abs(diag(D))<10^-15);
87                  D_new=D;
88                  for i=1:length(tmp)
89                      D_new(tmp(i),tmp(i)) = 10^-15;
90                  end
91                  P0_matrix=V*D_new*inv(V);
92          end
93          chol_P0=chol(P0_matrix,'lower');
94          inv_chol_P0=chol_P0\eye(length(t));
95          g=ortho(t,P0,covtype);
96
97          dn=floor(d(n));
98          if dn>=length(t);
99              dn=length(t);
100         end
101
102         [a_hat, b_hat,c_hat, alpha_hat] = estimate_b_ind( Z,W,chol_P0
                ...
103              ,n,dn);
104         [P_hat]=compute_P(t,P0, a_hat,g);
105         [zbar_hat,zbar_hat_vec]=compute_zbar(alpha_hat,g,t);
106         estimates=struct('a',a_hat,'b',b_hat,'c',c_hat,'alpha',
                alpha_hat);
107         varargout{1}=zbar_hat;
108         varargout{2}=zbar_hat_vec;
109         [zprime_hat,zprime_hat_vec]=compute_zprime(G_t,inv_chol_P0,
                b_hat,t,zbar_hat,'unrelated');
110
111     case 'related'
```

```
112         estimates=struct('a',zeros(length(t),1),'b',zeros(length(t)
                ,1),...
113              'c',zeros(length(t),1),'a_minus',zeros(length(t),1));
114         G0=varargin{1};
115         E0=varargin{2};
116         num_fams=varargin{3};
117         reln=varargin{4};
118         Z_cell=cell(num_fams,1);
119         if ismatrix(Z)
120              start=1;
121              for i=1:num_fams
122                   tmp=sum(n(1:i));
123                   Z_cell{i}=Z(start:tmp,:);
124                   start=start+n(i);
125              end
126         elseif isa(Z,cell)
127              Z_cell=Z;
128         end
129         P0=@(s,t) G0(s,t)+E0(s,t);
130         Psi0=@(s,t)reln.*G0(s,t);
131         P0_minus=@(s,t)(1-reln).*G0(s,t)+E0(s,t);
132         [T,S]=meshgrid(t,t);
133         P0_matrix=P0(T,S);
134         chol_P0=chol(P0_matrix,'lower');
135         inv_chol_P0=chol_P0\eye(length(t));
136         g=ortho(t,P0,covtype);
137         f_minus=ortho(t,P0_minus,covtype);
138
139         P0_minus_mat=P0_minus(T,S);
140         chol_minus=chol(P0_minus_mat,'lower');
141         f_minus_temp=cell2mat(cellfun(@(x) x(t),f_minus,...
142              'UniformOutput',false));
143         f_minus_mat=reshape(f_minus_temp,length(t),length(t));
144         %   functions are in columns
145
146         % round the inner product
147         Ndecimals = 12;
148         H = 10.^Ndecimals ;
149         rkhs_ip_minus = round(H*(chol_P0\f_minus_mat))/H;
150
151         sum_P_plus_hat=@(s,t)0;
152         dn=floor(d(min(n)));
153         if dn>=length(t);
154              dn=length(t);
155         end
```

```
156         for i=1:num_fams
157             n_fam=n(i);
158             P0_plus=@(s,t)P0(s,t)+(n_fam-1).*Psi0(s,t);
159             f_plus=ortho(t,P0_plus,covtype);
160             P0_plus_mat=P0_plus(S,T);
161             chol_plus=chol(P0_plus_mat,'lower');
162             f_plus_temp=cell2mat(cellfun(@(x) x(t),f_plus,'
                    UniformOutput',false));
163             f_plus_mat=reshape(f_plus_temp,length(t),length(t));
164             %functions are in columns
165
166             % round inner product
167             Ndecimals = 12;
168             H = 10.^Ndecimals ;
169             rkhs_ip_plus = round(H*(chol_P0\f_plus_mat))/H;
170             %inner product <fk,gl> is the l,k entry
171
172             Transform=transform(n_fam);
173             Y=Transform'*Z_cell{i};
174             Y1=Y(1,:);
175             Yend=Y(2:end,:);
176             a_minus_temp=estimate_a( Yend',chol_minus,dn);
177
178
179             aplus_temp=estimate_a(Y1',chol_plus,dn)-[ones(dn,1);zeros
                    (length(t)-dn,1)];
180             [P_plus_hat]=compute_P(t,P0_plus,aplus_temp,f_plus);
181             estimates.a=estimates.a+(rkhs_ip_plus.^2*aplus_temp);
182             P_plus_temp=P_plus_hat;
183             sum_P_plus_hat=@(s,t)sum_P_plus_hat(s,t)+P_plus_temp(s,t)
                    ;
184         end
185         Z_all_mat=vertcat(Z_cell{:});
186         num_obs=size(Z,1);
187         estimates.a_minus=sum(a_minus_temp,3)./(num_obs-num_fams)-[
                ones(dn,1);zeros(length(t)-dn,1)];
188
189         estimates.a_minus((dn+1):end)=zeros(length(t)-dn,1);
190
191         [P_minus_hat]=compute_P(t,P0_minus,estimates.a_minus,f_minus)
                ;
192
193         estimates.a=estimates.a./num_obs+(rkhs_ip_minus.^2)*(
                estimates.a_minus).* (num_obs-num_fams)./(num_obs);
194         estimates.a((dn+1):end)=zeros(length(t)-dn,1);
```

```
195             estimates.c=estimate_c('Z_all_mat',W,chol_P0,num_obs,dn);
196             estimates.b=estimate_b(estimates.a,estimates.c);
197
198             P_hat=compute_P(t,P0,estimates.a,g);
199
200             Psi_hat=@(s,t)(sum_P_plus_hat(s,t))./num_obs-P_minus_hat(s,t)
                    .*num_fams./num_obs;
201             G_hat=@(s,t)Psi_hat(s,t)./reln;
202             G_hat_t=cell(length(t),1);
203             for i=1:length(t)
204                 G_hat_t{i}=@(s)G_hat(t(i),s);
205             end
206             % zero function for zbar
207             z_bar=@(s) 0;
208             [zprime_hat,zprime_hat_vec]=compute_zprime(G_hat_t,
                    inv_chol_P0,estimates.b,t,z_bar,'related');
209             varargout{1}=G_hat;
210     end
211     end
212
213     function g=ortho(t,P0,covtype)
214     % Orthonormalize the sections P_0t
215     P0_t=cell(length(t),1);
216     g=cell(length(t),1);
217     [T,S]=meshgrid(t,t);
218     switch covtype
219         case 'covfn'
220             P0_matrix=P0(T,S);
221         case 'regression'
222             P0_matrix=P0(T,S);
223             [V,D]=eig(P0_matrix);
224             [tmp,~]=find(abs(diag(D))<10^-15);
225             D_new=D;
226             for i=1:length(tmp)
227                 D_new(tmp(i),tmp(i)) = 10^-15;
228             end
229             P0_matrix=V*D_new*inv(V);
230     end
231     chol_P0=chol(P0_matrix,'lower');
232     inv_chol_P0=chol_P0\eye(length(t));
233     %create a cell array of sections of P0
234     for i=1:length(t)
235         P0_t{i}=@(s)P0(t(i),s);
236     end
237     %orthonormalize P0 to calculate g
```

```
238 for j=1:length(t)
239     g{j}=compute_lincomb(P0_t,inv_chol_P0(j,:));
240 end
241 end
242
243 function [P]=compute_P(t,P0,a,g)
244 %Compute P=P0+sum ak gk*gk
245 P=@(s,t)0;
246 for i=1:length(t)
247     f=g{i};
248     if a(i)==0
249         continue
250     else
251         P=@(s,t)P(s,t)+a(i).*f(s).*f(t);
252     end
253 end
254 P=@(s,t)P0(s,t)+P(s,t);
255 end
256
257 function g=compute_lincomb(X,v)
258 %computes linear combinations of the form sum(v_i*X_i)
259 %where X is a cell array of anonymous functions
260 g=@(s)0;
261 for i=1:length(v)
262     if v(i)==0
263         continue
264     else
265         g=@(s)(g(s)+v(i).*X{i}(s));
266     end
267 end
268 end
269
270 function M=makeSymmetric(M,~)
271 %remove rounding error from P_matrix to make P_matrix symmetric
272 issym=@(x) all(all(x==x.'));
273 i=20;
274 while ~issym(M)
275     Ndecimals = i ;
276     H = 10.^Ndecimals;
277     M = round(H*M)/H;
278
279     % M=round(M,i);
280     i=i-1;
281 end
282 end
```

```
283
284 function a_hat=estimate_a(Z,chol_P0,dn)
285 %estimates ahat+1
286 %Z_i must be in columns
287 U=chol_P0\Z; %U_k is in the kth row
288 U_sq=sum(U.^2,2);
289 a_hat=[(U_sq(1:dn)); zeros(numel(U_sq)-dn,1)];
290 end
291
292 function c_hat=estimate_c(Z,W,chol_P0,n,dn)
293 U=chol_P0\Z; %U_k is in the kth row
294 U_bar=mean(U,2);
295 W_U=U*W./n;
296 w_U=W_U./mean(W);
297 c_hat=w_U-U_bar;
298 c_hat=[c_hat(1:dn); zeros(numel(U_bar)-dn,1)];
299
300 end
301
302 function [ b_hat ] = estimate_b(a_hat,c_hat)
303 b_hat=c_hat./(a_hat+1);
304
305 end
306
307 function T= transform(n)
308 T = gallery('orthog',n,4)';
309 return;
310 end
311
312
313 function [zbar,zbar_vec]=compute_zbar(alpha,g,t)
314 zbar=compute_lincomb(g,alpha);
315 zbar_vec=zbar(t);
316 end
317
318 function [a_hat, b_hat,c_hat,alpha_hat ] = estimate_b_ind( Z,W,
      chol_P0,n,dn)
319 %Estimates the vectors a,b,c and alpha
320 % chol_P0*U=Z
321 U=chol_P0\Z; %U_k is in the kth row
322 U_bar=mean(U,2);
323 U_Ubar=zeros(size(U)); %Uk-Ubar
324 for i=1:n
325     U_Ubar(:,i)=U(:,i)-U_bar;
326 end
```

```
% U_sq=sum(U_Ubar.^2,2)./n;
U_sq=sum(U_Ubar.^2,2)./(n-1);
a_hat=[(U_sq(1:dn)-1);zeros(numel(U_sq)-dn,1)];
% ----
w_hat=W./mean(W);
c_hat=U_Ubar*w_hat./n;
% W_U=U*W./n;
% w_U=W_U./mean(W);
% c_hat=w_U-U_bar;
c_hat=[c_hat(1:dn);zeros(numel(U_sq)-dn,1)];
b_hat=[c_hat(1:dn)./U_sq(1:dn);zeros(numel(U_sq)-dn,1)];
alpha_hat=[U_bar(1:dn);zeros(numel(U_sq)-dn,1)];
end

%%
function [zprime,zprime_vec]=compute_zprime(G_t,inv_chol_P0,b,t,zbar,
    related)
%compute/estimate zprime
%G_t = cell array of symbolic functions of sections of the genetic
    covariance function
%b = coefficients of selection gradient
%zbar = mean process zbar(t) - symbolic function
n_t=size(G_t,1);
gamma_G=cell(n_t,1);
gamma_G_vec=zeros(n_t,length(t));
if size(t,1)>1
    t=t';
end
for j=1:n_t
    gamma_G{j}=compute_lincomb(G_t,inv_chol_P0(j,:));
    gamma_G_vec(j,:)=G_t{j}(t);
end
switch related
    case 'unrelated'
        zprime=compute_lincomb(gamma_G,b);
        zprime=@(s)zprime(s)+zbar(s);
        zprime_vec=zprime(t)';
    case 'related'
        zprime=compute_lincomb(gamma_G,b);
        zprime_vec=b'*inv_chol_P0*gamma_G_vec;
end
end
```

B. Orstein-Uhlenbeck and Wiener

```
clear variables
close all
clc

%% Description: Estimation of the selection gradient

% The MATLAB code for the estimations is based on the MATLAB script 'selectgrad.m' by Tyler Baur's, PhD at the University of Wisconsin - Milwaukee, with modifications and improvements by me (documented in 'select_grad.m')

% The goal of this test is to use the theory and methods described to compute estimates (functions) for the pre-selection mean, the mean of the trait among newborns of the next generation and the covariance function. (The estimate for the selection gradient itself is not given as it is not a practical result by its own.) During the process the observations are assumed to be
%       1 independent
%       2 dependent

% Data of Tribolium larvea for generations 0,1,2,3,4 (and 4.2) given by Carter and Irwin. Data has been cleaned using Microsoft Excel before import into MATLAB. Individuals with incomplete data were removed. The biggest subset of individuals with measurements (DSH1, DSH2,...) at the same age was chosen. Due to sample size restrictions, it was decided to include observations with slightly different ages at measurements, e.g. all observations of larvea where the first measurement (DSH1) was conducted at age 1 and 2 (age in days) were included and a weighted average was assigned as the new age at first measurement (DSH1_new).

% For more detailed information on the data set, see the included .rtf document 'trib_dataset_description.rtf'.

% Note: All denotations "GenX", X=0,1,2,3 refer to the generation used for estimation and NOT necessarly to the generation the function describes. E.g. zprime_hat_gen1 is the estimate of the mean among newborns of the offspring generation (generation 2) using generation 1 as underlying data for the estimation.

%% Candidates for covariance functions
% Use Ornstein-Uhlenbeck and Wiener covariance function as candidate for the covariance function P and the additive genetic covariance function G.

```

```
22 G_cand=cell(3,1);
23 cov_name=cell(3,1);
24
25 % Ornstein-Uhlenbeck covariance
26 G_cand{1} = @(s,t) exp(-abs(s-t));
27 cov_name{1} = 'O-U';
28
29 % Wiener covariance
30 G_cand{2} = @(s,t) min(s,t);
31 cov_name{2} = 'Wiener';
32
33 % Wiener covariance 2
34 G_cand{3} = @(s,t) 10^-2*min(s,t);
35 cov_name{3} = 'Wiener 2';
36
37 %% Estimation using Genereation 0
38 % ----------------------------------------
39 %% Import Data (Parent generation)
40 %
41 % Z: N-by-T matrix containing the weight of the Tribolium larvae,
       where N is the number of observations and T the number of
       measurements
42 % DSH: N-by-M matrix containing the respective days of measurements.
       The k-th column contains the day of each organism's k-th
       measurement.
43 % n: vector of family sizes, still contains 0s that need to be
       removed.
44
45 [~, ~, raw] = xlsread('Sel_Gen0Data_dryad_cleaned.xls','estimation');
46 n = cell2mat(raw(2:end,9));
47 n=n(n~=0);
48 DSH=cell2mat(raw(2:end,10:15));
49 Z=cell2mat(raw(2:end,16:21));
50
51 % Log-transform data
52 Z_log=log(Z);
53
54 % Create vector weighted average of ages at which the first, second
       ,... measurements were taken
55 t=mean(DSH,1);
56
57 Z_tmp0=Z;
58 t_tmp0=t;
59
60 d=@(n) min(n);  % sieve
```

```
n_obs=size(Z,1);

% Plot data
figure
subplot(1,2,1)
plot(t,Z_log)
% axis([t(1) t(end) 0 4*10^-3])
title('Gen0: Data (Log-transformed)')
subplot(1,2,2)
plot(t,Z)
% axis([t(1) t(end) -11 -5])
title('Gen0: Data')

%% Import Data (Offspring generation)
[~, ~, raw2] = xlsread('Sel_Gen1Data_dryad_cleaned.xlsx','estimation');
DSH=cell2mat(raw2(2:end,11:16));
t_nextgen=mean(DSH,1);
Z_nextgen=cell2mat(raw2(2:end,17:22));
Z_nextgen_log=log(Z_nextgen);
Z_nextgen_log_mean=mean(Z_nextgen_log,1);

%% Clear temporary variables
clearvars raw raw2;

%% Check import data Z and vector of family sizes n before estimation
if (size(Z,1)~=sum(n))
    error(['Error in data Z or vector n. Number of observations does not match vector of family sizes'])
end

%% Compute fitness function: Directional selection, W=exp(X)

% Choices of f
% f(t) = 1
f=ones(1,length(t));

% f(t) = t
% f=t;

% Method: Integral
% for i=1:n_obs
%    X(i)=trapz(t,Z(i,:).*f);
% end

```

```
104 % Method: Sum
105 X=Z*f';
106 W=exp(X);
107
108 % for i=1:2
109 %% Independent Case
110 % Assumption: Sample of unrelated organisms. Family sizes are
        irrelevant, only the number of observations is needed.
111
112 [estimates_gen0_ou,zprime_hat_gen0_ou,zprime_hat_vec_gen0_ou,
        P_hat_gen0_ou,zbar_hat_gen0_ou,zbar_hat_vec_gen0_ou]=selectgrad(
        Z_log,W,d,n_obs,t,'unrelated','covfn',G_cand{1},G_cand{1})
113 [estimates_gen0_wi,zprime_hat_gen0_wi,zprime_hat_vec_gen0_wi,
        P_hat_gen0_wi,zbar_hat_gen0_wi,zbar_hat_vec_gen0_wi]=selectgrad(
        Z_log,W,d,n_obs,t,'unrelated','covfn',G_cand{2},G_cand{2})
114 % [estimates_gen0_wi2,zprime_hat_gen0_wi2,zprime_hat_vec_gen0_wi2,
        P_hat_gen0_wi2,zbar_hat_gen0_wi2,zbar_hat_vec_gen0_wi2]=selectgrad
        (Z_log,W,d,n_obs,t,'unrelated','covfn',G_cand{3},G_cand{3})
115
116 %% Dependent Case
117 % Observations are structured in independent families of organisms
        with the same relation. Observations are full siblings giving the
        the relationship coefficient 0.5. The Wiener covariance is used
        as a candidate for the environmental covariance function.
118
119 E0 = @(s,t) 10^-3*min(s,t);
120 num_fams=length(n);
121 reln=0.5;
122
123 [estimates2_gen0_ou,zprime_hat2_gen0_ou,zprime_hat_vec_2_gen0_ou,
        P_hat2_gen0_ou,G_hat2_gen0_ou]=selectgrad(Z_log,W,d,n,t,'related',
        'covfn',G_cand{1},E0,num_fams,reln)
124 [estimates2_gen0_wi,zprime_hat2_gen0_wi,zprime_hat_vec_2_gen0_wi,
        P_hat2_gen0_wi,G_hat2_gen0_wi]=selectgrad(Z_log,W,d,n,t,'related',
        'covfn',G_cand{2},E0,num_fams,reln)
125 % [estimates2_gen0_wi2,zprime_hat2_gen0_wi2,zprime_hat_vec_2_gen0_wi2
        ,P_hat2_gen0_wi2,G_hat2_gen0_wi2]=selectgrad(Z_log,W,d,n,t,'
        related','covfn',G_cand{3},E0,num_fams,reln)
126
127 %% Plots
128 gray = 1/255*[200,200,200];
129 t_grid=linspace(t(1),t(end),1000);
130 [S,T]=meshgrid(t_grid,t_grid);
131 lw=1.1;
132
```

```
% Plots for Ornstein-Uhlenbeck
figure
subplot(1,2,1)
plot(t,Z_log,'Color',gray)
hold on
plot(t_grid,zbar_hat_gen0_ou(t_grid),'Color','r','LineWidth',lw)
title(['Gen0: zbarhat (independent, log-transf), ',cov_name{1}])
%% axis([t(1) t(end) -11 -2])
subplot(1,2,2)
plot(t,Z,'Color',gray)
hold on
plot(t_grid,exp(zbar_hat_gen0_ou(t_grid)),'Color','r','LineWidth',lw)
title(['Gen0: zbarhat (independent), ',cov_name{1}])
% axis([t(1) t(end) 0 80*10^-3])

figure
subplot(1,2,1)
plot(t_nextgen,Z_nextgen_log,'Color',gray)
hold on
plot(t_grid,zprime_hat_gen0_ou(t_grid),'Color','r','LineWidth',lw)
title(['Gen0: zprimehat (independent, log-transf), ',cov_name{1}])
% axis([t(1) t(end) -11 -2])
subplot(1,2,2)
plot(t_nextgen,Z_nextgen,'Color',gray)
hold on
plot(t_grid,exp(zprime_hat_gen0_ou(t_grid)),'Color','r','LineWidth',
    lw)
title(['Gen0: zprimehat (independent), ',cov_name{1}])
% axis([t(1) t(end) 0 80*10^-3])

figure
subplot(1,2,1)
surf(S,T,G_cand{1}(S,T),'LineStyle','none')
view(-15,15)
title(['Gen0: P0 (independent), ',cov_name{1}])
subplot(1,2,2)
surf(S,T,P_hat_gen0_ou(S,T),'LineStyle','none')
view(-15,15)
title(['Gen0: Phat (independent), ',cov_name{1}])

t_grid2=linspace(0,t(end),1000);
[S2,T2]=meshgrid(t_grid2,t_grid2);
figure
subplot(1,2,1)
surf(S2,T2,G_hat2_gen0_ou(S2,T2),'LineStyle','none')
```

```
177 view(30,15)
178 title(['Gen0: Ghat (dependent), ',cov_name{1}])
179 subplot(1,2,2)
180 surf(S2,T2,P_hat2_gen0_ou(S,T),'LineStyle','none')
181 view(30,15)
182 title(['Gen0: Phat (dependent), ',cov_name{1}])
183
184 % Plots for Wiener
185 figure
186 subplot(1,2,1)
187 plot(t,Z_log,'Color',gray)
188 hold on
189 plot(t_grid,zbar_hat_gen0_wi(t_grid),'Color','b','LineWidth',lw)
190 title(['Gen0: zbarhat (independent, log-transf), ',cov_name{2}])
191 % axis([t(1) t(end) -11 -5])
192 subplot(1,2,2)
193 plot(t,Z,'Color',gray)
194 hold on
195 plot(t_grid,exp(zbar_hat_gen0_wi(t_grid)),'Color','b','LineWidth',lw)
196 title(['Gen0: zbarhat (independent), ',cov_name{2}])
197 % axis([t(1) t(end) 0 4*10^-3])
198
199 figure
200 subplot(1,2,1)
201 plot(t_nextgen,Z_nextgen_log,'Color',gray)
202 hold on
203 plot(t_grid,zprime_hat_gen0_wi(t_grid),'Color','b','LineWidth',lw)
204 title(['Gen0: zprimehat (independent, log-transf), ',cov_name{2}])
205 % axis([t(1) t(end) -11 -5])
206 subplot(1,2,2)
207 plot(t_nextgen,Z_nextgen,'Color',gray)
208 hold on
209 plot(t_grid,exp(zprime_hat_gen0_wi(t_grid)),'Color','b','LineWidth',
      lw)
210 title(['Gen0: zprimehat (independent), ',cov_name{2}])
211 % axis([t(1) t(end) 0 4*10^-3])
212
213 figure
214 subplot(1,2,1)
215 surf(S,T,G_cand{2}(S,T),'LineStyle','none')
216 view(-15,15)
217 title(['Gen0: P0 (independent), ',cov_name{2}])
218 subplot(1,2,2)
219 surf(S,T,P_hat_gen0_wi(S,T),'LineStyle','none')
220 view(-15,15)
```

```
221 title(['Gen0: Phat (independent), ',cov_name{2}])
222
223 t_grid2=linspace(t(1)+0.3,t(end),1000);
224 [S2,T2]=meshgrid(t_grid2,t_grid2);
225 figure
226 subplot(1,2,1)
227 surf(S2,T2,G_hat2_gen0_wi(S2,T2),'LineStyle','none')
228 view(30,15)
229 title(['Gen0: Ghat (dependent), ',cov_name{2}])
230 subplot(1,2,2)
231 surf(S2,T2,P_hat2_gen0_wi(S2,T2),'LineStyle','none')
232 view(30,15)
233 title(['Gen0: Phat (dependent), ',cov_name{2}])
234
235 %% Plots for Wiener 2
236 % figure
237 % subplot(1,2,1)
238 % plot(t,Z_log,'Color',gray)
239 % hold on
240 % plot(t_grid,zbar_hat_gen0_wi2(t_grid),'Color','b','LineWidth',lw)
241 % title(['Gen0: zbarhat (independent, log-transf), ',cov_name{3}])
242 %% axis([t(1) t(end) -11 -5])
243 % subplot(1,2,2)
244 % plot(t,Z,'Color',gray)
245 % hold on
246 % plot(t_grid,exp(zbar_hat_gen0_wi2(t_grid)),'Color','b','LineWidth',
      lw)
247 % title(['Gen0: zbarhat (independent), ',cov_name{3}])
248 %% axis([t(1) t(end) 0 4*10^-3])
249 %
250 % figure
251 % subplot(1,2,1)
252 % plot(t_nextgen,Z_nextgen_log,'Color',gray)
253 % hold on
254 % plot(t_grid,zprime_hat_gen0_wi2(t_grid),'Color','b','LineWidth',lw)
255 % title(['Gen0: zprimehat (independent, log-transf), ',cov_name{3}])
256 %% axis([t(1) t(end) -11 -5])
257 % subplot(1,2,2)
258 % plot(t_nextgen,Z_nextgen,'Color',gray)
259 % hold on
260 % plot(t_grid,exp(zprime_hat_gen0_wi2(t_grid)),'Color','b','LineWidth
      ',lw)
261 % title(['Gen0: zprimehat (independent), ',cov_name{3}])
262 %% axis([t(1) t(end) 0 4*10^-3])
263 %
```

```
% figure
% subplot(1,2,1)
% surf(S,T,G_cand{3}(S,T),'LineStyle','none')
% view(-15,15)
% title(['Gen0: P0 (independent), ',cov_name{3}])
% subplot(1,2,2)
% surf(S,T,P_hat_gen0_wi2(S,T),'LineStyle','none')
% view(-15,15)
% title(['Gen0: Phat (independent), ',cov_name{3}])
%
% t_grid2=linspace(t(1)+0.3,t(end),1000);
% [S2,T2]=meshgrid(t_grid2,t_grid2);
% figure
% subplot(1,2,1)
% surf(S2,T2,G_hat2_gen0_wi2(S2,T2),'LineStyle','none')
% view(30,15)
% title(['Gen0: Ghat (dependent), ',cov_name{3}])
% subplot(1,2,2)
% surf(S2,T2,P_hat2_gen0_wi2(S2,T2),'LineStyle','none')
% view(30,15)
% title(['Gen0: Phat (dependent), ',cov_name{3}])

%% Figure comparing zbar OU to Wiener, log-trans
figure
subplot(1,2,1)
plot(t,Z_log,'Color',gray)
hold on
plot(t_grid,zbar_hat_gen0_ou(t_grid),'Color','r','LineWidth',lw)
title(['Gen0: zbarhat (independent, log-transf), ',cov_name{1}])
subplot(1,2,2)
plot(t,Z_log,'Color',gray)
hold on
plot(t_grid,zbar_hat_gen0_wi(t_grid),'Color','b','LineWidth',lw)
title(['Gen0: zbarhat (independent, log-transf), ',cov_name{2}])

% Comparisons
figure
subplot(1,2,1)
plot(t,Z,'Color',gray)
hold on
plot(t_grid,exp(zbar_hat_gen0_ou(t_grid)),'Color','r','LineWidth',lw)
% axis([t(1) t(end) 0 80*10^-3])
title(['Gen0: zbarhat (independent), ',cov_name{1}])
subplot(1,2,2)
plot(t,Z,'Color',gray)
```

```
hold on
plot(t_grid,exp(zbar_hat_gen0_wi(t_grid)),'Color','b','LineWidth',lw)
title(['Gen0: zbarhat (independent), ',cov_name{2}])
% axis([t(1) t(end) 0 4*10^-3])

figure
subplot(2,2,1)
surf(S,T,G_cand{1}(S,T),'LineStyle','none')
view(-15,15)
title(['Gen0: P0, ',cov_name{1}])
subplot(2,2,2)
surf(S,T,P_hat_gen0_ou(S,T),'LineStyle','none')
view(-15,15)
title(['Gen0: Phat, ',cov_name{1}])
subplot(2,2,3)
surf(S,T,G_cand{2}(S,T),'LineStyle','none')
view(-15,15)
title(['Gen0: P0, ',cov_name{2}])
subplot(2,2,4)
surf(S,T,P_hat_gen0_wi(S,T),'LineStyle','none')
view(-15,15)
title(['Gen0: Phat, ',cov_name{2}])

% figure
% subplot(1,3,1)
% surf(S,T,P_hat_gen0_ou(S,T),'LineStyle','none')
% title(['Gen0: Phat, ',cov_name{1}])
% view(-15,15)
% subplot(1,3,2)
% surf(S,T,P_hat_gen0_wi(S,T),'LineStyle','none')
% title(['Gen0: Phat, ',cov_name{2}])
% view(-15,15)
% subplot(1,3,3)
% surf(S,T,P_hat_gen0_wi2(S,T),'LineStyle','none')
% title(['Gen0: Phat, ',cov_name{3}])
% view(-15,15)

fig=sort(get(0,'children'));
for i=1:length(fig)
    saveas(fig(i), ['figure' num2str(i)], 'epsc');
end
% clearvars -except G_cand cov_name gray lw reln E0
```

C. Fitting Orthonormal Functions

C.1. Test Script

```
clear variables
close all
clc

% Angles for view in plots
az=-10;
ez=40;
%% Test: Estimate G for Generation 0
% Kirkpatrick, Lofsvold and Bulmer give a method to estmate the
    additive
% covariance function G for infinite-dimensional traits

%% Import Data: Generation 0
[~, ~, raw] = xlsread('Sel_Gen0Data_dryad_cleaned.xls','estimation');
DSH=cell2mat(raw(2:end,10:15));
Z=cell2mat(raw(2:end,16:21));
% Log-transform data
Z_log = log(Z);
% Weighted average for time points of (DSH1,...,DSH6)
t = mean(DSH,1);
% Number of time points
n_t = length(t);

%% Estimate covariance G_hat
G_hat_data = cov(Z_log);
tmp=repmat(t',1,n_t);
y0=tmp(:);
tmp=tmp';
x0=tmp(:);
z0=G_hat_data(:);
figure
scatter3(x0,y0,z0,'MarkerFaceColor',[0 .75 .75])
view(az,ez)
title('Gen0: Sample covariance')

%% Normalized Legendre polynomials
p_nleg=compute_legendre(n_t);

%% Compute matrix Phi
% Adjust/scale time point vector t to the domain of the Legendre
% polynomials
```

```
u = -1;
v = 1;
t_adj = adjust(t,t,u,v);
Phi=zeros(n_t,n_t);
for i=1:n_t
    for j=1:n_t
        Phi(i,j) = p_nleg{j}(t_adj(i));
    end
end

%% Compute Coefficient matrix C_G_hat
C_G_hat = inv(Phi)*G_hat_data*inv(Phi');

%% Compute estimate for additive covariance function G for x,y in [t
    (1),t(end)]
G_hat_gen0=compute_G_hat(C_G_hat,n_t,p_nleg,u,v,t);

n_grid=50;
t_grid=linspace(t(1),t(end),n_grid);
[S0,T0]=meshgrid(t_grid,t_grid);
figure
surf(S0,T0,G_hat_gen0(S0,T0),'LineStyle','none')
view(az,ez)
hold on
scatter3(x0,y0,z0,'MarkerFaceColor',[0 .75 .75])

% Extra plot
figure
subplot(1,2,1)
scatter3(x0,y0,z0,'MarkerFaceColor',[0 .75 .75])
view(-10,20)
title('Gen0: Sample covariance')
subplot(1,2,2)
surf(S0,T0,G_hat_gen0(S0,T0),'LineStyle','-.')
view(-10,20)
hold on
scatter3(x0,y0,z0,'MarkerFaceColor',[0 .75 .75])
title('Gen0: Ghat')

%% Test selectgrad.m for Gen0: (G_hat might not be positive definite,
    in this case OK)
[~, ~, raw2] = xlsread('Sel_Gen1Data_dryad_cleaned.xlsx','estimation'
    );
DSH=cell2mat(raw2(2:end,11:16));
t_nextgen=mean(DSH,1);
```

```
83  Z_nextgen=cell2mat(raw2(2:end,17:22));
84
85  n = cell2mat(raw(2:end,9));
86  n=n(n~=0);
87
88  f=ones(1,length(t));
89  X=Z*f';
90  W=exp(X);
91
92  d=@(n) min(n);
93  n_obs=size(Z,1);
94
95  [estimates_gen0_leg,zprime_hat_gen0_leg,zprime_hat_vec_gen0_leg,
        P_hat_gen0_leg,zbar_hat_gen0_leg,zbar_hat_vec_gen0_leg]=selectgrad
        (Z_log,W,d,n_obs,t,'unrelated','covfn',G_hat_gen0,G_hat_gen0)
96
97  E0 = @(s,t) min(s,t);
98  num_fams=length(n);
99  reln=0.5;
100 [estimates2_gen0_leg,zprime_hat2_gen0_leg,zprime_hat_vec_2_gen0_leg,
        P_hat2_gen0_leg,G_hat2_gen0_leg]=selectgrad(Z_log,W,d,n,t,'related
        ','covfn',G_hat_gen0,E0,num_fams,reln)
101
102 % Plots
103 gray = 1/255*[200,200,200];
104 t_grid_tmp=linspace(t(1),t(end),1000);
105 % [S,T]=meshgrid(t_grid_tmp,t_grid_tmp);
106 lw=1.1;
107
108 figure
109 subplot(1,2,1)
110 % plot(t,Z_log,'Color',gray)
111 % hold on
112 % plot(t_grid_tmp,zbar_hat_gen0_leg(t_grid_tmp),'Color','m','
        LineWidth',lw)
113 % title('Gen0: zbarhat (independent, log-transf), Legendre ')
114 % subplot(1,2,2)
115 plot(t,Z,'Color',gray)
116 hold on
117 plot(t_grid_tmp,exp(zbar_hat_gen0_leg(t_grid_tmp)),'Color','m','
        LineWidth',lw)
118 title('Gen0: zbarhat (independent), Legendre')
119
120 t_grid_tmp_2=linspace(t_nextgen(1),t_nextgen(end),1000);
121 % figure
```

```
% subplot(1,2,1)
% plot(t_nextgen,Z_nextgen_log,'Color',gray)
% hold on
% plot(t_grid_tmp_2,zprime_hat_gen0_leg(t_grid_tmp_2),'Color','m','
    LineWidth',lw)
% title('Gen0: zprimehat (independent, log-transf), Legendre ')
subplot(1,2,2)
plot(t_nextgen,Z_nextgen,'Color',gray)
hold on
plot(t_grid_tmp_2,exp(zprime_hat_gen0_leg(t_grid_tmp_2)),'Color','m',
    'LineWidth',lw)
title('Gen0: zprimehat (independent), Legendre')

figure
subplot(1,2,1)
surf(S0,T0,G_hat_gen0(S0,T0),'LineStyle','-.')
view(az,ez)
title('Gen0: P0 (Legendre)')
hold on
scatter3(x0,y0,z0,'MarkerFaceColor',[0 .75 .75])
subplot(1,2,2)
surf(S0,T0,P_hat_gen0_leg(S0,T0),'LineStyle','-.')
view(az,ez)
title('Gen0: Phat (Legendre)')

figure
subplot(1,2,1)
surf(S0,T0,G_hat_gen0(S0,T0),'LineStyle','-.')
view(az,ez)
title('Gen0: P0 (Legendre)')
hold on
scatter3(x0,y0,z0,'MarkerFaceColor',[0 .75 .75])
subplot(1,2,2)
surf(S0,T0,P_hat2_gen0_leg(S0,T0),'LineStyle','-.')
view(az,ez)
title('Gen0: Phat (dependent, Legendre)')

% clear variables
data_tmp=cell(5,2);
data_tmp{1,1}=t;
data_tmp{1,2}=Z;

fig=sort(get(0,'children'));
for i=1:length(fig)
    saveas(fig(i), ['figure' num2str(i)], 'epsc');
```

```
end
```

C.2. Additional Functions

```
%% Compute normalized Legendre polynomials
function p_nleg=compute_legendre(n)
p_nleg=cell(n,1);
for j=0:n-1
    p_nleg{j+1}=@(x) 0;
    for k=0:floor(j/2)
        p_nleg{j+1}=@(x) p_nleg{j+1}(x)+(-1)^k*nchoosek(j,k)*nchoosek
            (2*j-2*k,j)*x.^(j-2*k);
    end
    p_nleg{j+1}=@(x)(1/2)^j*sqrt((2*j+1)/2)*p_nleg{j+1}(x);
end
```

```
%% Adjust point in time between first and last time point of
    measurement to a given range [u,v] (e.g. the domain of legendre
    polynomials [-1,1])
function s_adj=adjust(s,t,u,v)
    s_adj = u + (v-u)/(t(end)-t(1))*(s-t(1));
end
```

```
%% Compute estimate for additive-genetic covariance function G
function G_hat=compute_G_hat(C_G_hat,n_t,p_nleg,u,v,t)
    G_hat=@(x,y) 0;
    for i=1:n_t
        for j=1:n_t
            G_hat=@(x,y) G_hat(x,y)+C_G_hat(i,j).*p_nleg{i}(adjust(x,
                t,u,v)).*p_nleg{j}(adjust(y,t,u,v));
        end
    end
end
```

D. Adjusted Ornstein-Uhlenbeck and Wiener

```
clear variables
close all
clc
%% Candidates for covariance functions
% Use Ornstein-Uhlenbeck and Wiener covariance function as candidate
    for the covariance function P and the additive genetic covariance
    function G.

G_cand=cell(3,1);
cov_name=cell(3,1);
```

```
9
10 % Ornstein-Uhlenbeck covariance
11 G_cand{1} = @(s,t) exp(-0.25*abs(s-t));
12 cov_name{1} = 'O-U adj1';
13
14 G_cand{2} = @(s,t) 0.25*exp(-0.1*abs(s-t));
15 cov_name{2} = 'O-U adj2';
16
17 % Wiener covariance
18 G_cand{3} = @(s,t) 10^-2*min(s,t);
19 cov_name{3} = 'Wiener adj';
20
21 %% -2- Estimation using Genereation 0
22 % ------------------------------------
23 %% Import Data (Parent generation)
24 %
25 % Z: N-by-T matrix containing the weight of the Tribolium larvae,
       where N is the number of observations and T the number of
       measurements
26 % DSH: N-by-M matrix containing the respective days of measurements.
       The k-th column contains the day of each organism's k-th
       measurement.
27 % n: vector of family sizes, still contains 0s that need to be
       removed.
28
29 [~, ~, raw] = xlsread('Sel_Gen0Data_dryad_cleaned.xls','estimation');
30 n = cell2mat(raw(2:end,9));
31 n=n(n~=0);
32 DSH=cell2mat(raw(2:end,10:15));
33 Z=cell2mat(raw(2:end,16:21));
34
35 % Log-transform data
36 Z_log=log(Z);
37
38 % Create vector weighted average of ages at which the first, second
       ,... measurements were taken
39 t=mean(DSH,1);
40
41 Z_tmp0=Z_log;
42 t_tmp0=t;
43
44 d=@(n) min(n);  % sieve
45 n_obs=size(Z,1);
46
47 %% Import Data (Offspring generation)
```

```
[~, ~, raw2] = xlsread('Sel_Gen1Data_dryad_cleaned.xlsx','estimation');
DSH=cell2mat(raw2(2:end,11:16));
t_nextgen=mean(DSH,1);
Z_nextgen=cell2mat(raw2(2:end,17:22));
Z_nextgen_log=log(Z_nextgen);
Z_nextgen_log_mean=mean(Z_nextgen_log,1);

%% Clear temporary variables
clearvars raw raw2;

%% Check import data Z and vector of family sizes n before estimation
if (size(Z,1)~=sum(n))
    error(['Error in data Z or vector n. Number of observations does not match vector of family sizes'])
end

%% Compute fitness function: Directional selection, W=exp(X)

% Choices of f
% f(t) = 1
f=ones(1,length(t));

% f(t) = t
% f=t;

% Method: Integral
% for i=1:n_obs
%    X(i)=trapz(t,Z(i,:).*f);
% end

% Method: Sum
X=Z*f';
W=exp(X);

% for i=1:2
%% Independent Case
% Assumption: Sample of unrelated organisms. Family sizes are irrelevant, only the number of observations is needed.

% Note: variable naming can be confusing

[estimates_gen0_ou,zprime_hat_gen0_ou,zprime_hat_vec_gen0_ou,P_hat_gen0_ou,zbar_hat_gen0_ou,zbar_hat_vec_gen0_ou]=selectgrad(Z_log,W,d,n_obs,t,'unrelated','covfn',G_cand{1},G_cand{1})
```

```
88 [estimates_gen0_wi,zprime_hat_gen0_wi,zprime_hat_vec_gen0_wi,
       P_hat_gen0_wi,zbar_hat_gen0_wi,zbar_hat_vec_gen0_wi]=selectgrad(
       Z_log,W,d,n_obs,t,'unrelated','covfn',G_cand{2},G_cand{2})
89 [estimates_gen0_wi2,zprime_hat_gen0_wi2,zprime_hat_vec_gen0_wi2,
       P_hat_gen0_wi2,zbar_hat_gen0_wi2,zbar_hat_vec_gen0_wi2]=selectgrad
       (Z_log,W,d,n_obs,t,'unrelated','covfn',G_cand{3},G_cand{3})
90
91 %% Dependent Case
92 % Observations are structured in independent families of organisms
       with the same relation. Observations are full siblings giving the
       the relationship coefficient 0.5. The Wiener covariance is used
       as a candidate for the environmental covariance function.
93
94 E0 = @(s,t) 10^-3*min(s,t);
95 num_fams=length(n);
96 reln=0.5;
97
98 [estimates2_gen0_ou,zprime_hat2_gen0_ou,zprime_hat_vec_2_gen0_ou,
       P_hat2_gen0_ou,G_hat2_gen0_ou]=selectgrad(Z_log,W,d,n,t,'related',
       'covfn',G_cand{1},E0,num_fams,reln)
99 [estimates2_gen0_wi,zprime_hat2_gen0_wi,zprime_hat_vec_2_gen0_wi,
       P_hat2_gen0_wi,G_hat2_gen0_wi]=selectgrad(Z_log,W,d,n,t,'related',
       'covfn',G_cand{2},E0,num_fams,reln)
100 [estimates2_gen0_wi2,zprime_hat2_gen0_wi2,zprime_hat_vec_2_gen0_wi2,
       P_hat2_gen0_wi2,G_hat2_gen0_wi2]=selectgrad(Z_log,W,d,n,t,'related
       ','covfn',G_cand{3},E0,num_fams,reln)
101
102 %% Plots
103 gray = 1/255*[200,200,200];
104 t_grid=linspace(t(1),t(end),1000);
105 [S,T]=meshgrid(t_grid,t_grid);
106 lw=1.1;
107
108 figure
109 subplot(1,3,1)
110 plot(t,Z,'Color',gray)
111 hold on
112 plot(t_grid,exp(zbar_hat_gen0_ou(t_grid)),'Color','r','LineWidth',lw)
113 title(cov_name{1})
114 subplot(1,3,2)
115 plot(t,Z,'Color',gray)
116 hold on
117 plot(t_grid,exp(zbar_hat_gen0_wi(t_grid)),'Color','r','LineWidth',lw)
118 title(cov_name{2})
119 subplot(1,3,3)
```

```
120 plot(t,Z,'Color',gray)
121 hold on
122 plot(t_grid,exp(zbar_hat_gen0_wi2(t_grid)),'Color','b','LineWidth',lw
    )
123 title(cov_name{3})
124
125 figure
126 subplot(1,3,1)
127 surf(S,T,P_hat_gen0_ou(S,T),'LineStyle','none')
128 view(-15,15)
129 title(cov_name{1})
130 subplot(1,3,2)
131 surf(S,T,P_hat_gen0_wi(S,T),'LineStyle','none')
132 view(-15,15)
133 title(cov_name{2})
134 subplot(1,3,3)
135 surf(S,T,P_hat_gen0_wi2(S,T),'LineStyle','none')
136 view(-15,15)
137 title(cov_name{3})
138
139 %% Save figures
140 fig=sort(get(0,'children'));
141 for i=1:length(fig)
142     saveas(fig(i), ['adj' num2str(i)], 'epsc');
143 end
```