TOWARDS IMPROVEMENT OF **RUMI**NANT BREEDING
THROUGH **GEN**OMIC AND EPIGENOMIC APPROACHES

Grant agreement number: 101000226

H2020 – Research and Innovation Action

Deliverable 8.2

Report on novel statistical tools and breeding strategies for mitigating trade-offs between production, efficiency and resilience

Due date of deliverable: M30 – November 2023

Actual submission date: 13/12/23

WP concerned	WP8	
Lead partner	Mario CALUS	WU
Author(s)	Kareen ANENE Han MULDER Jérémie VANDENPLAS Beatriz CUYABANO	WU WU WR INRAE
Version	V1	
Dissemination level	PU: Public	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°101000226.

Call: H2020-SFS-2020-2

Topic: SFS-13-2020 ı Genome and epigenome enabled breeding in terrestrial livestock

Start date of the project: June 1st, 2021 **Duration:** 60 months

End date of the project: May 31st, 2026 **Project ID:** 101000226

Table of content

1. Summary

Objectives:

One of the main objectives of RUMIGEN is to provide statistical tools and breeding strategies for mitigating trade-offs between production, efficiency and resilience. The objectives of this deliverable 8.2 are to: (1) investigate the impact of the historic evolution of breeding goals on trade-offs between production, fertility, and resilience, (2) investigate possible benefits of using trade-offs at the genome level in balanced breeding strategies, and (3) develop optimal selection strategies to increase efficiency and resilience to climate change.

Rationale:

To achieve the first objective, we firstly developed theory to describe the expected evolution of genetic covariances and correlations between two traits that are under selection. Secondly, we empirically investigated the evolution of genetic variances of milk production traits and a fertility trait, as well as the evolution of the genetic covariances and correlations between them, in first-parity dairy cows during the last 30 years in France and the Netherlands. The breeds investigated were the Holstein breed, and two local breeds, Montbeliarde (France) and MRY (the Netherlands). Results showed that the evolution of estimates of observed genetic (co)variances and correlations between milk production traits and a fertility trait across birth years follow different trends for Dutch Holstein, French Montbeliarde and MRY first-parity cows. The different trends in genetic (co)variances and correlations could be due to either the Bulmer effect, as shown by the theoretical derivations, or to changes of national breeding goals. Therefore, further analyses will be needed to confirm these trends in each cow population.

To achieve the second objective, we wanted to make use of gametic Mendelian Sampling variances and covariances, which explain observed genetic variances within traits and covariances between traits among the offspring of individual animals. Given that individual differences in gametic Mendelian Sampling covariances have been observed, we hypothesized that using those in selection may reduce the magnitude of antagonistic genetic correlations. We directly simulated distributions of breeding value for two traits, and then performed selection based on: breeding values (following the current practice), or selection based on the so-called Index5, which in addition to the breeding values also included the individual gametic Mendelian Sampling variances and/or the individual gametic Mendelian Sampling covariances. Including Mendelian Sampling variances and covariances in selection resulted to up to 2% more genetic gain compared to ordinary selection on breeding values, while it was able to change the genetic correlation in the next generation from -0.3 to values ranging from -0.28 to -0.22, while ignoring the Mendelian Sampling covariance left the genetic correlation unaffected. These results provide a first indication that use of individual Mendelian Sampling covariances in selection decisions may help to mitigate the effect of antagonistic genetic correlations.

To achieve the third objective, we stochastically simulated a dairy cattle breeding program with two traits, reflecting conception rate and protein yield, based on genetic parameters estimated on Dutch dairy data in WP3. We simulated four scenarios: (1) heat stress was not present, and also not considered when estimating breeding values; (2) heat stress was present, but not considered when estimating breeding values; and finally, heat stress was present, and considered when estimating breeding values, while selection was either (3) based on breeding values corresponding to a situation without heat stress, or (4) based on breeding values corresponding to a situation with severe heat stress. Results showed that even when heat stress are included in the breeding value estimation, the overall sensitivity to heat stress for conception rate may still increase, as a correlated response

to selection for increased protein yield. Therefore, selecting animals that are less sensitive to heat stress requires that heat stress is considering in breeding value estimation, and that traits related to heat stress (e.g. conception rate and protein slope) are included in the breeding goal.

Teams involved:

The RUMIGEN partners that contributed to this Deliverable are WU, WR and INRAE. In addition, analyzed data was contributed by BDNZ (France) and CRV (the Netherlands).

2. Introduction

An important objective of RUMIGEN is to develop statistical tools and breeding strategies for mitigating trade-offs between production, efficiency and resilience.

The objectives of this deliverable 8.2 are to: 1) investigate the impact of the historic evolution of breeding goals on trade-offs between production, fertility, and resilience, 2) investigate possible benefits of using trade-offs at the genome level in balanced breeding strategies, and 3) develop optimal selection strategies to increase efficiency and resilience to climate change.

This deliverable report is organized as follows. Each of the three aims is described in a separate section, where each section starts with a brief description of the background, followed by descriptions of theory, used data, applied methods, and ends with a results & discussion subsection. The report finishes with a section with general conclusions across the three aims.

3. Impact evolution of breeding goals on trade-offs

3.1 Background

The evolution of genetic variance over generations is a topic that has been widely studied in quantitative genetics. In many breeding populations, both selection (resulting in the Bulmer effect (Bulmer, 1971)) and limited population sizes (which builds up co-ancestry, resulting in random drift and inbreeding) reduce a trait's genetic variance. Until recently most research related to this topic remained theoretical due to limitations of data availability with sufficient generations for many species. Furthermore, when it comes to the interest of genetic evaluations with a goal to rank selection candidates, changes in variance parameters due to selection can be disregarded without compromising the final rank of individual's genetic merit (Gianola et al., 2022).

However, if the goal is to define a breeding program by redesigning the breeding goals with a medium to long-term perspective, it is of paramount importance that variance parameters are adequately described and estimated at the current stage of the selected population, in order to build selection indexes that account for the current parameters. While for a genetic evaluation, the principle of the ignorable selection (Gianola and Fernando, 1986; Fernando and Gianola, 1990; Gianola et al., 2022) ensures adequate ranking at each trait of interest, for many generations of selected individuals, when it comes to building an individual's selection index based on multiple traits, special attention is drawn to the correct input of genetic correlations, which ensures a good compromise when selecting for negatively correlated traits.

Studies regarding the consequences of selection on the genetic correlation between traits (and on their evolution over generations) are limited both in the theoretical perspective and empirically (Cochran, 1951; Hill, 1965; Itoh, 1991; Villanueva et al., 1993; Sorensen et al., 2001; Hidalgo et al.,

2020). Moreover, until now those studies present limitations in their theoretical conclusions about the trends of genetic correlations in populations under selection. However, it is reasonable to expect that both the Bulmer effect and random drift impact genetic correlations between traits under selection. Furthermore, if changes are significant when compared to a base population (typically used for inference of genetic parameters), these changes should be accounted for when performing genetic evaluations comprising only the most recent generations of a breeding population.

A preliminary study on simulated data (Figure 1) that mimics a dairy cattle population with respect to production and fertility (i.e., heritability of 0.3 and of 0.01, respectively, and a genetic correlation between the two traits of -0.18) indicated that when the population evolved through 40 generations under random mating, although genetic variances present a minor decrease due to random drift, the genetic correlation remains unchanged. However, through 40 generations under selection for either one of the traits individually, the genetic correlation was significantly attenuated. This attenuation was systematically progressive while the Bulmer effect was present, most notably in the first five generations. Once the Bulmer effect was no longer present, the genetic correlation stabilized for a period in which random drift was not large enough to be perceived. Once the random drift becomes apparent in the population, the variance components will change, specifically presenting a reduction in their values. Because the variances present a smaller rate or reduction than that of the covariance, the result is a mild intensification of the genetic correlation. Nonetheless, the final correlation after 40 generations was attenuated to –0.08, much less strong than the original –0.18. Through 40 generations under selection for both traits with the same intensity, the genetic correlation was initially mildly attenuated during the first eight generations, and past that its value was greatly intensified, until reaching an apparent convergence around –0.57.

Based on this knowledge and on these first results, we first developed theory to describe the expected evolution of genetic covariances and correlations between two traits under selection. Second, we investigated the evolution of genetic variances of milk production traits and a fertility trait, as well as the evolution of the genetic covariances and correlations between them, in first-parity dairy cows during the last 30 years in France and the Netherlands. The breeds investigated in this study were a cosmopolitan breed, the Holstein breed, and two local breeds, Montbeliarde and MRY. Data for Montbeliarde were available in France, and data for Holstein and MRY were available in the Netherlands.

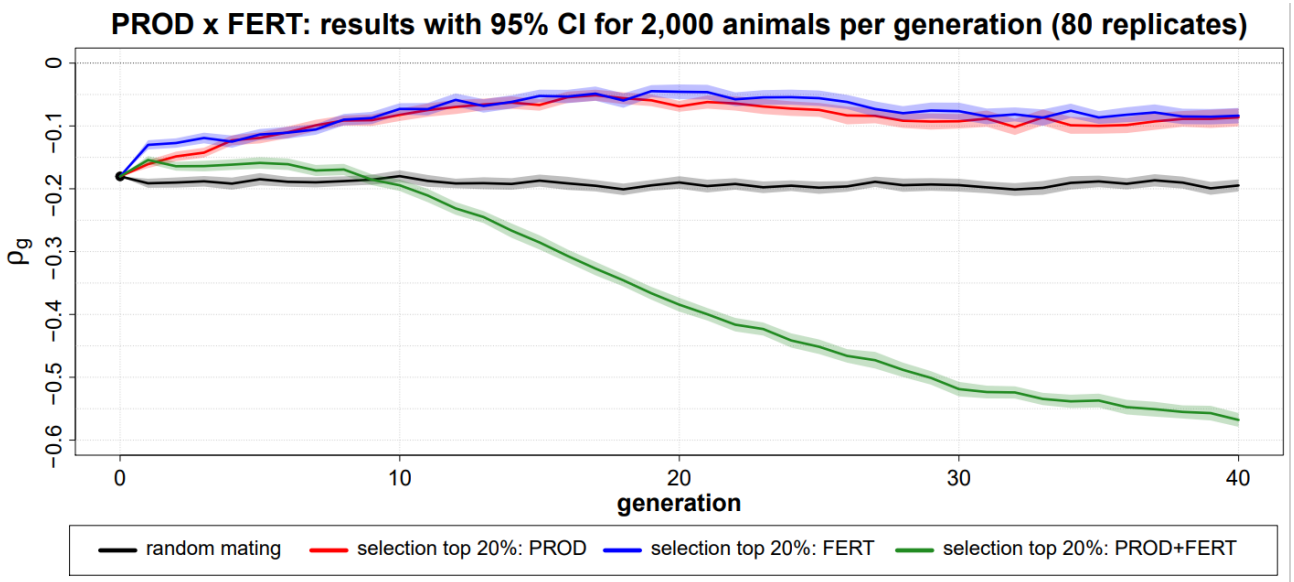

Figure 1. *Evolution of the genetic correlation between a production (PROD) and a fertility (FERT) trait, over forty generations starting from the same base population. Each scenario represents different selection criteria for the two traits (no selection, selection for a single trait, and selection for both traits).*

3.2 Theory

To accurately describe the trends expected for the genetic variances, covariances and correlations, over many generations of populations under selection, we have derived the theoretical calculus for the evolution of these parameters. Following Sorensen et al. (2001), we define $\sigma_{g_1}^{2(t)}$ and $\sigma_{g_2}^{2(t)}$ as the genetic variances of trait one (g_1) and two (g_2), respectively, at the t-th generation; $\sigma_{g_{12}}^{(t)}$ and $\rho^{(t)}$ are respectively the genetic covariance and correlation between the two traits at the same t-th generation. Now, let $\sigma_{g_1}^2 = \sigma_{g_1}^{2(0)}$, $\sigma_{g_2}^2 = \sigma_{g_2}^{2(0)}$,$\sigma_{g_{12}} = \sigma_{g_{12}}^{(0)}$, and $\rho = \rho^{(0)}$ be the variances, covariance, and correlation parameters at a base population, in which no selection has yet been applied. It is widely known that genetic variances are affected by both selection and random drift, and as per Sorensen et al. (2001),

$$\sigma_{g_1}^{2(t)} = Var\left(g_1^{(t)}\right) = \sigma_{g_1}^2\left(1 - \Delta_{1,sel}^{(t)} - \Delta_{drift}^{(t)}\right),$$

$$\sigma_{g_2}^{2(t)} = Var\left(g_2^{(t)}\right) = \sigma_{g_2}^2\left(1 - \Delta_{2,sel}^{(t)} - \Delta_{drift}^{(t)}\right),$$

in which the terms $\Delta^{(t)}$ represent the changes in $\sigma_{g_1}^2$ and $\sigma_{g_2}^2$ due to selection and random drift. It is reasonable to assume that the genetic covariance is also affected by both selection and random drift. Thus,

$$\sigma_{g_{12}}^{(t)} = Cov\left(g_1^{(t)}, g_2^{(t)}\right) = \sigma_{g_{12}}\left(1 - \Delta_{12,sel}^{(t)} - \Delta_{12,drift}^{(t)}\right).$$

We assume that the change in the covariance due to drift may differ from the change that the random drift imposes on the genetic variances.

Let **A** be the numerator relationship matrix, $\Delta_{drift}^{(t)} = E\left(A_{ij}^{(t)} + F_i^{(t)}\right)$ (Sorensen et al., 2001), such that the changes in the genetic variances due to the random drift can be estimated as follows:

$$D_{drift}^{(t)} = \overline{A}_{ij}^{(t)} + \overline{F}_i^{(t)}.$$

Using the moments from the truncated bivariate normal distribution, we can show that, at *t*=1:

$$\Delta_{1,sel}^{(1)} = \frac{\left(a_1\sigma_{g_1}^2 + a_2\sigma_{g_{12}}\right)^2}{\sigma_{g_1}^2\sigma_Y^2}\frac{\varphi(z)}{\alpha}\left(\frac{\varphi(z)}{\alpha} - z\right),$$

$$\Delta_{2,sel}^{(1)} = \frac{\left(a_2\sigma_{g_2}^2 + a_1\sigma_{g_{12}}\right)^2}{\sigma_{g_2}^2\sigma_Y^2}\frac{\varphi(z)}{\alpha}\left(\frac{\varphi(z)}{\alpha} - z\right),$$

$$\Delta_{12,sel}^{(1)} = \frac{\left(a_1\sigma_{g_1}^2 + a_2\sigma_{g_{12}}\right)\left(a_2\sigma_{g_2}^2 + a_1\sigma_{g_{12}}\right)}{\sigma_{g_{12}}\sigma_Y^2}\frac{\varphi(z)}{\alpha}\left(\frac{\varphi(z)}{\alpha} - z\right),$$

in which a_1 and a_2 are weights for traits 1 and 2, respectively, on the selection index, σ_Y^2 is the variance of the selection index $Y = a_1g_1 + a_2g_2$, $\varphi(z)$ is the density of the $N(0,1)$ at the point z, which satisfies $P(N(0,1) > z) = \alpha$, being α the proportion of selected individuals. Disregarding the random drift at this moment, the genetic correlation with respect to changes due to selection is given by:

$$\rho^{(1)} = \rho\frac{1-\Delta_{12,sel}^{(1)}}{\sqrt{\left(1-\Delta_{1,sel}^{(1)}\right)\left(1-\Delta_{2,sel}^{(1)}\right)}},$$

A functional analysis of $\rho^{(1)}$ with respect to ρ leads us to the following conclusions:

1. For $\rho \in \left(-1, -\sqrt{\frac{\min\{a_1,a_2\}}{\min\{a_1,a_2\}}}\right)$, $\rho^{(1)} > \rho$, and since ρ is negative, this correlation is attenuated;
2. For $\rho = -\sqrt{\frac{\min\{a_1,a_2\}}{\min\{a_1,a_2\}}}$, $\rho^{(1)} = \rho$;
3. For $\rho \in \left(-\sqrt{\frac{\min\{a_1,a_2\}}{\min\{a_1,a_2\}}}, 0\right)$, $\rho^{(1)} < \rho$, and since ρ is negative, this correlation is intensified;
4. For $\rho = 0$, $\rho^{(1)} < \rho$, becoming negative;
5. For $\rho \in (0, R)$, $\rho^{(1)} < \rho$, and we observe that the original $\rho > 0$ becomes negative;
6. For $\rho = R$,$\rho^{(1)} = 0$;
7. For $\rho \in (R, 1)$, $\rho^{(1)} < \rho$, and since ρ is positive, this correlation is attenuated while remaining positive.

The value of R, the point in which positive genetic correlations become negative is still under theoretical assessment to be described as a function of the parameters defining the selection index. Figure 2 presents a visual assessment of the conclusions listed above, including a medium-term projection of the trajectory of genetic correlations in populations under selection.

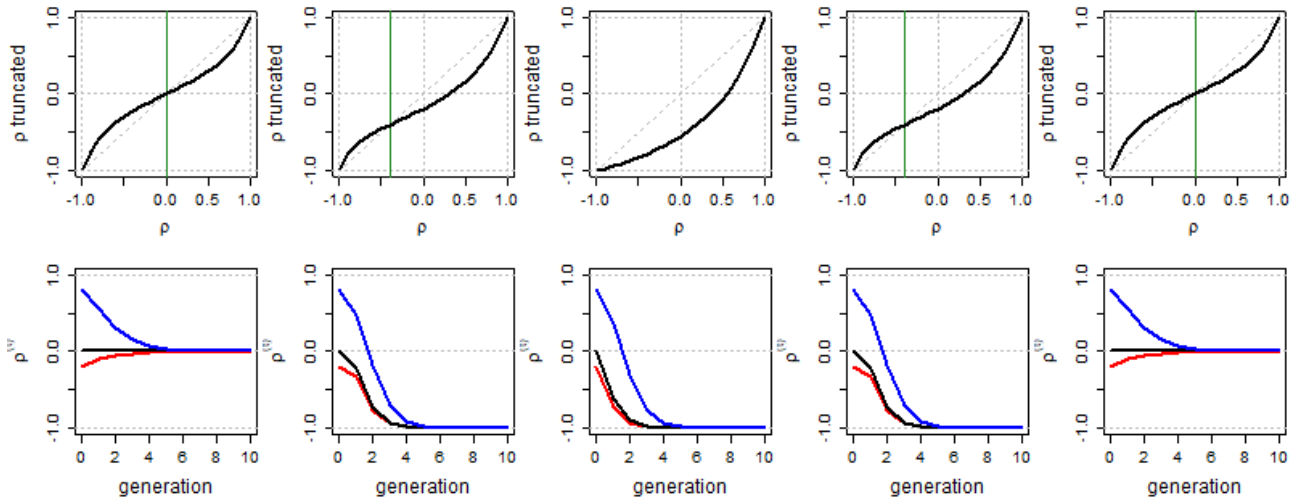

Figure 2. *Evolution of the genetic correlations following theoretical developments for different weights of dual selection for two correlated traits.*

These theoretical results align with the observed results obtained with simulations in our preliminary study (Figure 2), considering the population stages in which changes due to random drift are negligible.

Changes due to drift become relevant past the Bulmer effect takes place, i.e., once $\Delta_{1,sel}^{(t+1)} \approx \Delta_{1,sel}^{(t)}$, $\Delta_{2,sel}^{(t+1)} \approx \Delta_{2,sel}^{(t)}$, and $\Delta_{12,sel}^{(t+1)} \approx \Delta_{12,sel}^{(t)}$. Relying on simulation studies, we observed that $\Delta_{12,drift}^{(t)} > \Delta_{drift}^{(t)}$, i.e., that the random drift has a greater effect in the genetic covariance between traits, than it has on their genetic variances. Again, this result aligns with the observations in our preliminary study.

3.3 Data

Milk production traits included 305-day milk yield (MY, in kg), 305-day protein yield (PY, in kg) and 305-day fat yield (FY, in kg). Records of French Montbeliarde cows, of Dutch Holstein cows and of Dutch MRY cows, were extracted from the respective genetic evaluation data bases (BDNZ, France; and CRV, the Netherlands). It is worth noting that the records for the Netherlands were extracted for Dutch herds having a majority of cows genotyped.

For both countries, the extracted datasets used for this study cover a period of 30 years, starting from 1990 until 2020. Briefly, after editing, the datasets for the milk production traits (MY, PY, FY) included around 810 thousand records for first-parity French Montbeliarde cows, 855 thousand records for first-parity Dutch Holstein cows and 27 thousand records for first-parity Dutch MRY cows.

The fertility trait chosen to investigate the evolution of genetic (co)variances and correlations is the conception rate (CR) defined as a the success (0/1) of the first artificial insemination (AI) in first-parity cows. Similarly to the milk production traits, phenotypes were extracted from the French and the Dutch national data bases. Briefly, all datasets covered a period of 30 years, starting from 1990 until 2020. After editing, the datasets included around 810 thousand records for first-parity French Montbeliarde cows, 676 thousand records for first-parity Dutch Holstein cows and 20 thousand records for first-parity Dutch MRY cows. For all the analyses, pedigree information of phenotyped cows was traced back for 3 generations. A detailed description of the data and edits in each country and for each breed and trait, can be found in Appendix 1.

3.4 Models

All variance component estimations were based on pedigree BLUP animal models, and were performed for each breed and country separately.

3.4.1 Milk production traits

The linear model for the milk production traits (MY, FY, PY) of first-parity cows can be written as follows:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{Z}_a\mathbf{a} + \mathbf{e}_h$$

where $\mathbf{y}$, $\mathbf{b}$, $\mathbf{a}$, and $\mathbf{e}_h$ are the vectors of phenotypes, fixed effects, random additive genetic effects, and random residuals, respectively. The matrices $\mathbf{X}$ and $\mathbf{Z}_a$ are the incidence matrices for the corresponding effects. The residual variances are assumed to be heterogeneous and are defined by year of calving for the Dutch analyses. For the French analyses, the residual variances were also

assumed to be heterogeneous, however defined by the year of calving combined with the animal's herd.

The fixed effects were almost the same in both countries, and included:

- France: overall mean, herd-year of calving-season of calving, age at calving, and lactation length;
- The Netherlands: herd-year of calving, age at calving, year of calving-season of calving.

3.4.2 Conception rate

The linear model for conception rate of first-parity cows can be written as follows:

$$\mathbf{y} = \mathbf{Xb} + \mathbf{Z}_a\mathbf{a} + \mathbf{Z}_u\mathbf{u} + \mathbf{e}$$

where $\mathbf{y}$, $\mathbf{b}$, $\mathbf{a}$ and $\mathbf{e}$ are the vectors of phenotypes, fixed effects, random additive genetic effects, and random residuals, respectively. The vector $\mathbf{u}$ is the vector of random AI bull-year effects for the French analyses, and the vector of random effect for AI bull in the Dutch analyses. The matrices $\mathbf{X}$, $\mathbf{Z}_a$ and $\mathbf{Z}_u$ are the incidence matrices for the corresponding effects.

Finally, the residual variances were assumed to be homogenous for CR for the Dutch analyses and heterogenous for CR in the French analyses.

The fixed effects were almost the same in both countries, and included:

- France: overall mean, herd-year-season of calving, age at calving, interval between calving and AI, sexed semen (Y/N), AI operator, and week day of insemination;
- The Netherlands: herd-year of insemination, year-month of insemination, week day of insemination, age of bull, and sexed semen (Y/N).

3.5 Estimation of genetic (co)variances and correlations

We followed and extended the method presented in Sorensen et al. (2001) and Macedo et al. (2021) to estimate the trends in genetic (co)variances and correlations over time in both populations. In this study, we focused on the evolution of genetic (co)variances and correlations over time of first-parity cows with at least one record.

The computation of estimated genetic and residual (co)variances and estimated breeding values, was performed using Gibbs sampling with 150,000 iterations, a burn-in of 15,000, and saving samples each 150 iterations, totalizing 900 samples for computing the (co)variances. Due to computational issues, the results with the French data are preliminary, and based on a total of ~280 samples to compute the (co)variances. The results of the Gibbs sampling approach included estimates of variances at the base population, as well as a set of samples (stored at each 150 iterations) of EBVs of all animals. First-parity cows with records were grouped by year of birth, and (co)variances of the corresponding samples of EBVs for each group of cows were computed. Finally, estimated genetic (co)variances of each group of cows were computed as the mean of the (co)variances across all samples of EBVs associated with a specific group. Estimated genetic correlations were computed for each group using the averaged (co)variances. These estimated genetic (co)variances and correlations for each group of cows are hereafter referred to as estimates of observed genetic (co)variances and correlations.

3.6 Results & Discussion

Table 1 shows the estimated variances at the base population for MY and CR for the Dutch Holstein, French Montbeliarde and Dutch MRY populations. The estimated genetic variances at the base population for 305-day MY (posterior SD, in $(kg/100)^2$) were 70.78 (0.52) for Dutch Holstein, 32.56 (0.56) for French Montbeliarde, and 32.61 (1.39) for Dutch MRY. The estimated genetic variances at the base population for conception rate (posterior SD) were 79.24 (3.70) for Dutch Holstein, 3.49 (0.0084) for French Montbeliarde, and 87.3 (23.1) for Dutch MRY. The estimated variance for AI bull was 9.7 (0.8) for Dutch Holstein and 23.1 (7.5) for MRY (Table 1).

Table 1. *Estimated variances at the base population for 305-day milk yield (MY) and conception rate (CR). Posterior standard deviations are within brackets.*

Breed	Trait	Genetic variance[1]	Country-specific AI bull variance	Residual variance[1]
Holstein	MY	70.78 (0.52)	-	39.76 (6.00) – 93.33 (0.94)
	CR	79.24 (3.77)	9.71 (0.91)	2350.6 (4.94)
Montbeliarde	MY	32.56 (0.56)	-	75.97[2]
	CR	3.49 (0.008)	1.31(0.005)	17.9[2]
MRY	MY	32.61 (1.39)	-	22.56 (2.65) – 51.54 (3.71)
	CR	87.31 (21.71)	23.14 (7.54)	2353.8 (30.6)

[1]Genetic and residual variances for MY must be multiplied by 10,000.

[2]Average residual variance among heterogeneous groups

Figures 4, 5 and 6 show the evolution of genetic variances for MY and CR along birth year for Dutch Holstein first-parity cows, for French Montbeliarde first-parity cows, and for Dutch MRY first-parity cows, respectively. For all populations, and more specifically for Holstein and Montbeliarde, estimates of observed genetic variance by birth year for MY show an increasing trend along birth year. The rate is even larger in Holstein (Figure 4) and Montbeliarde (Figure 5) after the introduction of genomic selection in both populations (around 2008 in Holstein and around 2009 in Montbeliarde). Regarding CR, estimates of observed variances are similar along birth year for each population.

Additionally, Figures 4 and 6 also depict the estimates of genetic variances at the base population, and of expected genetic variances for the Dutch Holstein and MRY populations. Briefly, the difference between the genetic variance at the base population and the expected genetic variance is considered to be the reduction of genetic variance due to drift, and the difference between the expected genetic variance and the observed genetic variance is considered to be the reduction of genetic variance due to (pre)selection (Macedo et al., 2021). Therefore, it can be observed that the reduction of genetic variance for MY and CR for the Dutch Holstein cows, and for MY for MRY cows,

is mainly due to (pre)selection, and that the increase in inbreeding and relationships (Figure 14) explains a small proportion of the reduction of genetic variances for both traits. Macedo et al. (2021) also concluded that the reduction in genetic variance for MY in dairy sheep was mainly due to (pre)selection.

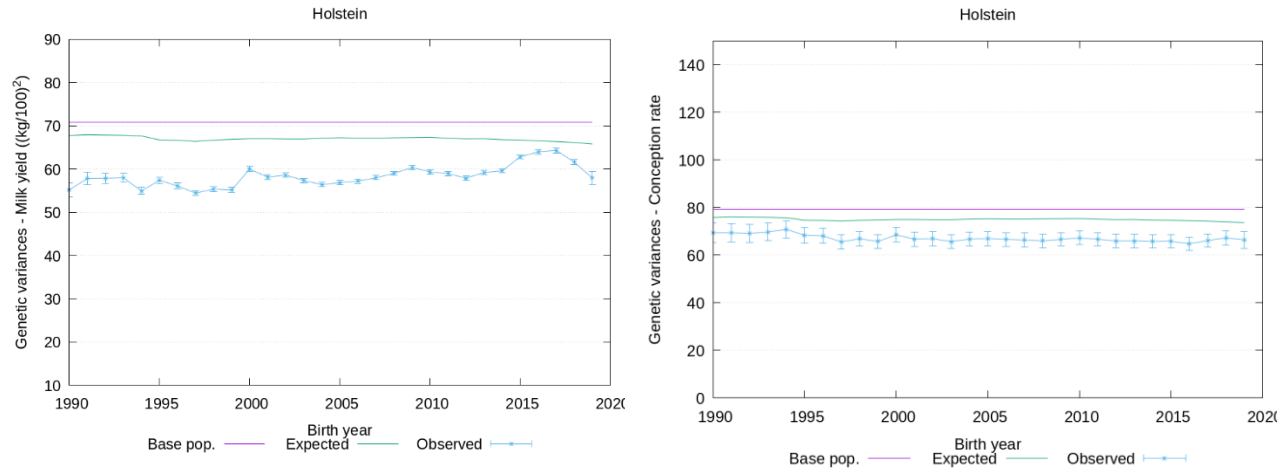

Figure 4. *Evolution of genetic variance for 305-day milk yield (MY; in (kg/100)^2) and conception rate (CR) along birth year for Dutch Holstein first-parity cows. Depicted genetic variances are genetic variances at the base population, expected genetic variances, and the observed genetic variances. Posterior standard deviations are represented by error bars.*

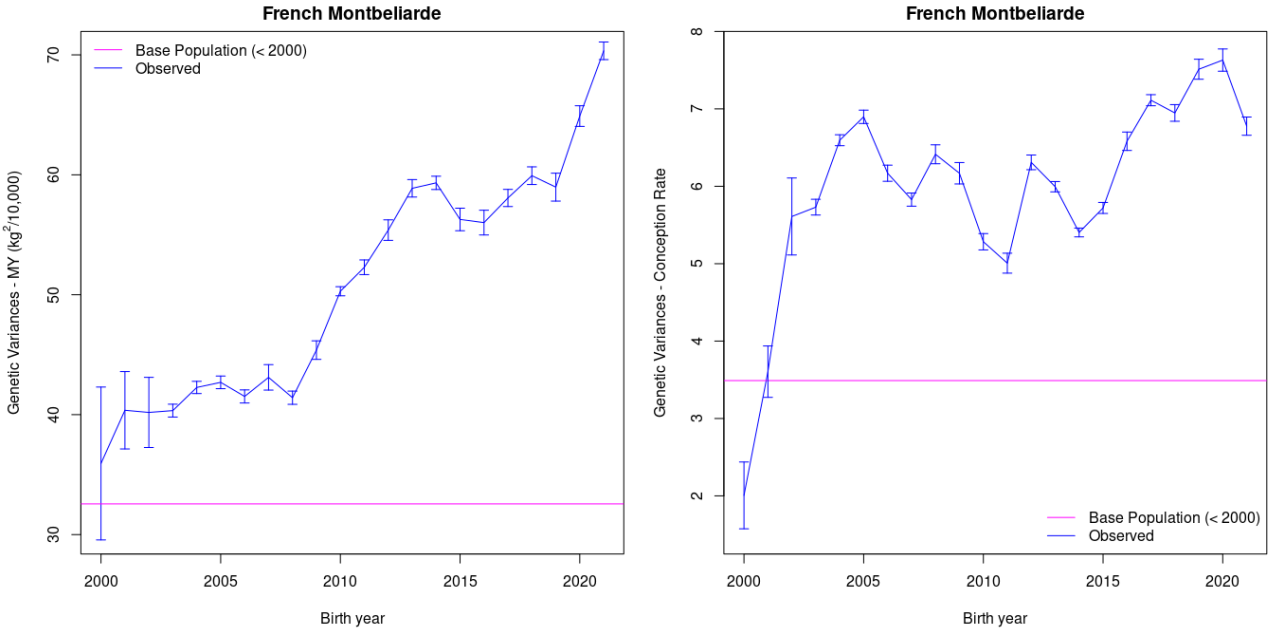

Figure 5. *Evolution of genetic variance for 305-day milk yield (MY; in (kg/100)^2) and conception rate (CR) along birth year for French Montbeliarde first-parity cows. Depicted genetic variances are genetic variances at the base population (1991-1999) and the observed genetic variances for the following generations. Year-by-year estimates for the period 1991-1999 were discarded for presentation due to a lack of convergence.*

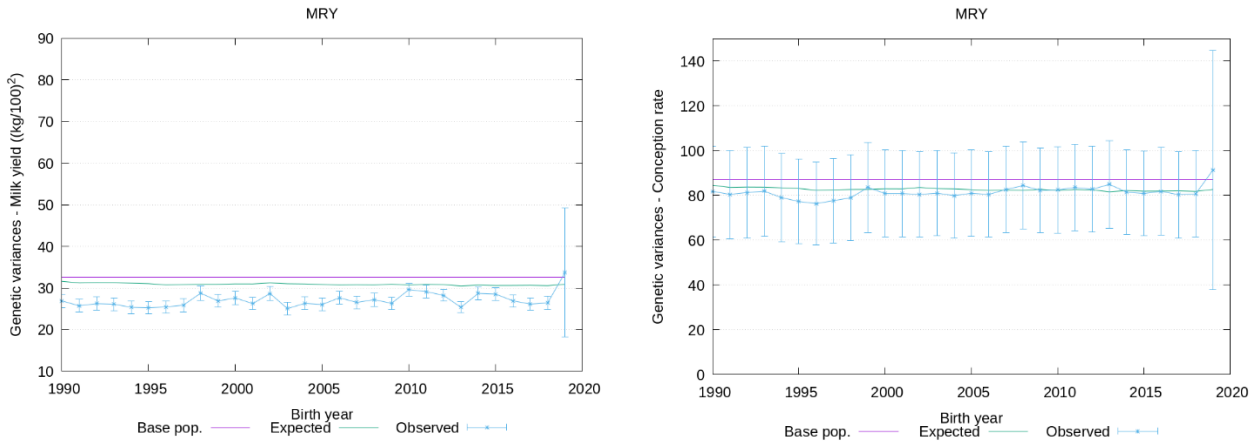

Figure 6. *Evolution of genetic variance for 305-day milk yield (MY; in (kg/100)^2) and conception rate (CR) along birth year for MRY first-parity cows. Depicted genetic variances are genetic variances at the base population, expected genetic variances, and the observed genetic variances. Posterior standard deviations are represented by error bars.*

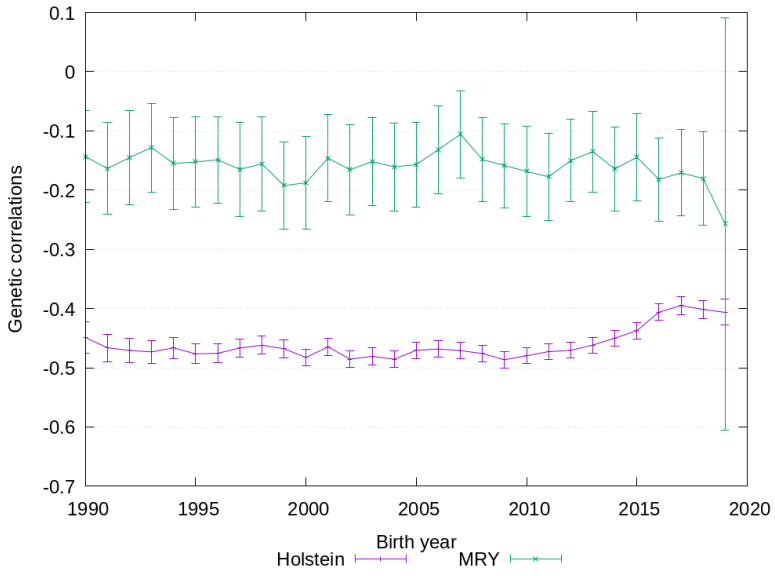

Figure 7. *Evolution of estimates of observed genetic correlations between 305-day milk yield and conception rate along birth years for Dutch Holstein and MRY first-parity cows. Posterior standard deviations are represented by error bars.*

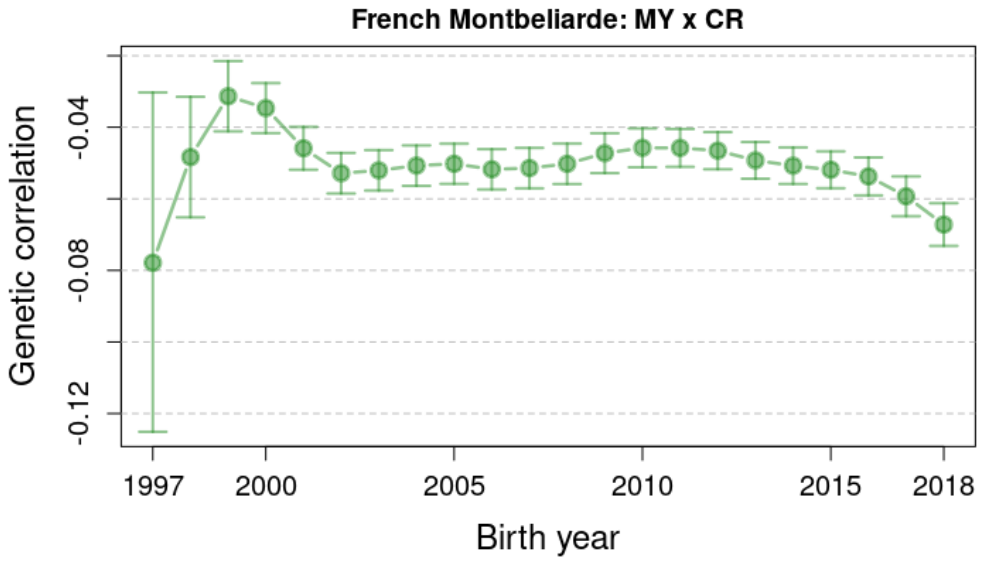

Figure 8. *Evolution of estimates of observed genetic correlations between 305-day milk yield and conception rate along birth years for French Montbeliarde first-parity cows. Posterior standard deviations are represented by error bars.*

Figures 7 and 8 depict the evolution of estimates of observed genetic correlations between 305-day MY and CR along birth year for Dutch Holstein, French Montbeliarde and MRY first-parity cows. A different trend can be observed for each population. For MRY, the genetic correlation remains constant along birth year, within values of -0.1 and -0.3. For Holstein, a plateau (close to -0.50) can be also observed until 2009. From the year 2009, the estimates of genetic correlation between 305-day MY and CR become less negative to achieve a value of around -0.40 in 2017, followed by a possible stabilization. These changes are difficult to interpret in light of the introduction of genomic selection in the late 2000's and of the history of the Dutch breeding goal. Indeed, the Dutch breeding goal changed a few times during the last 20 years, as it includes fertility traits, in addition to milk production, since August 2001 (M. van Pelt, personal communication), and it has been updated several times (respectively, in 2007, 2012, 2018, and 2022), for example, by including new traits (https://www.cooperatie-crv.nl/wp-content/uploads/2023/04/E_20-NVI-April-2023-Engels.pdf). Therefore, further analyses will be needed to confirm these trends for the Dutch cow populations. For the French Montbeliarde population, from 2009, we start to see a suggestion of an intensification of the negative genetic correlation between 305-day milk yield and conception rate along birth years. It is worth noting that fertility was included into the breeding goals for the French Montbeliarde in 2001, mostly for the AI bulls. The effect of this becomes more visible in the cows around 2009, which also coincides with the period in which fertility was also largely extended as a breeding goal directly for the cows. This result of intensifying genetic correlation matches the expected by the theoretical derivations.

Finally, additional analyses for the Dutch Holstein, French Montbeliarde and Dutch MRY populations, have been performed to investigate the evolution of genetic correlations between conception rate and 305-day fat yield (FY), and between CR and 305-day protein yield (PY). As these results are similar to the results obtained for MY and CR, these results for CR, FY, and PY are reported in Appendix 2. The similarity of these results among all traits is likely due to the very high genetic correlation between the production traits. For example, with respect to the correlations between MY-FY, MY-PY, and FY-PY, these correlations were consistently high (~0.97) for the French

Montbeliarde, and did not present significant changes in their values throughout the birth years studied.

In conclusion, we first proposed a theory on the evolution of genetic correlations, and second we observed that the evolution of estimates of observed genetic (co)variances and correlations between milk production traits and a fertility trait across birth years follow different trends for Dutch Holstein, French Montbeliarde and MRY first-parity cows. The different trends in genetic (co)variances and correlations could be due to either the Bulmer effect, as shown by the theoretical derivations, or to changes of national breeding goals. Therefore, further analyses will be needed to confirm these trends in each cow population.

4. Using genome level trade-offs in breeding strategies

4.1 Background

Many examples exist of antagonistic genetic correlations between important traits, such as between milk yield traits and fertility in dairy cattle. The common approach to deal with those is to include the traits involved in the breeding goal, using weights to either improve both traits simultaneously, or improve one trait, while avoiding that the other trait deteriorates. An example of the first approach is shown for the selection of increased litter size (total number born) and piglet mortality by Knap et al. (2023), as illustrated in Figure 9. This illustration shows that simultaneously selecting to improve both traits works, while the genetic correlation remains antagonistic. Theoretical studies have shown that with multitrait selection, genetic correlations may actually become more antagonistic (Sheridan and Barker, 1974).

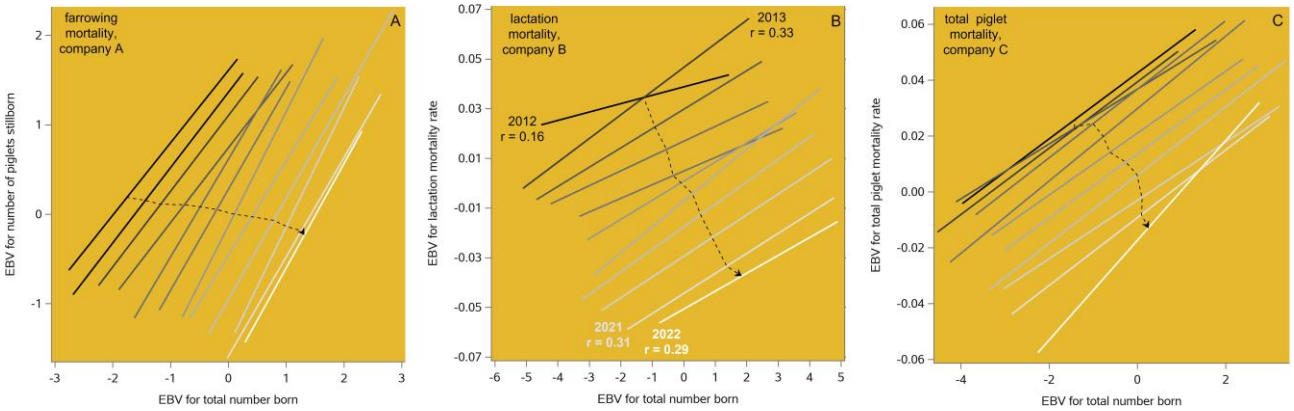

Figure 9. *Relationship between routinely estimated breeding values (EBVs) for piglet mortality rate and the total number born (litter size) in pigs, across the years 2012 to 2022 (Source: Knap et al. (2023)).*

Two important models that explain genetic correlations between traits, are: 1) pleiotropy, meaning that one or more alleles may have a desirable effect on the one trait, and an undesirable effect on the other trait, and 2) linkage between loci, such that a desirable allele for one trait is linked to an undesirable allele for the other trait. In the case of pleiotropy, selection may not be able to change the correlation between the traits, while in the case of linked loci, recombinations between those loci could yield to joint inheritance of combinations of alleles that are favorable for both traits. Considering that marker effects are estimated in routine genomic evaluations, this information could potentially be used to identify animals that carry more combinations of desirable alleles than what is observed on average in the population.

This task aimed to investigate to which extent exploiting variation in realized genetic correlations across families can help to achieve balanced breeding goals including antagonistic traits, and to mitigate the impact of antagonistic genetic correlations between them.

4.2 Approach

In animal breeding, commonly truncation selection based on estimated breeding values is applied, with the aim to maximize the average breeding value in the offspring generation. It is generally believed that this optimizes the rate of genetic gain achieved in the next generation. In the last decade, a few studies have suggested to not only consider the expected average breeding value of matings of selected individuals, but also the expected variances, and effectively select and mate individuals whose offspring display a larger gametic Mendelian Variance (gMV) (Segelke et al., 2014; Bonk et al., 2016; Bijma et al., 2020). The logic behind this, is that matings resulting in larger gMV, result in offspring from which the best animals may actually be better, compared to a situation where their parents were selected based on truncation selection. To be able to include the gMV in the selection criterion, one needs to compute the gMV for each individual. This can be done either algebraically (Bonk et al., 2016; Santos et al., 2019), or by simulating gametes from the phased genotypes of an individual, and then computing the variance of the expected breeding values of those gametes (Segelke et al., 2014).

This idea thus far mostly has been applied to single traits, but in principle can be extended to multiple traits (Bonk et al., 2016). The logic here is that across the gametes of an individual, a covariance between two traits can be computed. If this gametic Mendelian Covariance (gMC) varies across individuals just like the gMV, then it can be hypothesized that the impact of unfavourable genetic correlations can be alleviated by including the gMC in the selection criterion. To investigate this, some simulations were undertaken, and a selection strategy was identified to increase response to selection in the presence of trade-offs.

4.3 Description of simulations

We adopted the simulation strategy described by Bijma et al. (2020), that either considers that the initial generation was selected (i.e. in Bulmer equilibrium) or not, and extended it to two traits. This simulation strategy directly samples breeding values and gMV and gMC for a parental generation, and breeding values for an offspring generation from distributions that are parameterized based on empirical observations, without explicitly simulating e.g. genomes, loci or genotypes. Firstly, an initial generation was simulated, by drawing breeding values for the two traits (A_1 and A_2) both with a genetic variance of 1, a genetic correlation of -0.3 between them, and assumed equal breeding goal weights of 1 for both traits. For the scenario assuming that the initial population was unselected, this initial generation was used as the parental breeding values in the next step. For the scenario assuming the initial population was selected, starting from the initial generation four generations of truncation selection based on breeding values were performed, using 50 selected males and 50 selected females as in later generations. The fourth generation was assumed to be in Bulmer equilibrium (Bulmer, 1971), and this generation was considered to be the parental generation for the selected scenario. For both scenarios, gMV (σ^2_{g1} and σ^2_{g2}) and gMC (σ_{g12}) were simulated for the parental generation. Within trait, gMV were assumed to be independent from the breeding values, following the empirical observations by Segelke et al. (2014). In contrast to Bijma et al. (2020), we sampled the gMV from a normal distribution, and not a log normal distribution, to simplify the implementation with two traits. gMV within trait were assumed to be distributed as $N(0.25,0.0025)$, where the mean value of 0.25 is equal to half of the Mendelian Sampling variance, and assuming a coefficient of variation for the gMV of 20% as observed by Bonk et al. (2016), yields a standard deviation of 0.05 and a variance of 0.0025. To simulate variation in the gMC, we used the observed

distributions of gametic Mendelian correlations between, on the one hand, fat and protein yield, and, on the other hand, somatic cell count and the direct genetic effect for stillbirth, by Bonk et al. (2016). These distributions showed a standard deviation of gametic Mendelian correlations of ~0.125. Following this, for each possible parent an individual gametic Mendelian correlation ($r_{g12,parent}$) was drawn from *N*(-0.3,0.125). Finally, the simulated gMV for the two traits for each possible parent were drawn from the following distribution:

$$\begin{bmatrix}\sigma^2_{g1,parent}\\ \sigma^2_{g2,parent}\end{bmatrix} = \left(\begin{bmatrix}0.25\\ 0.25\end{bmatrix}, \begin{bmatrix}0.0025 & \sigma_{g12,parent}\\ \sigma_{g12,parent} & 0.0025\end{bmatrix}\right).$$

From this assumed parental generation, 100 parents (of which at random 50 were assumed to be males and 50 to be females) were selected, assuming a selected proportion (*p*) of 0.1, 0.01 or 0.001. To enable this, in the previous step the parental generation included 100/*p* individuals. Selection of the 100 parents was based on one of the following four criteria:

1. $A_1 + A_2$
2. $A_1 + A_2 + \sqrt{2}x_p\left(\sigma_{g1,parent} + \sigma_{g1,parent}\right)$
3. $A_1 + A_2 + \sqrt{2}x_p\sqrt{\sigma^2_{g1,parent} + 2\sigma_{g12,parent} + \sigma^2_{g2,parent}}$
4. Random

where the first criterion is truncation selection based on breeding values, the second criterion is the sum of the Index5 defined by Bijma et al. (2020) of both traits, where x_p is the truncation point of a standard normal distribution belonging to the upper-tail proportion *p* (i.e. x_p=1.282 for *p*=0.1, x_p=2.326 for *p*=0.01, and x_p=3.090 for *p*=0.001). The third criterion is also based on Index5, however, in this case we first computed the gMV of the index, which is equal to $\sigma^2_{g1,parent} + 2\sigma_{g12,parent} + \sigma^2_{g2,parent}$, given that both traits have weights of 1 in the breeding goal, and then took the square root to get the gametic Mendelian standard deviation of the index. The fourth criterion simply selected parents at random to provide a baseline scenario for the observed genetic correlation in the offspring generation in the absence of selection.

Using the 50 selected males and 50 selected females, 100/*p* offspring were generated. For each offspring, a breeding value was simulated for each trait:

$A_{1,off} = 0.5A_{1,sire} + 0.5A_{1,dam} + MS_1 = 0.5A_{1,sire} + 0.5A_{1,dam} + g_{1,sire} + g_{1,dam}$

$A_{2,off} = 0.5A_{2,sire} + 0.5A_{2,dam} + MS_2 = 0.5A_{2,sire} + 0.5A_{2,dam} + g_{2,sire} + g_{2,dam}$

where MS_1 and MS_2 are the Mendelian sampling terms for traits 1 and 2, $g_{1,sire}$ and $g_{2,sire}$ were the breeding values for the gametes for the sire were drawn from $\begin{bmatrix}\sigma^2_{g1,sire} & \sigma_{g12,sire}\\ \sigma_{g12,sire} & \sigma^2_{g2,sire}\end{bmatrix}$, and the breeding values for the gametes of the dam ($g_{1,dam}$ and $g_{2,dam}$) were drawn from $\begin{bmatrix}\sigma^2_{g1,dam} & \sigma_{g12,dam}\\ \sigma_{g12,dam} & \sigma^2_{g2,dam}\end{bmatrix}$. From those 100/*p* generated offspring, the best 100 were selected based on their breeding values ($A_1 + A_2$). Genetic gain was computed as the difference in mean $A_1 + A_2$ between the selected and all offspring. Genetic gain obtained when selecting parents using criteria 2 and 3, were expressed relative to genetic gain obtained using criterion 1. Finally, the genetic correlation in the offspring generation was computed as the Pearson correlation between MS_1 and MS_2 of all generated offspring, since this is a representation of the full genetic (co)variances in this generation, not affected by being computed from a selected sample.

For both the unselected and selected scenario, 10,000 replicates were simulated, and final results were averages across those replicates. Together with the averages standard errors are presented

that are computed as the standard deviation across the replicates divided by 100 (the square root of the number of iterations). The R-script to perform the simulations and compute the obtained results, can be found in Appendix 3.

4.4 Results and Discussion

The observed genetic gain in the offspring generation, considering selection in the parental generation based on Index5, relative to the genetic gain observed with selection based on breeding values only, is shown in Table 2. The results were very similar between the unselected and selected scenario, and showed that at the highest selection intensity (p=0.001), using Index5 ignoring gMC increased genetic gain by 0.8%, while this increase was 2.0% when including gMC.

***Table 2**. Observed genetic gain considering selection based on Index5 ignoring or including gametic Mendelian Covariances (gMC), expressed as percentagewise improvement relative to the genetic gain with selection based on breeding values only, considering scenarios with Bulmer equilibrium (Selected) or not (Unselected), and different selected proportions (p).*

Scenario	P	Index5 excl. gMC (SE)	Index5 incl. gMC (SE)
Unselected	0.1	0.4 (0.05)	0.8 (0.05)
	0.01	0.5 (0.03)	1.3 (0.03)
	0.001	0.8 (0.02)	2.0 (0.02)
Selected	0.1	0.5 (0.05)	0.7 (0.05)
	0.01	0.6 (0.03)	1.4 (0.03)
	0.001	0.8 (0.02)	2.0 (0.02)

Observed genetic correlations in the offspring generation for each of the four selection criteria applied in the parental generation, are shown in Table 3. Results show that the genetic correlation is virtually equal to the simulated value of -0.3 in the parental generation, except for the scenario where selection was based on Index5 including the gMC, in which case the genetic correlation was actually closer to 0, i.e. it was -0.28 with the lowest selection intensity, and -0.22 with the highest selection intensity. Just like for the genetic gain, results were very similar between the unselected and selected scenarios.

***Table 3.** Observed genetic correlation in the offspring generation, when parents are selected at random, based on breeding values, or considering selection based on Index5 ignoring or including gametic Mendelian covariances (gMC).*

		Selection of parents based on:			
Scenario	p	Random	Breeding values	Index5 excl. gMC	Index5 incl. gMC
Unselected	0.1	-0.295 (0.003)	-0.296 (0.003)	-0.295 (0.003)	-0.278 (0.003)
	0.01	-0.296 (0.002)	-0.296 (0.002)	-0.295 (0.002)	-0.249 (0.002)
	0.001	-0.296 (0.001)	-0.296 (0.001)	-0.293 (0.001)	-0.217 (0.001)
Selected	0.1	-0.295 (0.003)	-0.296 (0.003)	-0.295 (0.003)	-0.279 (0.003)
	0.01	-0.296 (0.002)	-0.296 (0.002)	-0.295 (0.002)	-0.249 (0.002)
	0.001	-0.296 (0.001)	-0.296 (0.001)	-0.293 (0.001)	-0.217 (0.001)

These results suggest that in situations where selection is targeting traits that are antagonistically correlated with each other, using selection criteria that include individual gMV and covariances may

not only lead to small increases in genetic gain, as has been observed for single trait selection (Bijma et al., 2020), but may also have a positive effect on the antagonistic genetic correlations. Other studies have suggested that multitrait selection may make genetic correlations more antagonistic. In our simulations, this would imply that we expect the genetic correlations in the offspring generation to be more negative. This was, however, not the case for any of the considered scenarios. This observation shows that our results need verification using full-blown stochastic simulations, including genomes, loci and contributions of both linkage and pleiotropy to the genetic correlations. It should also be noted that our approach to obtain for each individual the gametic Mendelian covariance matrix, likely can be improved upon. To inform future work, it is important that more empirical estimates of the distributions of gametic Mendelian covariance matrices for multiple traits are computed and published. To our knowledge, gametic Mendelian correlations are only reported by Bonk et al. (2016), while thus far no empirically derived parameters of the multivariate distributions of gametic Mendelian covariance matrices including multiple traits have been published.

5. Selection strategies to increase resilience to climate change

5.1 Background

In tropical and temperate regions, global warming and climate change are major threats to the livestock industry and the world's economy because they negatively impact livestock production by reducing productivity, reproductive efficiency and welfare of the animals. In dairy cattle production, this adverse weather condition is evident in the form of heat stress which advertently reduces milk production and impairs fertility (Liu et al., 2019; Wolfenson and Roth, 2019). Although several management strategies (Renaudeau et al., 2012) are being deployed to reduce the impact of heat stress in dairy cattle, there is a need to include genetic selection for heat-tolerant animals in the breeding program as this will ensure the long-term sustainability of the dairy industry. Genetic selection of heat-tolerant animals is beneficial because the additive genetic component of heat tolerance could be inherited by subsequent generations resulting in cumulative positive genetic gains for heat tolerance in the long term (Simms, 2000).

The temperature-humidity index (THI) which combines the temperature and relative humidity on a test day is a heat stress function mostly used in dairy cattle breeding to determine the heat load in an animal's environment (Ravagnolo et al., 2000). The reaction norm model is one modelling technique used to determine the performance of an animal across different environmental conditions. The model combines routinely collected phenotypic data such as milk production, public weather station records such as temperature, and humidity to determine the degree of environmental sensitivity (Ravagnolo and Misztal, 2000). The slope of the reaction norm model reflects the environmental sensitivity of a genotype, while any genetic variation in the environmental sensitivity reflects the presence genotype-by-environment interaction. This model can be used to genetically select heat-tolerant animals because the performance of an environmental sensitive animal will vary across different THI levels, while a robust animal would maintain production across different THI levels (Bohmanova et al., 2007; Carabaño et al., 2014; Mbuthia et al., 2021).

Although several previous studies have modelled the effect of heat stress and estimated the genetic parameters for heat tolerance in dairy cattle production, there is a lack of information on how these results can be practically implemented in the breeding program to select animals that are heat tolerant (Carabaño et al., 2019). The broken line model has been used to study the genetic variability

of heat tolerance and to determine the genetic value of heat tolerance of animals. The broken line model is defined by two parameters: 1) The thermoneutrality threshold and 2) the slope of decay in production after passing this threshold due to heat stress (Bernabucci et al., 2014). In this model, the slope of the decay after the threshold determines the sensitivity of the animal to heat stress and is used as one of the selection criteria.

A variety of breeding program simulation tools (Sargolzaei and Schenkel, 2009; Faux et al., 2016; Pook et al., 2020) have been developed to assist breeders in virtually modelling different breeding scenarios and investigate the future implication of their breeding decisions. These simulation softwares help breeders to optimize their breeding program in silico, to avoid the loss of opportunities and penalties from wrong decisions in real livestock selection programs. The objective of this study is to simulate the impact of selecting for heat tolerance in dairy cattle breeding using stochastic simulations of a breeding program, using the simulation software MoBPS. Wers obtained from , to inform the simulation.

5.2 Description of simulations (Material & Methods)

Historical population

To achieve our objective, we simulated a standard dairy cattle breeding program. For this simulation, the R-package MoBPS (Pook et al., 2020) simulation software was used. A founder population was first generated using an exemplary cattle map file (map_cattle3) found in the maps available in the R-package MoBPS. The map file contains 6,600 markers and based on this, a founder population of 5000 animals were generated (50 bulls and 50 cows). To generate a population structure, the founder population were randomly mated and an arbitrarily chosen size of 5000 animals was generated from this random mating. In each generation, 100,000 cows were mated to 2500 bulls to produce 100,000 offspring, which means that 40 females were mated to 1 bull, and the females had one offspring per generation while the males had multiple offspring. Cows that completed their first lactation (i.e., two years back) were phenotyped, and breeding value estimation was done for cows and bulls from the last two generations.

Breeding Scenarios

In total, four different scenarios were simulated (or an overview, see Table 4). Scenario 1 represents a situation where heat stress is ignored and is assumed not to be affecting either of the traits. In this scenario, two quantitative traits (Conception rate (CR) and Protein yield (PY)) were simulated assuming 1000 underlying additive quantitative trait loci each and having heritabilities of 0.023 and 0.24 respectively. Those heritabilities were estimated for CR and PY in a Dutch dataset in WP3 (see Deliverable 3.1 report). We first simulated phenotypes for CR on a continuous scale, and then assigned within generation the (100-X)% animals with the lowest value a phenotype of 0 (not pregnant) and the remaining X% a value of 100 (pregnant). These 0/100 phenotypes were then inserted into the population to replace the continuous phenotypes. Given that this transformation reduces the observed genetic variance, we found out that to achieve a heritability at the 0-100 scale of 0.023, we had to double the heritability used to simulate the underlying continuous trait (and thus used a value of 0.046). The traits have a positive genetic correlation of 0.01, a genetic variance of 112 and 1.8 respectively, and a mean of 46% and 991 g/day respectively. Since the goal is to improve both traits simultaneously, breeding value estimation was based on a multi-trait model. Estimated breeding values were included in a selection index, with weights of 1 and 1.79 for conception rate and protein yield respectively. The weights represent the weights given to the milk

production (INET) and fertility index included in the Dutch NVI index (https://www.cooperatie-crv.nl/wp-content/uploads/2023/04/E_20-NVI-April-2023-Engels.pdf). The breeding cycle was repeated for 20 generations and the average of 50 independent replicates of the simulations were used as results in making the plots.

Scenario 2 simulated a situation where the phenotypes of the animals were affected by heat stress but this was not taken into account in the breeding value estimation. In this scenario, five traits were used in the simulation, conception rate level (CRL), conception rate slope (CRS), protein yield level (PYL), protein yield slope (PYS) and protein yield 2 (PY2). A detailed description of how the traits were defined and the phenotypes subjected to heat stress is written in scenario 3 below. Estimated breeding values were included in a selection index, with weights of 1, 0, 1.79, 0, and 0 for CRL, CRS, PYL, PYS and PY2 respectively. In this scenario, the breeding cycle was also repeated for 20 generations and the results averaged over 50 independent simulations were reported

In scenario 3 and 4, the impact of heat stress on both traits was simulated by modelling a heat stress threshold and a slope of decay. To model this impact of heat stress in MoBPS, an intercept and a slope were defined for the traits. Because the genetic parameters of CR and PY were previously estimated using a broken line model and a second order random regression model respectively, conception rate was defined by two traits (conception rate level (CRL) and slope (CRS)), while protein yield was defined by three traits (protein yield level (PYL), slope (PYS), and protein yield 2 (PY2), i.e. the quadratic term). Similar to scenario 1, all five traits were simulated assuming 1000 underlying additive quantitative trait loci each. The heritability, mean and genetic variance of CRL and PYL are the same as in scenario 1, while CRS, PYS and PY2 all have a heritability of 1, a mean of 0 and a genetic variance of 80, 0.016, 0.028 respectively, based on estimates obtained in WP3. To imitate reality, phenotyped animals were randomly assigned THI values based on the daily average THI value in the Netherlands for the past ten years. Those values, obtained from the website https://www.knmi.nl/klimaat, followed approximately a normal distribution with a mean of 53 and a standard deviation of 9.76, with a minimum of 14 and a maximum of 78. Thus, THI values were drawn from *N*(53,9.76), and values < 14 were set to 14, and values > 78 were set to 78.

Based on estimates of the broken line model applied to Dutch Holstein data for CR (see Deliverable 3.1 report), a THI threshold of 60 was used. To ensure that THI values below the threshold had zero effect on the trait and the resulting value ranged between zero and one, we transformed the THI values as we did in the estimation on the Dutch data:

$$THI_{adj} = \frac{\max\{(THI - 60), 0\}}{13.75}$$

The impact of THI on CR in the simulation was then modeled using the principle of reaction norms, such that the phenotype (CR) of each animal is a function of the breeding value of CRL, the breeding value of CRS multiplied by the adjusted THI value and the residual environmental effect. The simulated breeding value of each animal for CRL and CRS were extracted and used for the computation and the residual environmental effect was drawn from a normal distribution, using the *rnorm* function in R. This was modelled using the following equation:

$$P_i = A_{0,i} + (A_{1,i} \times THI_{adj}) + E_i$$

Where P_i is the phenotype of an animal, $A_{0,i}$ is the breeding value of the level, $A_{1,i}$ is the breeding value of the slope, and E_i is the residual environmental effect.

For PY, no threshold was identified, but a level, a slope and a quadratic term were defined and the THI values were standardized, following the procedure used in the estimation based on the Dutch data, such that after standardization a value of -1 corresponded to a THI of 10, and a value of 1 to a THI of 83. An equation showing how this was modelled is seen below:

$$THI2 = (THI - \frac{83+10}{2})/(\frac{83-10}{2})$$

$$P_i = A_{0,i} \times cov1 + A_{1,i} \times cov2 + A_{2,i} \times cov3 + E_i$$

Where $THI2$ is the standardized THI value, P_i is the phenotype of an animal for protein yield, $A_{0,i}$ is the breeding value for the level, $A_{1,i}$ is the breeding value of the slope, $A_{2,i}$ is the breeding value of the quadratic term and E_i is the residual environmental effect. The covariates were based on Legendre polynomials as done in the estimation on the Dutch data (for details, see Deliverable 3.1 report), and were defined as:

$$cov1 = \sqrt{0.5}$$

$$cov2 = \sqrt{1.5} \times (\mathrm{THI2})$$

$$cov3 = \sqrt{1.5} \times \sqrt{2.5} \times 0.5 \times (3 \times (THI2^2) - 1)$$

Just like for the first scenario, the phenotypes for CR were transformed to have values of 0 or 100, such that 46% of the animals had a phenotype of 100 (i.e. the animal is pregnant) and the remaining 54% had a phenotype of 0 (i.e. the animal is not pregnant). Afterwards, breeding values for the last three generations were estimated in MiXBLUP (Vandenplas et al., 2022) using a random regression model where the slopes and quadratic term were estimated as coefficients of the random regression on the corresponding covariates.

To model a scenario (scenario 3) where selection targets situations where the impact of heat stress is minimal, the covariates at a THI of ≤ 60 and 50 (based on the estimation from the Dutch data) for CR and PY respectively were multiplied with the estimated breeding values, to get one breeding value for CR and PY. Similarly, a scenario (scenario 4) where selection targets situations where the impact of heat stress is extreme was modelled by multiplying the covariates at a THI of 70 for CR and PY with the estimated breeding values to get one breeding value for CR and PY. In both scenarios, after the multiplication, the estimated breeding values were inserted back into the population and selection was done using the same selection index as in scenario 1. The breeding cycle was repeated for 20 generations and results averaged over 50 independent simulations were reported.

The R-scripts for each of the four scenarios are included in Appendix 4.

Table 4. *Summary of the different breeding scenarios simulated*

Name	Scenario 1 (No_HS)	Scenario 2 (TP)	Scenario 3 (CZ)	Scenario 4 (HS)
Heat stress present?	No	Yes	Yes	Yes
Phenotypes are affected by heat stress?	No	Yes	Yes	Yes
Heat stress is considered in EBVs?	No	No	Yes	Yes
Number of traits defined in the simulation	Two traits (CR and PY)	Five traits (CRL, CRS, PYL, PYS, PY2)	Same as in scenario 2	Same as in scenario 2
Number of traits used in selection	Two traits (CR and PY)	Two traits (CR, PY)	Two traits CR (i.e. CRL + CRS) and PY (i.e. PYL + PYS + PY2) at a THI of 50.	Two traits CR (i.e. CRL + CRS) and PY (i.e. PYL + PYS + PY2) at a THI of 70.
Selection index	A selection index of 1 and 1.79 for CR and PY respectively	Same as in scenario 1	Same as in scenario 1	Same as in scenario 1

No_HS = No heat stress, TP = Traditional practice, CZ = Comfort Zone, HS = Heat stress

5.3 Results & discussion

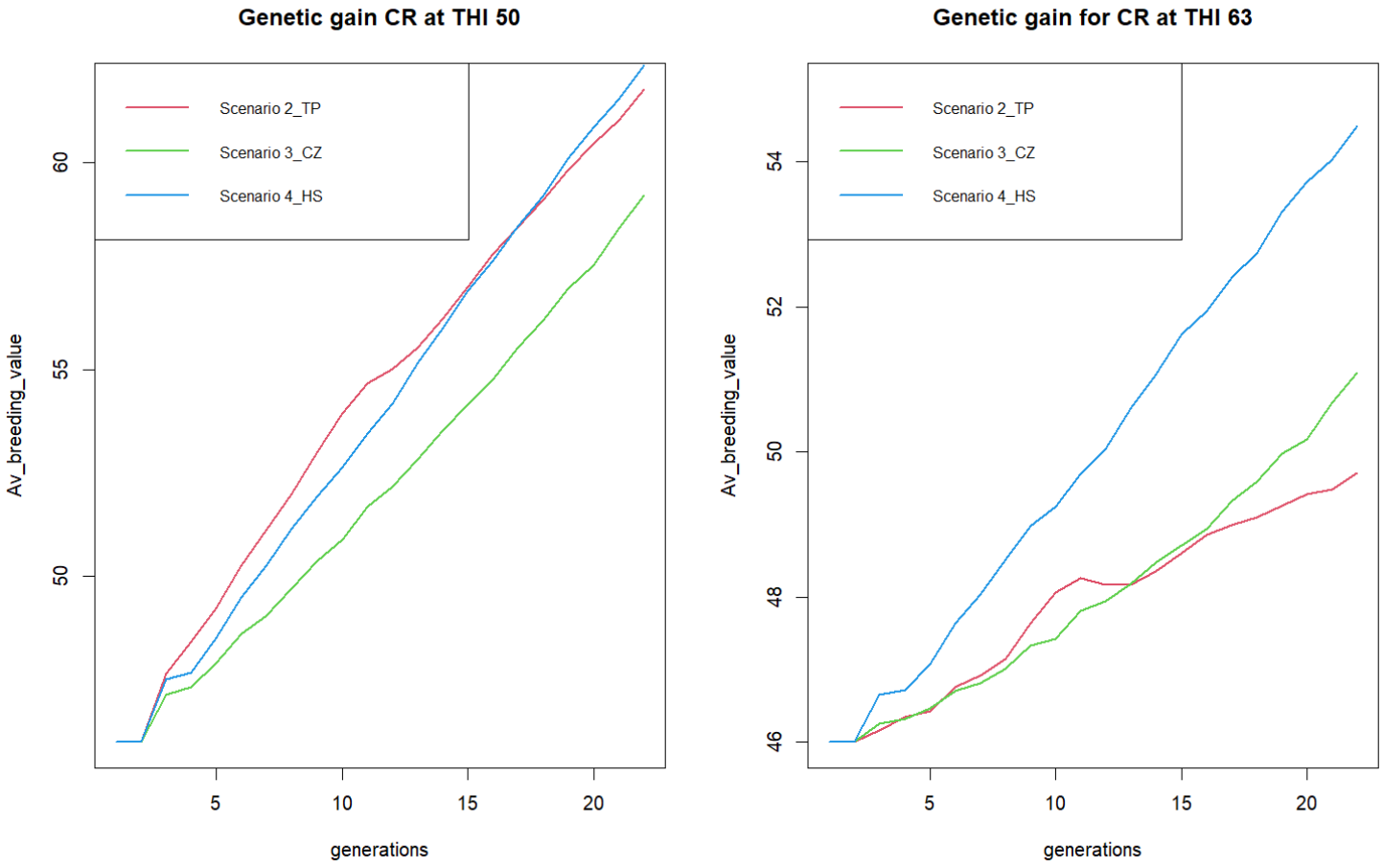

Figure 10. *Comparison between the genetic gain of conception rate at a THI of 50 (i.e. heat stress is minimal and animals are in their comfort zone) and a THI of 63 (i.e. heat stress is extreme) after 20 generations. TP = Traditional practice, CZ = Comfort zone, HS = Heat stress*

Results for the genetic gain of CR, at a THI of 50 and 63, are shown in Figure 10 for the three scenarios where heat stress did affect the phenotypes of CR. Figure 10 shows that at a THI of 50, when the animals are assumed to be in their comfort zone, the genetic gain for conception rate was consistent across generations in the three scenarios. The genetic gain is highest in scenario 2 (which depicts the current practice (TP) where heat stress is not taken into consideration in the breeding value estimation) and lowest in scenario 3 (where heat stress was considered when estimating breeding values, while selection was based on estimated breeding values below the threshold). The genetic gain in scenario 4 (where selection was based on estimated breeding values above the threshold) is very similar to the genetic gain in scenario 2. The genetic gain at a THI of 63, where heat stress is affecting CR, is highest in scenario 4, followed by scenario 2 and scenario 3. In comparing the two plots, the genetic gain at THI of 63 is more reflective of future cases where heat stress will most likely increase due to climate change. Hence, genetic gain is better in the scenarios where heat stress was taken into consideration while estimating the breeding values than in the scenario where it was not. The results of Rockett et al. (2023) and Rumigen WP3 showed that after a threshold of 60, there was a reranking in the estimated breeding values of bulls which means that

the daughters of some bulls are more tolerant to heat and can perform well at a high THI. Relating this to our result of the genetic gain for CR at THI 63, when heat stress is taken into account in the estimated breeding values used for selection, animals that are more tolerant to heat stress are selected. Hence, genetic gain is higher and the sensitivity of the animals to heat stress is reduced.

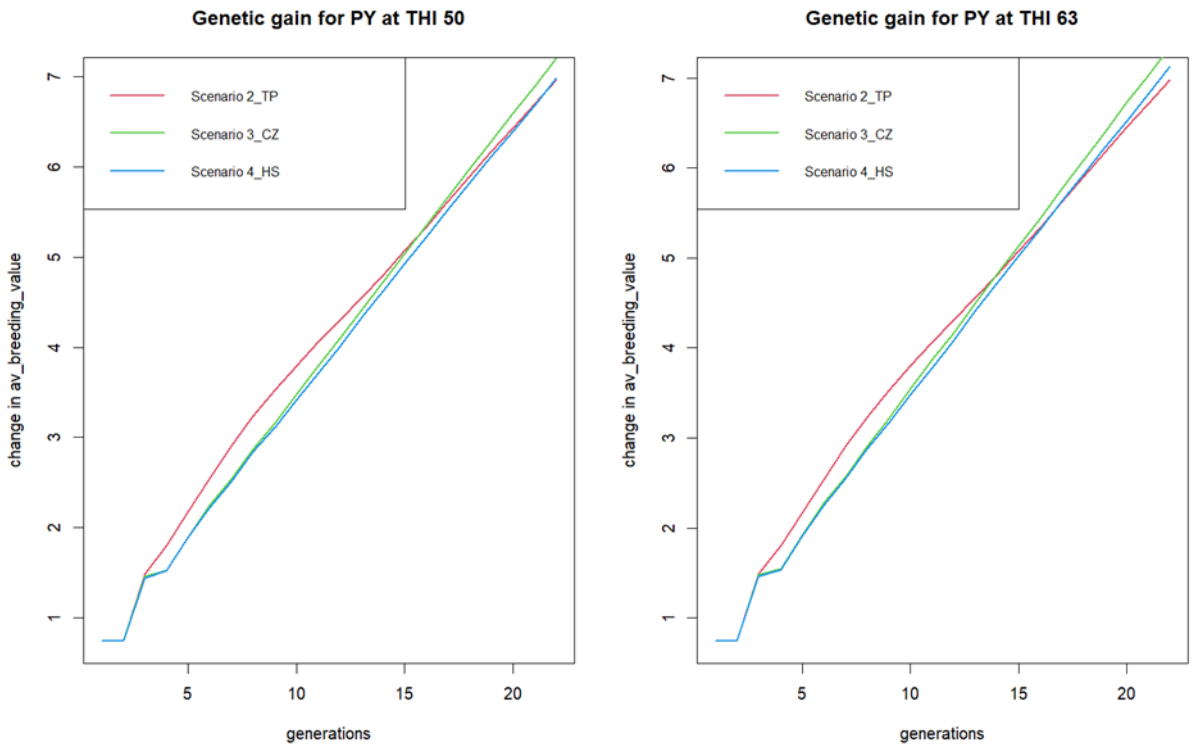

Figure 11. *Comparison between the genetic gain of protein yield at a THI of 50 (i.e. heat stress is minimal and animals are in their comfort zone) and a THI of 63 (i.e. heat stress is extreme) after 20 generations. TP = Traditional practice, CZ = Comfort zone, HS = Heat stress*

Figure 11 shows that the genetic gain of protein yield at a THI of 50 and 63 was consistent across generations in the three scenarios. However, comparing the two plots shows that at a THI of 63 which depicts a future increase in heat stress, there is a reranking in the genetic gain of the three scenarios such that genetic gain in scenario 2 (TP) becomes slightly lower towards after about 15 generations. This slight change in the genetic gain in scenario 2 probably means that with a higher number of generations and increasing climate change and heat stress, the cumulative achieved genetic gain in scenario 2 will be smaller.

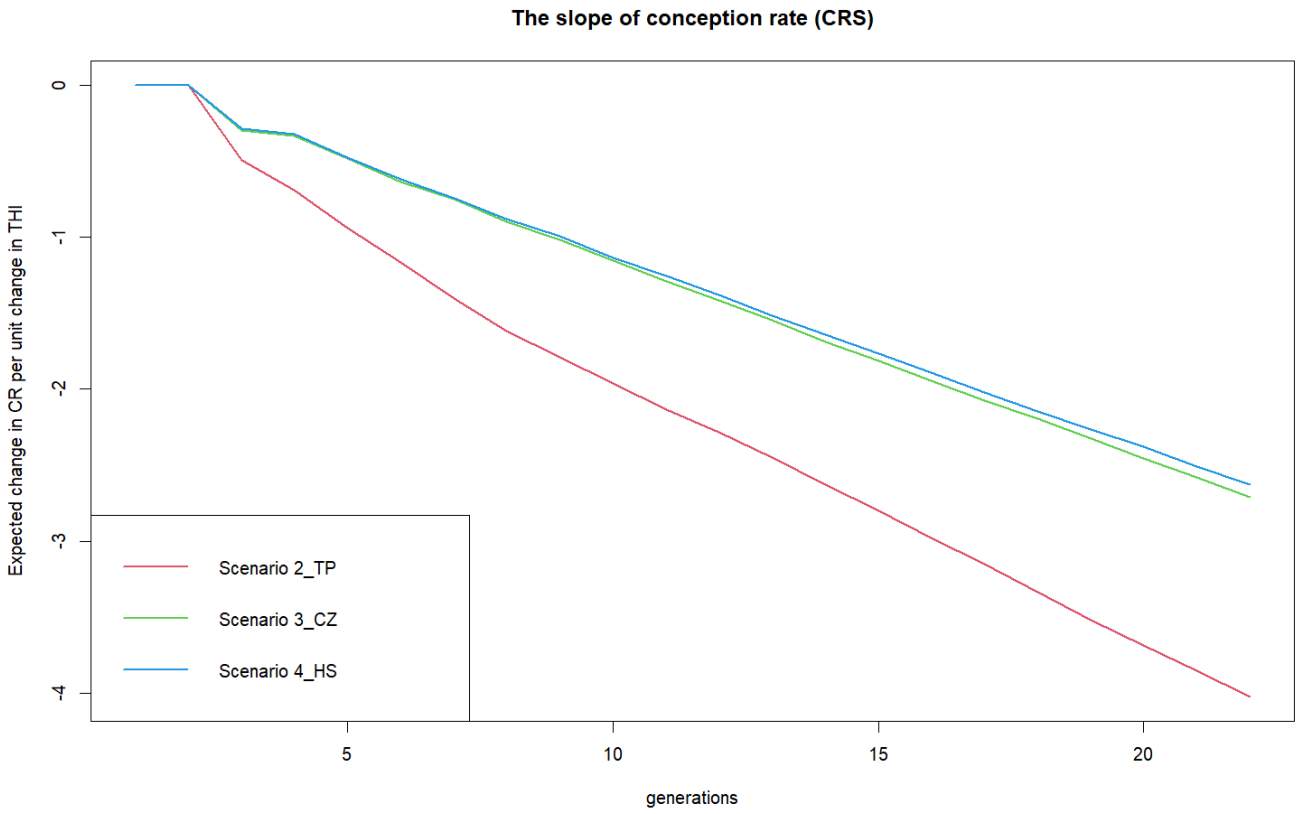

Figure 12. The genetic gain for the slope of conception rate expressed as the change in CR per unit increase in THI. TP = Traditional practice, CZ = Comfort zone, HS = Heat stress

In Figure 12, it can be seen that for all three scenarios, the slope of CR decreased across generations. In reaction norm models, the slope defines the sensitivity of the phenotypes to environmental changes, such that the steeper the slope, the more sensitive the phenotype is to the environment and vice versa (Simms, 2000). This means that the phenotypes in scenario 2 are more sensitive to heat stress than the phenotypes in scenarios 3 and 4. Therefore, this implies that considering heat stress in the breeding value estimation reduces the sensitivity of the phenotypes and indirectly leads to the selection of animals that are more productive and less sensitive to the environment. This practice would be beneficial for both production and welfare because of the increasing rate of global warming.

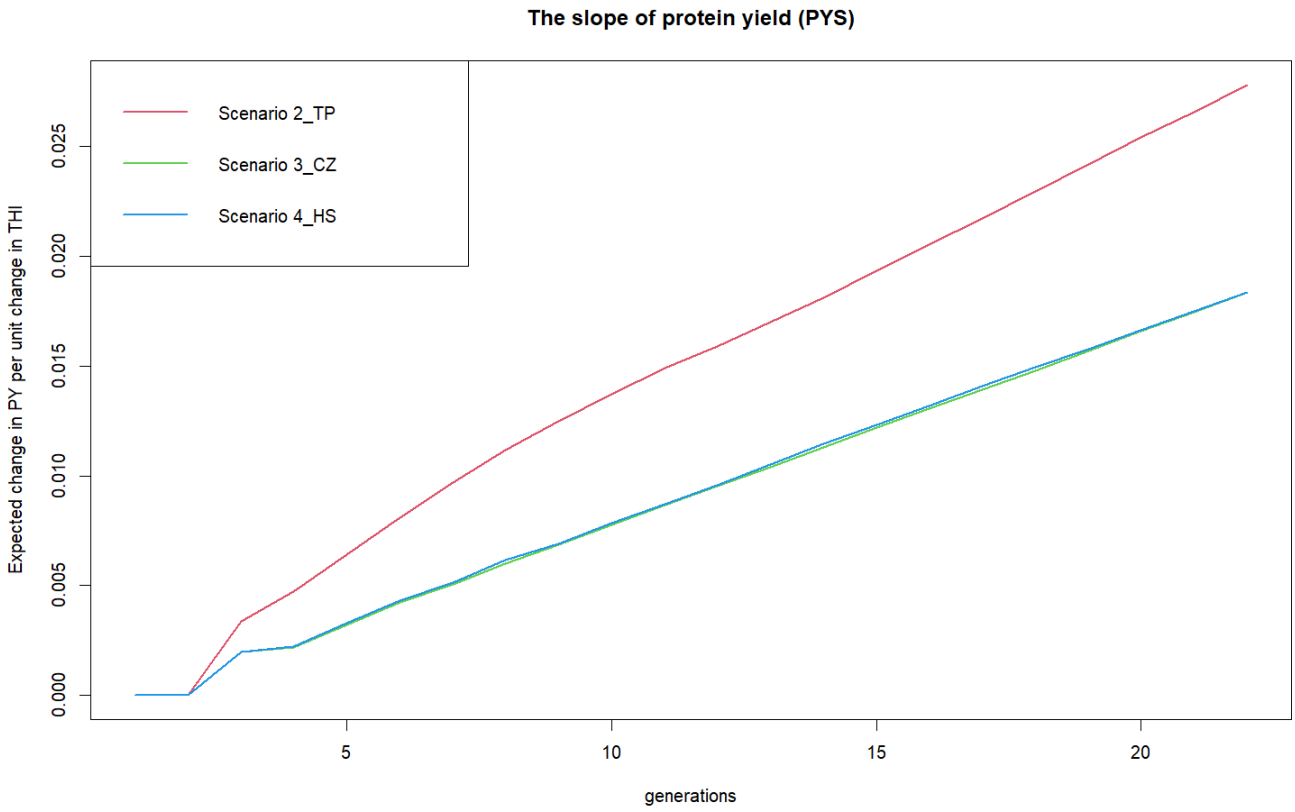

Figure 13. *The genetic gain for the slope of protein yield expressed as the change in PY per unit increase in THI. TP = Traditional practice, CZ = Comfort zone, HS = Heat stress*

Figure 13 shows that the slope of protein yield increased in the three scenarios across generations. The slope of protein yield was higher in scenario 2 than in scenarios 3 and 4. This means that the phenotypes in scenario 2 are less sensitive to heat stress than the phenotypes in scenarios 3 and 4. Nevertheless, the total genetic gain for protein yield is very similar in these scenarios 2, 3 and 4 (see Figure 11) and this likely due to the relatively small contribution of the slope to the total breeding value, compared to the contribution of the intercept.

In conclusion, the increasing rate of global warming and future forecast on climate change requires that farmers and breeding companies improve the robustness of their animals by leveraging on the genotype and environment interaction and selecting animals that are less sensitive to environment. This seems particularly important for fertility, and can be achieved by including traits (e.g. conception rate slope) that are related to heat stress in their breeding goal and considering heat stress in their breeding value estimation.

6. Conclusion

One of the main objectives of RUMIGEN is to provide statistical tools and breeding strategies for mitigating trade-offs between production, efficiency and resilience. The objectives of this deliverable 8.2 were to: (1) investigate the impact of the historic evolution of breeding goals on trade-offs between production, fertility, and resilience, (2) investigate possible benefits of using trade-offs at the genome level in balanced breeding strategies, and (3) develop optimal selection strategies to increase efficiency and resilience to climate change.

To achieve objective (1), we first proposed a theory on the evolution of genetic correlations, and second we observed that the evolution of estimates of observed genetic (co)variances and correlations between milk production traits and a fertility trait across birth years follow different trends for Dutch Holstein, French Montbeliarde and MRY first-parity cows. The different trends in genetic (co)variances and correlations could be due to either the Bulmer effect, as shown by the theoretical derivations, or to changes of national breeding goals.

To achieve objective (2), we investigated the potential to make use of gametic Mendelian Sampling variances and covariances, which explain observed genetic variances within traits and covariances between traits among the offspring of individual animals. Including Mendelian Sampling variances and covariances in selection resulted to up to 2% more genetic gain compared to ordinary selection on breeding values, while it was able to change the genetic correlation in the next generation from -0.3 to values ranging from -0.28 to -0.22, while ignoring the Mendelian Sampling covariance left the genetic correlation unaffected. Based on these results we conclude that use of individual Mendelian Sampling covariances in selection decisions may help to mitigate the effect of antagonistic genetic correlations in breeding programs.

To achieve objective (3), we stochastically simulated a dairy cattle breeding program with two traits, reflecting conception rate and protein yield, based on genetic parameters estimated on Dutch dairy data in WP3. Results showed that even when heat stress is considered in the breeding value estimation, the overall sensitivity to heat stress for conception rate may still increase, as a correlated response to selection for increased protein yield. Therefore, selecting animals that are less sensitive to heat stress requires that heat stress is considering in breeding value estimation, and that traits related to heat stress (e.g. conception rate and protein slope) are included in the breeding goal.

7. References

Bernabucci, U., S. Biffani, L. Buggiotti, A. Vitali, N. Lacetera, and A. Nardone. 2014. The effects of heat stress in Italian Holstein dairy cattle. J. Dairy Sci. 97:471-486. https://doi.org/10.3168/jds.2013-6611

Bijma, P., Y. C. J. Wientjes, and M. P. L. Calus. 2020. Breeding top genotypes and accelerating response to recurrent selection by selecting parents with greater gametic variance. Genetics. 214:91-107. https://doi.org/10.1534/genetics.119.302643

Bohmanova, J., I. Misztal, and J. B. Cole. 2007. Temperature-Humidity Indices as Indicators of Milk Production Losses due to Heat Stress. J. Dairy Sci. 90:1947-1956. https://doi.org/10.3168/jds.2006-513

Bonk, S., M. Reichelt, F. Teuscher, D. Segelke, and N. Reinsch. 2016. Mendelian sampling covariability of marker effects and genetic values. Genet. Sel. Evol. 48:36. http://dx.doi.org/10.1186/s12711-016-0214-0

Bulmer, M. G. 1971. The effect of selection on genetic variability. Am. Nat. 105:201-211. https://doi.org/10.1086/282718

Carabaño, M. J., K. Bachagha, M. Ramón, and C. Díaz. 2014. Modeling heat stress effect on Holstein cows under hot and dry conditions: Selection tools. J. Dairy Sci. 97:7889-7904. https://doi.org/10.3168/jds.2014-8023

Carabaño, M. J., M. Ramón, A. Menéndez-Buxadera, A. Molina, and C. Díaz. 2019. Selecting for heat tolerance. Anim. Front. 9:62-68. https://doi.org/10.1093/af/vfy033

Cochran, W. G. 1951. Improvement by means of selection. University of California Press. Pages p. 449-471 *in* Proceedings of the second Berkeley symposium on mathematical statistics and probability. University of California Press.

Faux, A.-M., G. Gorjanc, R. C. Gaynor, M. Battagin, S. M. Edwards, D. L. Wilson, S. J. Hearne, S. Gonen, and J. M. Hickey. 2016. AlphaSim: Software for Breeding Program Simulation. The Plant Genome. http://dx.doi.org/10.3835/plantgenome2016.02.0013

Fernando, R. L. and D. Gianola. 1990. Statistical Inferences in Populations Undergoing Selection or Non-Random Mating. Pages 437-453 *in* Advances in Statistical Methods for Genetic Improvement of Livestock. D. Gianola and K. Hammond, eds. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-74487-7_19

Gianola, D. and R. L. Fernando. 1986. Bayesian Methods in Animal Breeding Theory. J. Anim. Sci. 63:217-244. http://dx.doi.org/10.2527/jas1986.631217x

Gianola, D., R. L. Fernando, and C. C. Schön. 2022. Inference about quantitative traits under selection: a Bayesian revisitation for the post-genomic era. Genet. Sel. Evol. 54:78. https://doi.org/10.1186/s12711-022-00765-z

Hidalgo, J., S. Tsuruta, D. Lourenco, Y. Masuda, Y. Huang, K. A. Gray, and I. Misztal. 2020. Changes in genetic parameters for fitness and growth traits in pigs under genomic selection. J. Anim Sci. 98:skaa032. https://doi.org/10.1093/jas/skaa032

Hill, W. G. 1965. Studies on artificial selection. PhD thesis. University of Edinburgh.

Itoh, Y. 1991. Changes in genetic correlations by index selection. Genet Sel Evol. 23:301-308. https://doi.org/10.1186/1297-9686-23-4-301

Knap, P. W., E. F. Knol , A. C. Sørensen, A. E. Huisman, D. van der Spek, L. J. Zak, A. Granados Chapatte, and C. R. G. Lewis. 2023. Genetic and phenotypic time trends of litter size, piglet mortality, and birth weight in pigs. Front. Anim. Sci. 4:1218175. https://doi.org/10.3389/fanim.2023.1218175

Liu, J., L. Li, X. Chen, Y. Lu, and D. Wang. 2019. Effects of heat stress on body temperature, milk production, and reproduction in dairy cows: a novel idea for monitoring and evaluation of heat stress — A review. Asian-Australas. J. Anim. Sci. 32:1332-1339. https://doi.org/10.5713/ajas.18.0743

Macedo, F. L., O. F. Christensen, and A. Legarra. 2021. Selection and drift reduce genetic variation for milk yield in Manech Tête Rousse dairy sheep. JDS Comm. 2:31-34. https://doi.org/10.3168/jdsc.2020-0010

Mbuthia, J. M., M. Mayer, and N. Reinsch. 2021. Modeling heat stress effects on dairy cattle milk production in a tropical environment using test-day records and random regression models. Animal. 15:100222. https://doi.org/10.1016/j.animal.2021.100222

Pook, T., M. Schlather, and H. Simianer. 2020. MoBPS - Modular Breeding Program Simulator. G3 (Bethesda). 10:1915-1918. https://doi.org/10.1534/g3.120.401193

Ravagnolo, O. and I. Misztal. 2000. Genetic component of heat stress in dairy cattle, parameter estimation. J. Dairy Sci. 83:2126-2130. https://doi.org/10.3168/jds.S0022-0302(00)75095-8

Ravagnolo, O., I. Misztal, and G. Hoogenboom. 2000. Genetic component of heat stress in dairy cattle, development of heat index function. J. Dairy Sci. 83:2120-2125. https://doi.org/10.3168/jds.S0022-0302(00)75094-6

Renaudeau, D., A. Collin, S. Yahav, V. de Basilio, J. L. Gourdine, and R. J. Collier. 2012. Adaptation to hot climate and strategies to alleviate heat stress in livestock production. animal. 6:707-728. https://doi.org/10.1017/S1751731111002448

Rockett, P. L., I. L. Campos, C. F. Baes, D. Tulpan, F. Miglior, and F. S. Schenkel. 2023. Phenotypic analysis of heat stress in Holsteins using test-day production records and NASA POWER meteorological data. J. Dairy Sci. 106:1142-1158. https://doi.org/10.3168/jds.2022-22370

Santos, D. J. A., J. B. Cole, T. J. Lawlor, P. M. VanRaden, H. Tonhati, and L. Ma. 2019. Variance of gametic diversity and its application in selection programs. J. Dairy Sci. 102:5279-5294. https://doi.org/10.3168/jds.2018-15971

Sargolzaei, M. and F. S. Schenkel. 2009. QMSim: a large-scale genome simulator for livestock. Bioinformatics. 25:680-681. https://doi.org/10.1093/bioinformatics/btp045

Segelke, D., F. Reinhardt, Z. Liu, and G. Thaller. 2014. Prediction of expected genetic variation within groups of offspring for innovative mating schemes. Genet. Sel. Evol. 46:42. https://doi.org/10.1186/1297-9686-46-42

Sheridan, A. K. and J. S. F. Barker. 1974. Two-trait selection and the genetic correlation II. Changes in the genetic correlation during two-trait selection. Aust. J. Biol. Sci. 27:89-102. https://doi.org/10.1071/BI9740089

Simms, E. L. 2000. Defining tolerance as a norm of reaction. Evol. Ecol. 14:563-570. https://doi.org/10.1023/A:1010956716539

Sorensen, D., R. Fernando, and D. Gianola. 2001. Inferring the trajectory of genetic variance in the course of artificial selection. Genet. Res. 77:83-94. https://doi.org/10.1017/S0016672300004845

Vandenplas, J., R. F. Veerkamp, M. P. L. Calus, M. H. Lidauer, I. Strandén, M. Taskinen, M. Schrauf, and J. ten Napel. 2022. 358. MiXBLUP 3.0 – software for large genomic evaluations in animal breeding programs. Pages 1498-1501 *in* Proceedings of 12th World Congress on Genetics Applied to Livestock Production (WCGALP). https://doi.org/10.3920/978-90-8686-940-4_358

Villanueva, B., N. R. Wray, and R. Thompson. 1993. Prediction of asymptotic rates of response from selection on multiple traits using univariate and multivariate best linear unbiased predictors. Anim. Prod. 57:1-13. https://doi.org/10.1017/S0003356100006541

Wolfenson, D. and Z. Roth. 2019. Impact of heat stress on cow reproduction and fertility. Anim. Front. 9:32-38. https://doi.org/10.1093/af/vfy027

8. Annexes

Appendix 1: Description of the phenotypes used to investigate the evolution of genetic (co)variances across time

France

Records from cows with unknown parents or with missing data for any of the traits were excluded.

Performances and pedigree information of Holstein and Montbeliarde cows were extracted from the French national database. Phenotypic records for the years 1991 to 2021 were available. This complete period was used to estimate breeding values necessary for the study. The pedigree file totalized approximately 4 million individuals, with those retained for the study comprised, and for building the numerator relationship matrix, the pedigree was traced back three generations for the animals with records. Table 5 presents the size of the datasets available corresponding to the periods 1991-2021 and used for the estimation of breeding values.

Table 5. *Description of the datasets in the French Montbeliarde breed.*

Trait[1]	Period	Number of herds	Number of 305d records	Number of animals with records
MY, FY, PY (parity 1)	1991-2021	16,317	806,159	806,159
Conception rate (Parity 1)	1991-2021	16,317	806,159	806,159

[1]MY = milk yield; FY = fat yield; PY = protein yield.

The Netherlands

Only records from 100% Holstein cows and at least 87.5% Meuse-Rhine-Yssel (MRY) cows with both parents known were kept for all analyses..

For milk production traits, 305-day records for milk yield (kg), fat yield (kg), and protein yield (kg) of Holstein and (at least 7/8) MRY first-parity cows after (and including) 1990 were extracted from the database used for the Dutch-Flemish genetic evaluations of production traits. Also, only 305-day records from first-parity Holstein and MRY cows, that calved between 16 and 41 months, were kept. Finally, for milk production traits, each contemporary group (i.e., the effect “herd x year of calving”) was required to have at least five 305-day records. Similar data edits were used for conception rate (at least five cows per herd-year).

The pedigrees were traced back three generations from Holstein and MRY first-parity cows with records. The pedigrees included 1,372,953 animals for first-parity Holstein datasets, and 45,460 animals for first-parity MRY datasets.

Table 6 presents the size of the datasets available for the estimation of genetic parameters and breeding values for each trait and each breed.

Table 6. *Description of the datasets for the Dutch Holstein breed available for the analyses.*

Breed	Trait	Period	Number of herds	Number of animals with records
Holstein	Milk production	1990-2020	1579	855,185
	Conception rate	1990-2020	1578	676,641
Meuse-Rhine-Yssel	Milk production	1990-2020	83	27,036
	Conception rate	1990-2020	83	19,667

Appendix 2: Evolution of genetic (co)variances and correlations along time between conception rate, and protein and fat yields

Table 7. *Estimated variances at the base population for 305-day fat (FY) and protein (PY) yields. Posterior standard deviations are within brackets.*

Breed	Trait	Genetic variance[1]	Residual variance[1]
Holstein	FY	8.96 (0.07)	8.08 (0.62) – 12.92 (0.13)
	PY	5.29 (0.05)	4.82 (0.51) – 10.30 (0.09)
MRY	FY	5.02 (0.25)	4.68 (0.48) – 9.48 (0. 60)
	PY	3.01 (0.15)	2..84 (0.30) – 6.60 (0.39)

[1]Genetic and residual variances must be multiplied by 100.

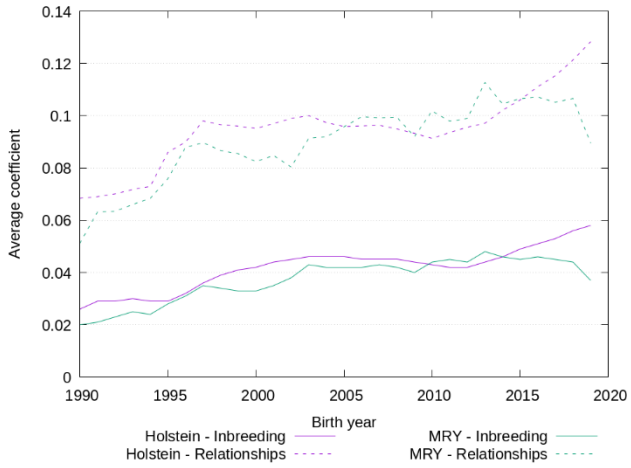

Figure 14. *Evolution of the average inbreeding coefficients and of average relationship coefficients by year of birth for Dutch Holstein and MRY first-parity cows. A complete pedigree of at least 3 generations was extracted for all first-parity Dutch cows.*

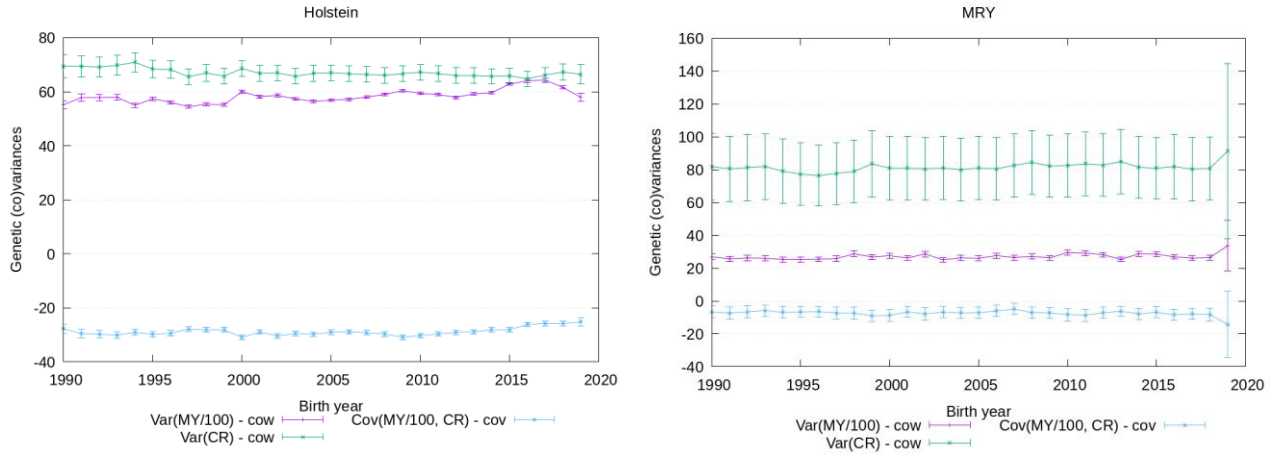

Figure 15. *Evolution of estimates of observed genetic (co)variances between 305-day milk yield (MY; in (kg/100)^2) and conception rate (CR) along birth years for Dutch Holstein and MRY first-parity cows. Posterior standard deviations are represented by error bars.*

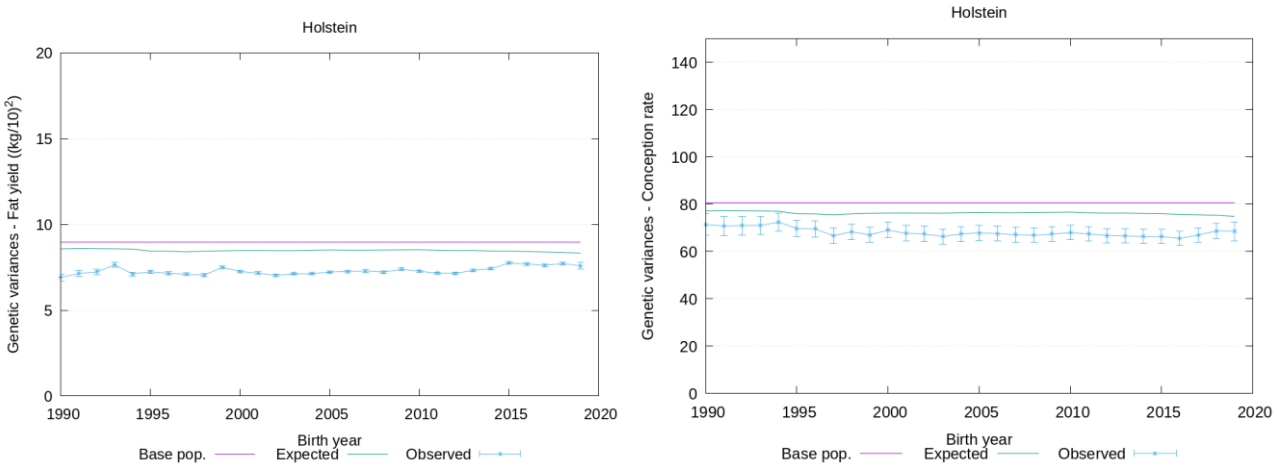

Figure 16. *Evolution of genetic variance for 305-day fat yield (FY; in (kg/10)^2) and conception rate (CR) along birth year for Dutch Holstein first-parity cows. Depicted genetic variances are genetic variances at the base population, expected genetic variances, and the observed genetic variances. Posterior standard deviations are represented by error bars.*

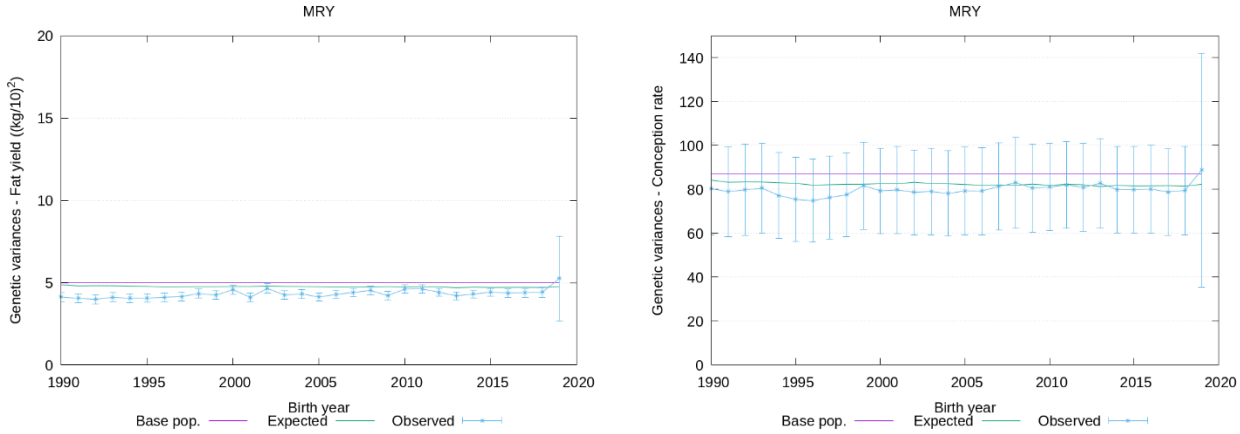

Figure 17. *Evolution of genetic variance for 305-day fat yield (FY; in (kg/10)^2) and conception rate (CR) along birth year for MRY first-parity cows. Depicted genetic variances are genetic variances at the base population, expected genetic variances, and the observed genetic variances. Posterior standard deviations are represented by error bars.*

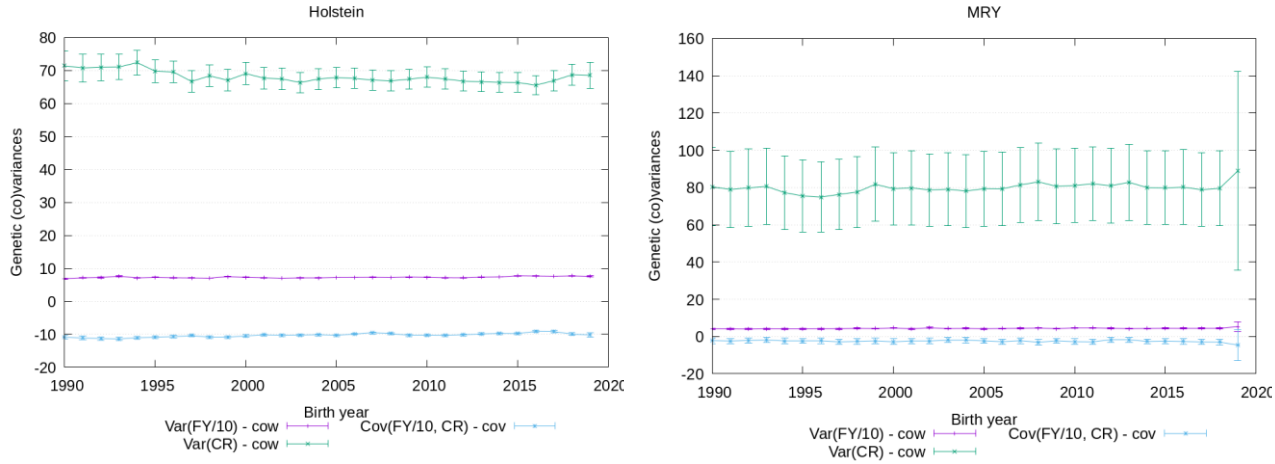

Figure 18. *Evolution of estimates of observed genetic (co)variances between 305-day fat yield (FY; in (kg/10)^2) and conception rate (CR) along birth years for Dutch Holstein and MRY first-parity cows. Posterior standard deviations are represented by error bars.*

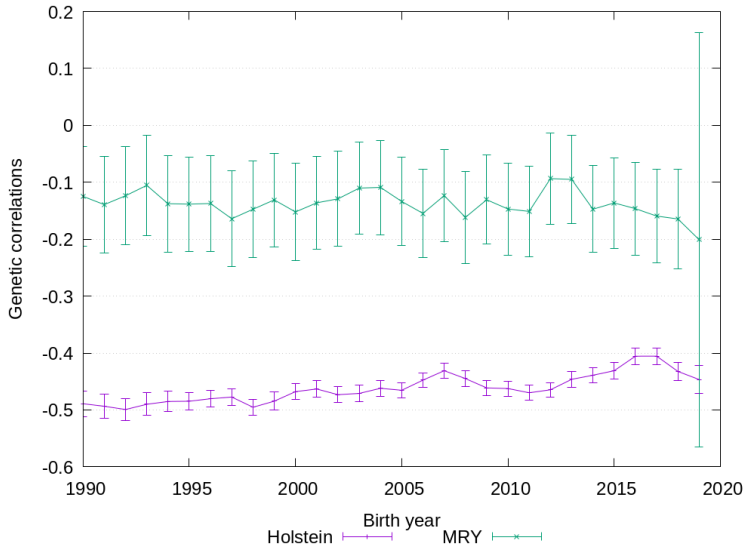

Figure 19. *Evolution of estimates of observed genetic correlations between 305-day fat yield and conception rate along birth years for Dutch Holstein and MRY first-parity cows. Posterior standard deviations are represented by error bars.*

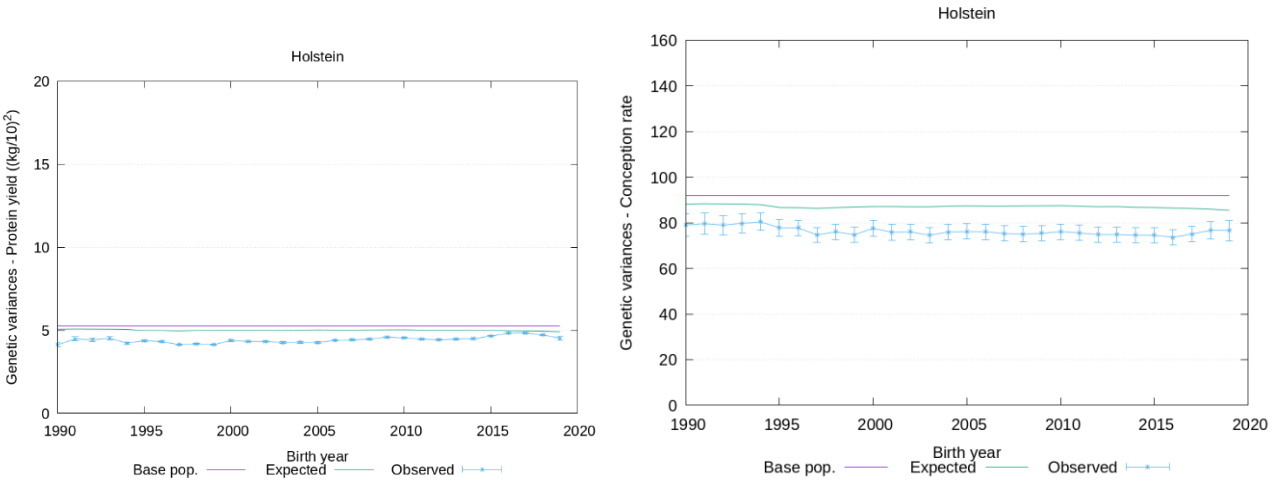

Figure 20. *Evolution of genetic variance for 305-day protein yield (PY; in $(kg/10)^2$) and conception rate (CR) along birth year for Dutch Holstein first-parity cows. Depicted genetic variances are genetic variances at the base population, expected genetic variances, and the observed genetic variances. Posterior standard deviations are represented by error bars.*

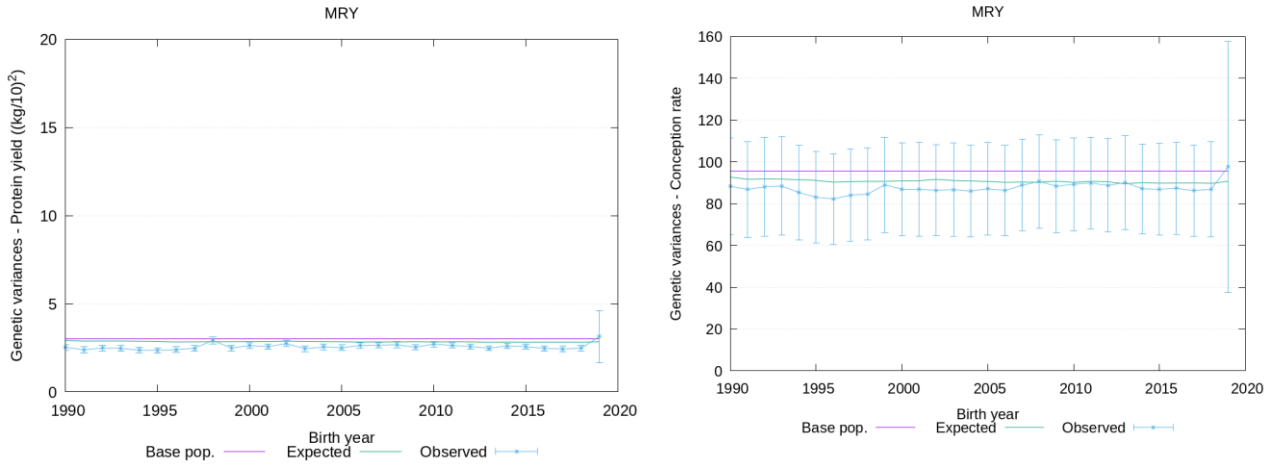

Figure 21. *Evolution of genetic variance for 305-day protein yield (PY; in (kg/10)^2) and conception rate (CR) along birth year for MRY first-parity cows. Depicted genetic variances are genetic variances at the base population, expected genetic variances, and the observed genetic variances. Posterior standard deviations are represented by error bars.*

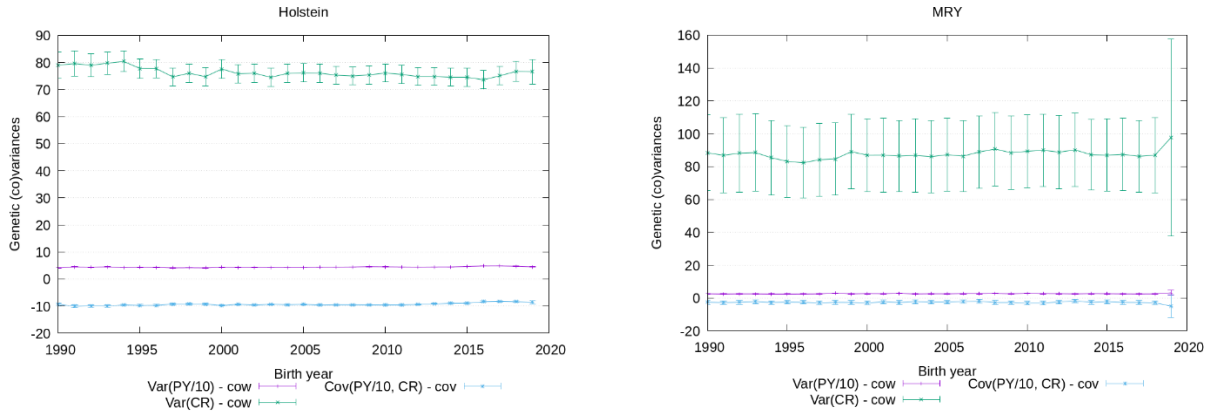

Figure 22. *Evolution of estimates of observed genetic (co)variances between 305-day protein yield (PY; in (kg/10)^2) and conception rate (CR) along birth years for Dutch Holstein and MRY first-parity cows. Posterior standard deviations are represented by error bars.*

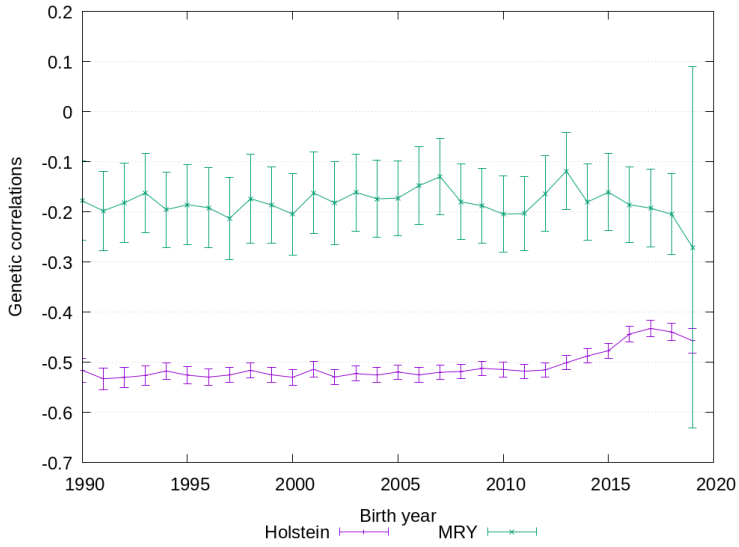

Figure 23. *Evolution of estimates of observed genetic correlations between 305-day protein yield and conception rate along birth years for Dutch Holstein and MRY first-parity cows. Posterior standard deviations are represented by error bars.*

Appendix 3: R-scripts to perform the simulations considering gametic Mendelian (co)variances

R-script for the unselected scenario:

```
setwd("…")
################################################################################
# Script to simulate and select on Mendelian sampling covariance terms
#  Context:
#    - Task 8.2 of Rumigen
#    - Assuming the initial population is unselected (and thus not in Bulmer equilibrium)
#    - Based on the simulations in Bijma et al 2020, with a few modifications:
#                - Two traits instead of one trait
#                - Gametic variances follow a normal instead of a log-normal distribution
#                - Mendelian correlations are based on Bonk et al 2016
#
# Mario Calus, November 2023
################################################################################

#############function for selection intensity (i), trunc piont (x) and var red factor (k) #########
ixk <- function(p) {
  x<-qnorm(1-p)        #truncation point
  i<-dnorm(x)/p        #selection intensity
  k<-i*(i-x)           #variance reduction coefficient
  return(c(i,x,k))
  }
################################################################################
#### Input parameters
nrep=10000
npar=100
vEBV=diag(1,2)                  #variance on EBV
rg=-0.3                         #genetic correlation
vEBV[2,1]<-vEBV[1,2]<-rg
vgam=0.25*vEBV                  #mean variance on a gamete
m=vgam[1,1]                     #mean variance on a gamete
nindx<-4                        #number of different selection indices to be compared. 1=random;
2=EBV; 3=Index5 without GMC; 4=Index5 with GMC
results<-data.frame()           #to store the results

### Directory & outputfile:
#setwd("…")
outfile<-"results_selindices.txt"

########START looping over the alternative schemes#####################################
##NOTE: the p applies to both the parent and the offspring generation, same p
################################
##but in parent generation npar is exact, whereas in offspring it is the number exceeding the
threshold####
##belonging to a fraction p selected based on the GEBV (I1)
################################################
#cvs<-c(0.000001,0.1,0.2,0.3,0.4)
```

```
ps<-c(0.1,0.01,0.001)
alt<-0                                  #counter for alternatives
for (p_alt in 1:length(ps)) {
  alt<-alt+1
  p<-ps[p_alt]
  ncand<-npar/p                         #number of candidates
  noff=2*ncand/npar   #number of offspring
each index
Gavg_off<-matrix(nrow=nrep,ncol=nindx)
rel_gain<-matrix(nrow=nrep,ncol=nindx)      #how much gain is higher than gain with selection on
EBV, last gen.
rg_ao<-matrix(nrow=nrep,ncol=nindx)         #rg observed among all offspring
rg_so<-matrix(nrow=nrep,ncol=nindx)         #rg observed among selected offspring


########LOOP over replicates############################
for (jrep in 1:nrep) {

#Draw EBV from multivariate distribution
RN=cbind(rnorm(ncand,0,1),rnorm(ncand,0,1))
EBV=RN%*%chol(vEBV)

#Draw gametic varianes from multivariate distribution
GAMvar<-array(0,c(ncand,9))
cvs=0.2 #This corresponds to the value of 0.1, found by Segelke et al. (2014)
cv_vgam=cvs
#mu & sigma are the same for both traits, since all parameters involved are the same
v=cv_vgam^2*vgam[1,1]^2
#draw Mendelian correlations from a norm. distr, with values ranging between rg-0.5 and rg+0.5
Mcor<-rnorm(ncand,rg,0.5/4)
Mcor[which(Mcor<(-0.95),arr.ind = TRUE)]=-0.95
Mcor[which(Mcor>0.95,arr.ind = TRUE)]=0.95
#Draw random numbers to be used to draw GAMvar from a multivariate distribution
RN=cbind(rnorm(ncand,0,1),rnorm(ncand,0,1))
#Initialize matrix with variances of Mendelian gametic variances
v_vgam<-diag(v,2)
GAMvar_tmp<-array(0,2)
for (jcand in 1:ncand){
v_vgam[1,2]<-v_vgam[2,1]<-Mcor[jcand]*v
GAMvar_tmp[1:2]=RN[jcand,]%*%chol(v_vgam)    #both GAM variances for each trait separately (in
the index the sum of SDs will be used)
 #Add a mean of m=0.25 to all variances
 GAMvar_tmp=GAMvar_tmp+m
 #Set negative values to 0.001, just to be sure (although it seems this does not happen...)
 GAMvar_tmp[GAMvar_tmp<0]=0.001
 GAMvar[jcand,1:2]=GAMvar_tmp
 GAMvar[jcand,3]=Mcor[jcand]*sqrt(GAMvar[jcand,1]*GAMvar[jcand,2])
 GAMvar[jcand,4]=sum(GAMvar[jcand,1:2])          #sum of both GAM variances (in the index the
sqrt of this sum will be used)
 #Note that in the following the GAM covariances does not belong to the sample, but to the
distribution from which the sample was drawn.
 GAMvar[jcand,5]=sum(GAMvar[jcand,1:2])+2*GAMvar[jcand,3]          #sum of both GAM
variances and the GAM covariances (in the index the sqrt of this sum will be used)
 #Take Cholesky decomposition of Gametic (co)variance matrix
```

```
 cholGamvar <-
chol(matrix(c(GAMvar[jcand,1],GAMvar[jcand,3],GAMvar[jcand,3],GAMvar[jcand,2]),2,2))
 #Store Cholesky decomposition as a 1x4 vector, such that gametes can be drawn by a vector-vector
(instead of matrix-vector) multiplication.
 GAMvar[jcand,6:7]=cholGamvar[,1]
 GAMvar[jcand,8:9]=cholGamvar[,2]
}
#Compute gametic SDs
#GAMSD<-sqrt(GAMvar)
#Put everything together, including an ID
id=seq(1:ncand)
a<-cbind(EBV,GAMvar[,1:3],sqrt(GAMvar[,4:5]),GAMvar[,6:9])
candidates<-data.frame(id,a)                    #vector with id, GEBV and gametic
variance for each selection candidate

########get expected selection threshold for the offspring generation, and mean selected parents;
same for all indices############
#p<-0.1
intxk<-ixk(p);i<-intxk[1];x<-intxk[2];k<-intxk[3]
#EvEBV_off<-(0.5*(1-k)+0.5)*vEBV          #Expected variance of offspring EBV after Bulmer
#tau_off<-mu_EBV_0+i*sqrt(vEBV)+x*sqrt(EvEBV_off)    #Expected truncation point in offspring
generation
#E_Aavg_par<-mu_EBV_0+i*sqrt(vEBV)    #Expected mean of selected parents

#Make index for each candidate
index<-matrix(nrow=dim(candidates)[1],ncol=nindx)
index[,1]<-candidates[,2]+candidates[,3] #EBV selection
index[,2]<-candidates[,2]+candidates[,3]+sqrt(2)*x*(candidates[,7]) #Index5, ignoring GMC
index[,3]<-candidates[,2]+candidates[,3]+sqrt(2)*x*(candidates[,8]) #Index5, including GMC
index[,4]<-rnorm(ncand,0,1) #random selection #candidates[,2]+candidates[,3]

#Select best parents based on the sum of BV of both traits (this mimics the assumption that both
traits have equal weights)
############now loop over the nindx indices within each replicate###############
for (jindx in 1:nindx)  {
o<-order(-1*index[,jindx])
candidates_sorted<-candidates[o,]                #from highest to lowest on this index
parents<-candidates_sorted[1:npar,]              #Select an exact fraction P
parents<-candidates[sample(parents$id,npar),]    #permute the selected parents to remove
ordering
sires<-parents[1:(npar/2),]
dams<-parents[(npar/2+1):npar,]

###########make the offspring#################################
#get sire and dam IDs
sire_ID<-rep(sires[,1],each=noff)
dam_ID<-rep(dams[,1],each=noff)
#create animal IDs
anim_ID<-seq((ncand+1),2*ncand,1)
#trait 1
### Sample the SDs from a bivariate distribution!!!!
#Draw random numbers to be used to draw gametic effects from a multivariate distribution
#For the sire
RNs=cbind(rnorm(ncand,0,1),rnorm(ncand,0,1))
```

```
#For the dam
RNd=cbind(rnorm(ncand,0,1),rnorm(ncand,0,1))
#shuf_gamd<-permute(seq(1,ncand,1))
mu_sire_gametes1<-rep(0.5*sires[,2],each=noff)              #half the ordinary EBV
mu_dam_gametes1<-rep(0.5*dams[,2],each=noff)
chol_sire_gametes1<-rep(sires[,9],each=noff) #Note that column 10 only contains 0's
chol_dam_gametes1<-rep(dams[,9],each=noff) #Note that column 10 only contains 0's
SD_sire_gametes1<-RNs[,1]*chol_sire_gametes1
SD_dam_gametes1<-RNd[,1]*chol_dam_gametes1
sire_gametes1<-mu_sire_gametes1+SD_sire_gametes1
dam_gametes1<-mu_dam_gametes1+SD_dam_gametes1
#compute the Mendelian Sampling Term:
MST1<-SD_sire_gametes1+SD_dam_gametes1 #[shuf_gamd]
EBV_off1<-sire_gametes1+dam_gametes1 #[shuf_gamd]
#trait 2
mu_sire_gametes2<-rep(0.5*sires[,3],each=noff)              #half the ordinary EBV
mu_dam_gametes2<-rep(0.5*dams[,3],each=noff)
chol_sire_gametes21<-rep(sires[,11],each=noff) #Note that column 10 only contains 0's
chol_dam_gametes21<-rep(dams[,11],each=noff) #Note that column 10 only contains 0's
chol_sire_gametes22<-rep(sires[,12],each=noff) #Note that column 10 only contains 0's
chol_dam_gametes22<-rep(dams[,12],each=noff) #Note that column 10 only contains 0's
SD_sire_gametes2<-RNs[,1]*chol_sire_gametes21+RNs[,2]*chol_sire_gametes22
SD_dam_gametes2<-RNd[,1]*chol_dam_gametes21+RNd[,2]*chol_dam_gametes22
sire_gametes2<-mu_sire_gametes2+SD_sire_gametes2
dam_gametes2<-mu_dam_gametes2+SD_dam_gametes2
#compute the Mendelian Sampling Term:
MST2<-SD_sire_gametes2+SD_dam_gametes2 #[shuf_gamd]
EBV_off2<-sire_gametes2+dam_gametes2 #[shuf_gamd]
EBV_off<-cbind(EBV_off1+EBV_off2,EBV_off1,EBV_off2)

rg_ao[jrep,jindx]<-cor(MST1,MST2)
if(jindx==4){
 o<-order(rnorm(ncand,0,1))
}else{
 o<-order(-1*EBV_off[,1])
}
EBV_off_sorted<-EBV_off[o,]
selected<-EBV_off_sorted[1:npar]
mean_off<-mean(EBV_off[,1])
Gavg_off[jrep,jindx]<-mean(selected)                        #mean GEBV of fraction P selected
rel_gain[jrep,jindx]<-100*(Gavg_off[jrep,jindx]-mean_off)/(Gavg_off[jrep,1]-mean_off)      #%relative
gain last gen compared to selection on EBV
rg_so[jrep,jindx]<-cor(MST1[o][1:npar],MST2[o][1:npar])

#write data to file:
#options(scipen = 999) #avoid that e.g. 200000 is written as 2e+05
#datafile=sprintf("data_%s_%s.txt",jindx,jrep)
#write.table(cbind(anim_ID,sire_ID,dam_ID[shuf_gamd],EBV_off1+EBV_off2,EBV_off1,EBV_off2),
#file=datafile,row.names = FALSE,col.names = FALSE)
#pedfile=sprintf("ped_%s_%s.txt",jindx,jrep)
#write.table(cbind(anim_ID,sire_ID,dam_ID[shuf_gamd]),file=pedfile,row.names                =
#FALSE,col.names = FALSE)
#options(scipen = 0)
```

```
  }       #end of indices loop
}         #end of replication

#############summarize replicates and indices for this scheme and write#########################
mu_G<-apply(Gavg_off,2,mean)
mu_relG<-apply(rel_gain,2,mean)
mu_rg_ao<-apply(rg_ao,2,mean)
mu_rg_so<-apply(rg_so,2,mean)
se_G<-apply(Gavg_off,2,sd)/sqrt(nrep)
se_relG<-apply(rel_gain,2,sd)/sqrt(nrep)
se_rg_ao<-apply(rg_ao,2,sd)/sqrt(nrep)
se_rg_so<-apply(rg_so,2,sd)/sqrt(nrep)

res<-
cbind(alt,cv_vgam,p,t(mu_G),t(mu_relG),t(mu_rg_ao),t(mu_rg_so),t(se_G),t(se_relG),t(se_rg_ao),t(se_rg_so))
results<-rbind(results,res)
}  #end of alternative schemes

names(results)<-c("alt","cv","p", #"nI1","nI2","nI3",
                  "GI1","GI2","GI3","GI4",
                  "rG1","rG2","rG3","rG4",
                  "rg(all)1","rg(all)2","rg(all)3","rg(all)4",
                  "rg(sel)1","rg(sel)2","rg(sel)3","rg(sel)4",
                  #"nSE1","nSE2","nSE3",
                  "GSE1","GSE2","GSE3","GSE4",
                  "rGSE1","rGSE2","rGSE3","rGSE4",
                  "rg(all)SE1","rg(all)SE2","rg(all)SE3","rg(all)SE4",
                  "rg(sel)SE1","rg(sel)SE2","rg(sel)SE3","rg(sel)SE4")
results<-format(results,digits=4)
write.table(results,file=outfile,append=TRUE,col.names=TRUE,row.names=FALSE,quote=FALSE)
```

R-script for the unselected scenario:

```
setwd("…")
#############################################################################
# Script to simulate and select on Mendelian sampling covariance terms
#  Context:
#    - Task 8.2 of Rumigen
#    - Assuming the initial population is selected (and thus in Bulmer equilibrium)
#    - Based on the simulations in Bijma et al 2020, with a few modifications:
#                - Two traits instead of one trait
#                - Gametic variances follow a normal instead of a log-normal distribution
#                - Mendelian correlations are based on Bonk et al 2016
#
# Mario Calus, November 2023
#############################################################################

#############function for selection intensity (i), trunc piont (x) and var red factor (k) ##########
ixk <- function(p) {
  x<-qnorm(1-p)        #truncation point
  i<-dnorm(x)/p        #selection intensity
  k<-i*(i-x)           #variance reduction coefficient
```

```
  return(c(i,x,k))
  }
#############################################################################
#### Input parameters
nrep=10000
npar=100
mu_EBV_0=0
vEBV0=diag(1,2)                          #variance on EBV
rg=-0.3                                  #genetic correlation
vEBV0[2,1]<-vEBV0[1,2]<-rg
vgam=0.25*vEBV0                          #mean variance on a gamete
m=vgam[1,1]                              #mean variance on a gamete
nindx<-4                    #number of different selection indices to be compared. 1=random;
2=EBV; 3=Index5 without GMC; 4=Index5 with GMC
results<-data.frame()       #to store the results

### Directory & outputfile:
#setwd("…")
outfile<-"results_selindices_selected_pop.txt"

########START looping over the alternative schemes#####################################
##NOTE: the p applies to both the parent and the offspring generation, same p ##############
##but in parent generation npar is exact, whereas in offspring it is the number exceeding the
threshold####
##belonging to a fraction p selected based on the GEBV (I1) ############################
#cvs<-c(0.000001,0.1,0.2,0.3,0.4)
ps<-c(0.1,0.01,0.001)
alt<-0                                   #counter for alternatives
for (p_alt in 1:length(ps)) {
  alt<-alt+1
  p<-ps[p_alt]
  ncand<-npar/p                  #number of candidates
  noff=2*ncand/npar   #number of offspring
each index
Gavg_off<-matrix(nrow=nrep,ncol=nindx)
rel_gain<-matrix(nrow=nrep,ncol=nindx)        #how much gain is higher than gain with selection
on EBV, last gen.
rg_ao<-matrix(nrow=nrep,ncol=nindx)                          #rg observed among all offspring
rg_so<-matrix(nrow=nrep,ncol=nindx)                          #rg observed among selected
offspring

########LOOP over replicates#########################
for (jrep in 1:nrep) {

#Draw EBV from multivariate distribution
#RN=cbind(rnorm(ncand,0,1),rnorm(ncand,0,1))
#EBV=RN%*%chol(vEBV)

########iterate to Bulmer equilibrium in 4 generations of selection on classical GEBV#########

vEBV<-vEBV0                                                  #start with base generation value
id<-seq(1:ncand)
#Draw EBV from multivariate distribution
RN=cbind(rnorm(ncand,0,1),rnorm(ncand,0,1))
```

```
EBV=RN%*%chol(vEBV)
#EBV <- rnorm(ncand,mu_EBV_0,sqrt(vEBV))
candidates<-data.frame(id,EBV[,1]+EBV[,2],EBV)                    #vector with id, GEBV and
gametic variance for each selection candidate
for (jgen in 1:4) {
  o<-order(-1*candidates[,2])
  candidates_sorted<-candidates[o,]                    #from highest to lowest on this index
  parents<-candidates_sorted[1:npar,]
  parents<-candidates[sample(parents$id,npar),]        #permute the parents to remove ordering
  sires<-parents[1:(npar/2),]
  dams<-parents[(npar/2+1):npar,]
  mu_sire_gametes1<-rep(0.5*sires[,3],each=noff)       #half the ordinary EBV
  mu_dam_gametes1<-rep(0.5*dams[,3],each=noff)
  mu_sire_gametes2<-rep(0.5*sires[,4],each=noff)       #half the ordinary EBV
  mu_dam_gametes2<-rep(0.5*dams[,4],each=noff)
  #Draw Mendelian Sampling terms
  RN=cbind(rnorm(ncand,0,1),rnorm(ncand,0,1))
  MS=RN%*%chol(0.5*vEBV)
  EBV_off1<-mu_sire_gametes1+mu_dam_gametes1+MS[,1]    #add simple MS-terms, MS-term
with base gen var.
  EBV_off2<-mu_sire_gametes2+mu_dam_gametes2+MS[,2]    #add simple MS-terms, MS-term
with base gen var.
  EBV_off<-EBV_off1+EBV_off2
  id<-seq(1:ncand)
  candidates<-data.frame(id,EBV_off,EBV_off1,EBV_off2)
}
mu_EBV<-mean(candidates[,2])     #account for gain due to first generations
vEBV<-var(candidates[,2])        #empirical variance of GEBV
#EvEBV_off<-(1/(1+k))*vEBV0      #Expected variance of offspring GEBV at Bulmer equilibrium
#tau_off<-mu_EBV+i*sqrt(vEBV)+x*sqrt(EvEBV_off)    #Expected truncation point in offspring
generation
#E_Aavg_par<-mu_EBV+i*sqrt(vEBV)      #Expected mean of selected parents
##############################################################################################

#Draw gametic varianes from multivariate distribution
GAMvar<-array(0,c(ncand,9))
cvs=0.2 #This corresponds to the value of 0.1, found by Segelke (?)
cv_vgam=cvs
#mu & sigma are the same for both traits, since all parameters involved are the same
v=cv_vgam^2*vgam[1,1]^2
#draw Mendelian correlations from a normal distribution with values ranging between rg-0.5 and
rg+0.5
Mcor<-rnorm(ncand,rg,0.5/4)
Mcor[which(Mcor<(-0.95),arr.ind = TRUE)]=-0.95
Mcor[which(Mcor>0.95,arr.ind = TRUE)]=0.95
#Draw random numbers to be used to draw GAMvar from a multivariate distribution
RN=cbind(rnorm(ncand,0,1),rnorm(ncand,0,1))
#Initialize matrix with variances of Mendelian gametic variances
v_vgam<-diag(v,2)
GAMvar_tmp<-array(0,2)
for (jcand in 1:ncand){
 #GAMvar <- rnorm(1,c(vgam,vgam),sigma_log)
 v_vgam[1,2]<-v_vgam[2,1]<-Mcor[jcand]*v
# cat(v_vgam, file = "tmp", sep = " ", fill = FALSE, labels = NULL, append = FALSE)
```

```
 GAMvar_tmp[1:2]=RN[jcand,]%*%chol(v_vgam)    #both GAM variances for each trait separately (in the index the sum of SDs will be used)
 #Add a mean of m=0.25 to all variances
 GAMvar_tmp=GAMvar_tmp+m
 #Set negative values to 0.001, just to be sure (although it seems this does not happen...)
 GAMvar_tmp[GAMvar_tmp<0]=0.001
 GAMvar[jcand,1:2]=GAMvar_tmp
 GAMvar[jcand,3]=Mcor[jcand]*sqrt(GAMvar[jcand,1]*GAMvar[jcand,2])
 GAMvar[jcand,4]=sum(GAMvar[jcand,1:2])          #sum of both GAM variances (in the index the sqrt of this sum will be used)
 #Note that in the following the GAM covariances does not belong to the sample, but to the distribution from which the sample was drawn.
 GAMvar[jcand,5]=sum(GAMvar[jcand,1:2])+2*GAMvar[jcand,3]          #sum of both GAM variances and the GAM covariances (in the index the sqrt of this sum will be used)
 #Take Cholesky decomposition of Gametic (co)variance matrix
cholGamvar                                                                    <-chol(matrix(c(GAMvar[jcand,1],GAMvar[jcand,3],GAMvar[jcand,3],GAMvar[jcand,2]),2,2))
 #Store Cholesky decomposition as a 1x4 vector, such that gametes can be drawn by a vector-vector (instead of matrix-vector) multiplication.
 GAMvar[jcand,6:7]=cholGamvar[,1]
 GAMvar[jcand,8:9]=cholGamvar[,2]
}
#Put everything together, including an ID
id=seq(1:ncand)
a<-cbind(EBV,GAMvar[,1:3],sqrt(GAMvar[,4:5]),GAMvar[,6:9])
candidates<-data.frame(id,a)                    #vector with id, GEBV and gametic variance for each selection candidate

########get expected selection threshold for the offspring generation, and mean selected parents; same for all indices#############
intxk<-ixk(p);i<-intxk[1];x<-intxk[2];k<-intxk[3]

#Make index for each candidate
index<-matrix(nrow=dim(candidates)[1],ncol=nindx)
index[,1]<-candidates[,2]+candidates[,3] #EBV selection
index[,2]<-candidates[,2]+candidates[,3]+sqrt(2)*x*(candidates[,7]) #Index5, ignoring GMC
index[,3]<-candidates[,2]+candidates[,3]+sqrt(2)*x*(candidates[,8]) #Index5, including GMC
index[,4]<-rnorm(ncand,0,1) #random selection #candidates[,2]+candidates[,3]

#Select best parents based on the sum of BV of both traits (this mimics the assumption that both traits have equal weights)
############now loop over the nindx indices within each replicate###############
for (jindx in 1:nindx)  {
o<-order(-1*index[,jindx])
candidates_sorted<-candidates[o,]                    #from highest to lowest on this index
parents<-candidates_sorted[1:npar,]                  #Select an exact fraction P
parents<-candidates[sample(parents$id,npar),]        #permute the selected parents to remove ordering
sires<-parents[1:(npar/2),]
dams<-parents[(npar/2+1):npar,]

##########make the offspring#################################
#get sire and dam IDs
sire_ID<-rep(sires[,1],each=noff)
```

```
dam_ID<-rep(dams[,1],each=noff)
#create animal IDs
anim_ID<-seq((ncand+1),2*ncand,1)
#trait 1
### Sample the SDs from a bivariate distribution!!!!
#Draw random numbers to be used to draw gametic effects from a multivariate distribution
#For the sire
RNs=cbind(rnorm(ncand,0,1),rnorm(ncand,0,1))
#For the dam
RNd=cbind(rnorm(ncand,0,1),rnorm(ncand,0,1))
#shuf_gamd<-permute(seq(1,ncand,1))
mu_sire_gametes1<-rep(0.5*sires[,2],each=noff)              #half the ordinary EBV
mu_dam_gametes1<-rep(0.5*dams[,2],each=noff)
#sd_sire_gametes1<-rep(sires[,4],each=noff)
#sd_dam_gametes1<-rep(dams[,4],each=noff)
chol_sire_gametes1<-rep(sires[,9],each=noff) #Note that column 10 only contains 0's
chol_dam_gametes1<-rep(dams[,9],each=noff) #Note that column 10 only contains 0's
SD_sire_gametes1<-RNs[,1]*chol_sire_gametes1
SD_dam_gametes1<-RNd[,1]*chol_dam_gametes1
sire_gametes1<-mu_sire_gametes1+SD_sire_gametes1
dam_gametes1<-mu_dam_gametes1+SD_dam_gametes1
#compute the Mendelian Sampling Term:
MST1<-SD_sire_gametes1+SD_dam_gametes1 #[shuf_gamd]
EBV_off1<-sire_gametes1+dam_gametes1 #[shuf_gamd]
#trait 2
mu_sire_gametes2<-rep(0.5*sires[,3],each=noff)              #half the ordinary EBV
mu_dam_gametes2<-rep(0.5*dams[,3],each=noff)
chol_sire_gametes21<-rep(sires[,11],each=noff) #Note that column 10 only contains 0's
chol_dam_gametes21<-rep(dams[,11],each=noff) #Note that column 10 only contains 0's
chol_sire_gametes22<-rep(sires[,12],each=noff) #Note that column 10 only contains 0's
chol_dam_gametes22<-rep(dams[,12],each=noff) #Note that column 10 only contains 0's
SD_sire_gametes2<-RNs[,1]*chol_sire_gametes21+RNs[,2]*chol_sire_gametes22
SD_dam_gametes2<-RNd[,1]*chol_dam_gametes21+RNd[,2]*chol_dam_gametes22
sire_gametes2<-mu_sire_gametes2+SD_sire_gametes2
dam_gametes2<-mu_dam_gametes2+SD_dam_gametes2
#compute the Mendelian Sampling Term:
MST2<-SD_sire_gametes2+SD_dam_gametes2 #[shuf_gamd]
EBV_off2<-sire_gametes2+dam_gametes2 #[shuf_gamd]
EBV_off<-cbind(EBV_off1+EBV_off2,EBV_off1,EBV_off2)

#rg_ao[jrep,jindx]<-cor(cbind(EBV_off1,EBV_off2))[2,1]
rg_ao[jrep,jindx]<-cor(MST1,MST2)
if(jindx==4){
 o<-order(rnorm(ncand,0,1))
}else{
 o<-order(-1*EBV_off[,1])
}
EBV_off_sorted<-EBV_off[o,]
selected<-EBV_off_sorted[1:npar]
mean_off<-mean(EBV_off[,1])
Gavg_off[jrep,jindx]<-mean(selected)                        #mean GEBV of fraction P selected
exceeding the threshold corresponding to a fraction P selected with selection on the GEBV
rel_gain[jrep,jindx]<-100*(Gavg_off[jrep,jindx]-mean_off)/(Gavg_off[jrep,1]-mean_off)      #%relative
gain last gen compared to selection on EBV
```

```
rg_so[jrep,jindx]<-cor(MST1[o][1:npar],MST2[o][1:npar])

#write data to file:
#options(scipen = 999) #avoid that e.g. 200000 is written as 2e+05
#datafile=sprintf("data_%s_%s.txt",jindx,jrep)
#write.table(cbind(anim_ID,sire_ID,dam_ID[shuf_gamd],EBV_off1+EBV_off2,EBV_off1,EBV_off2),
#file=datafile,row.names = FALSE,col.names = FALSE)
#pedfile=sprintf("ped_%s_%s.txt",jindx,jrep)
#write.table(cbind(anim_ID,sire_ID,dam_ID[shuf_gamd]),file=pedfile,row.names                    =
#FALSE,col.names = FALSE)
#options(scipen = 0)

  } #end of indices loop
}         #end of replication

##############summarize replicates and indices for this scheme and write##################
mu_G<-apply(Gavg_off,2,mean)
mu_relG<-apply(rel_gain,2,mean)
mu_rg_ao<-apply(rg_ao,2,mean)
mu_rg_so<-apply(rg_so,2,mean)
se_G<-apply(Gavg_off,2,sd)/sqrt(nrep)
se_relG<-apply(rel_gain,2,sd)/sqrt(nrep)
se_rg_ao<-apply(rg_ao,2,sd)/sqrt(nrep)
se_rg_so<-apply(rg_so,2,sd)/sqrt(nrep)

res<-
cbind(alt,cv_vgam,p,t(mu_G),t(mu_relG),t(mu_rg_ao),t(mu_rg_so),t(se_G),t(se_relG),t(se_rg_ao),t
(se_rg_so))
results<-rbind(results,res)
}  #end of alternative schemes

names(results)<-c("alt","cv","p", #"nI1","nI2","nI3",
                  "GI1","GI2","GI3","GI4",
                  "rG1","rG2","rG3","rG4",
                  "rg(all)1","rg(all)2","rg(all)3","rg(all)4",
                  "rg(sel)1","rg(sel)2","rg(sel)3","rg(sel)4",
                  #"nSE1","nSE2","nSE3",
                  "GSE1","GSE2","GSE3","GSE4",
                  "rGSE1","rGSE2","rGSE3","rGSE4",
                  "rg(all)SE1","rg(all)SE2","rg(all)SE3","rg(all)SE4",
                  "rg(sel)SE1","rg(sel)SE2","rg(sel)SE3","rg(sel)SE4")
results<-format(results,digits=4)
write.table(results,file=outfile,append=TRUE,col.names=TRUE,row.names=FALSE,quote=FALSE)
```

Appendix 4: R-scripts to simulate selection strategies to increase resilience to climate change using MoBPS

R-script for the simulation of scenario 1 (see Table 4 for details):

```
library(MoBPS)
library(MoBPSmaps)
rm(list = ls())
args = commandArgs(TRUE)
args = c(1:50)
setwd("/Heat_stress_1/")
mixblup.path = "/shared/legacyapps/mixblup/current/MiXBLUP.exe"

for (args in 1:50){
  temp1 = as.numeric(args)
  nr = as.numeric(args)
  seed = as.numeric(args)
  set.seed(seed)
  n_gen = 20 # number of generation to simulate
  h2 = c(0.046, 0.24) # heritability of the traits
  n_cow = 5000 # number of cows generated per generation
  n_bull = 5000 # number of bulls generated per generation
  n_bull_sel = 250 # number of bulls used as sires

  # N.B For cows, we assume one offspring per animal (i.e no embryo transfer, euro donors etc)
  # Animals from the last 3 generations are used as selection candidates

  # Generation of the founder population and traits
  map = MoBPSmaps::map_cattle1
  ## Scenario 1 is a current selection method without considering THI while scenario 2 considers
THI
    population <- creating.diploid(map=map,
                                   nindi= 1000,
                                   n.additive = rep(1000,2),
                                   trait.name = c("C.Rate", "Protein yield"),
                                   mean.target = c(46,991),
                                   var.target = c(28, 0.45)*4,
                                   trait.cor = matrix(c(1, 0.01, 0.01, 1 ),nrow =2),
                                   name.cohort = "Founder",
                                   progress.bar = FALSE)  ## TP: deactivate progress bars for a
cleaner log-file on server

   population <- breeding.diploid(population, heritability = h2)

# Generation of initial cohorts from random mating of founder population
    for(cycle in (-2):0){
      population = breeding.diploid(population, breeding.size = c(0, n_cow),
                                    selection.m.cohorts = "Founder_M",
                                    selection.f.cohorts = "Founder_F",
                                    name.cohort = c(paste0("Cow_", cycle)),
                                    add.gen = 2,
                                    time.point = 0,
                                    display.progress = FALSE)

      population = breeding.diploid(population, breeding.size = c(n_bull, 0),
                                    selection.m.cohorts = "Founder_M",
                                    selection.f.cohorts = "Founder_F",
                                    name.cohort = c(paste0("Bull_", cycle)),
                                    add.gen = 2,
                                    display.progress = FALSE,
                                    time.point = 0)
    }
```

```
    for(gen in 1:n_gen){
      # phenotyping of animals are completed their first lactation (2 years back)
      population <- breeding.diploid(population,
                                     phenotyping.cohorts = paste0("Cow_", gen-1:2))

      pheno = get.pheno(population, cohorts = paste0("Cow_", gen-1:2))

     ## Convert the phenotype for C.rate to 100 and 0

     index = order(as.numeric(pheno[1,]), decreasing = TRUE)
     pheno= pheno[,index]
     pointer = as.numeric(length(pheno[1,])* (46/100))
     pheno[1,][1: pointer] = 100
     pheno[1,][(pointer + 1): length(pheno[1,])] = 0
     pheno = pheno[,order(colnames(pheno))]
     new_pheno = cbind(colnames(pheno), t(pheno))

     population = insert.pheno(population, phenos = new_pheno)

    population = breeding.diploid(population, bve= TRUE, bve.cohorts = c(paste0("Cow_", gen-1:3),
                                                                         paste0("Bull_", gen-1:3)),
                mixblup.bve = TRUE, mixblup.path = mixblup.path, mixblup.files = "MiXBLUP_files")

    print(analyze.bv(population,cohorts = c(paste0("Bull_", gen-1:3),
                                    paste0("Cow_", gen-1:3))))

    print(analyze.bv(population,cohorts = c(paste0("Cow_", gen-1:2))))

    population = breeding.diploid(population,
                                    breeding.size = c(n_bull, n_cow),
                                    selection.size = c(n_bull_sel, n_bull + n_cow),
                                    selection.m.cohorts = paste0("Bull_", c(gen-1, gen-2, gen-3)),
                                    selection.f.cohorts = paste0("Cow_", c(gen-1, gen-2, gen-3)),
                                    max.offspring = c(Inf, 1),
                                    selection.criteria = 'bve',
                                    multiple.bve.weights.m = c(1, 1.79),
                                    display.progress = FALSE,
                                    name.cohort = c(paste0("Bull_", gen), paste0("Cow_", gen)))


}


    bv = matrix(0, nrow = 2, ncol = get.ngen(population))
    kin = matrix(0, nrow = 2, ncol =get.ngen(population))

    for(index in 1:get.ngen(population)){
      bv[,index] = rowMeans(get.bv(population, gen =index))
    }

    for(index2 in 1:get.ngen(population)){
      kin[,index2] = kinship.emp.fast(population, gen = index2)
    }

    save(file=paste0("heat_analysis_", "1_", nr, ".RData"), list = c("bv", "kin"))

 }
```

R-script for the simulation of scenario 2 (see Table 4 for details):

```
library(MoBPS)
library(MoBPSmaps)
library(msm)
rm(list = ls())
args = commandArgs(TRUE)
args = c(1:50)
setwd("/Heat_stress_2/")
mixblup.path = "/shared/legacyapps/mixblup/current/MiXBLUP.exe"

for (args in 1:50){
  nr = as.numeric(args)
  seed = as.numeric(args)
  set.seed(seed)
  n_gen = 20 # number of generation to simulate
  h2 = c(0.046, 1, 0.24, 1, 1) # heritability of the traits
  n_cow = 5000 # number of cows generated per generation
  n_bull = 5000 # number of bulls generated per generation
  n_bull_sel = 250 # number of bulls used as sires
  trait_name = c("C.Rate", "C.R_slope", "Protein yield", "Protein_slope", "Protein yield_2")
  trait_cor = matrix(c(1, -0.04, 0.01, 0.48, -0.40,
                       -0.04, 1,-0.61,-0.76,0.70,
                       0.01,-0.61,1,0.59,-0.51,
                       0.48,-0.76,0.59,1,-0.47,
                       -0.40,0.70,-0.51,-0.47,1),nrow =5)

  # N.B For cows, we assume one offspring per animal (i.e no embryo transfer, euro donors etc)
  # Animals from the last 3 generations are used as selection candidates

  # Generation of the founder population and traits
  map = MoBPSmaps::map_cattle1

  population = creating.diploid(nindi = 1000, map=map,
                                name.cohort = "Founder",
                                progress.bar = FALSE)

  population = creating.trait(population, n.additive = rep(1000,5),
                              trait.name = trait_name,
                              mean.target = c(46, 0, 991, 0, 0),
                              var.target = c(28, 20, 0.45, 0.004, 0.007)*4,
                              trait.cor = trait_cor)

  population = breeding.diploid(population, heritability = h2 )

  # Generation of initial cohorts from random mating of founder population
  for(cycle in (-2):0){
    population = breeding.diploid(population, breeding.size = c(0, n_cow),
                                  selection.m.cohorts = "Founder_M",
                                  selection.f.cohorts = "Founder_F",
                                  name.cohort = c(paste0("Cow_", cycle)),
                                  add.gen = 2,
                                  time.point = 0,
                                  display.progress = FALSE)

    population = breeding.diploid(population, breeding.size = c(n_bull, 0),
                                  selection.m.cohorts = "Founder_M",
                                  selection.f.cohorts = "Founder_F",
                                  name.cohort = c(paste0("Bull_", cycle)),
                                  add.gen = 2,
                                  display.progress = FALSE,
                                  time.point = 0)
  }

  for(gen in 1:n_gen){
```

```
    # phenotyping of animals that have completed their first lactation (2 years back)
    population <- breeding.diploid(population,
                                   phenotyping.cohorts = paste0("Cow_", gen-2))

    phenotypes_raw = get.pheno(population, cohorts = paste0("Cow_", gen-2))
    bv_raw = get.bv(population, cohorts = paste0("Cow_",gen-2))

    thi = rtnorm(5000, 53, sd =9.76, lower = 14, upper = 78)
    thi2 = (thi-(83+10)/2)/((83-10)/2)

    heat.stress = function(phenotypes_raw, bv_raw, thi, thi2){

      temp1 = thi-60
      temp1[temp1<0] = 0
      temp1 = temp1/13.75

phenotypes_raw[1,] = bv_raw[1,] + bv_raw[2,] * temp1 + rnorm(n_cow, mean = 0, sd =sqrt(4757.565))

phenotypes_raw[3,] = bv_raw[3,]*sqrt(0.5)+ bv_raw[4,]*sqrt(1.5)*(thi2) +
     bv_raw[5,]*sqrt(1.5)*sqrt(2.5)*0.5*(3*(thi2^2)-1) + rnorm(n_cow, mean = 0, sd = sqrt(5.7))

      return(phenotypes_raw)
    }


    phenotypes_heat = heat.stress(phenotypes_raw, bv_raw, thi, thi2)

    new_pheno = cbind(colnames(phenotypes_heat), t(phenotypes_heat))

    ## Convert the phenotype for C.rate to 100 and 0
    index = order(as.numeric(new_pheno[,2]), decreasing = TRUE)
    new_pheno = new_pheno[index,]
    pointer = as.numeric(length(new_pheno[,2])* (46/100))
    new_pheno[,2][1: pointer] = 100
    new_pheno[,2][(pointer + 1): length(new_pheno[,2])] = 0

    index = order(new_pheno[,1])
    new_pheno = new_pheno[index,]

    population = insert.pheno(population, phenos = new_pheno)

    # Breeding value estimation is skipped
   population = breeding.diploid(population, bve= TRUE, bve.cohorts = c(paste0("Cow_", gen-1:3),
                                                                        paste0("Bull_", gen-1:3)),
                mixblup.bve = TRUE, mixblup.path = mixblup.path, mixblup.files = "MiXBLUP_files")

    print("accuracies of the breeding value estimation:")

    print(analyze.bv(population,cohorts = c(paste0("Bull_", gen-1:3),
                                   paste0("Cow_", gen-1:3))))

     print(analyze.bv(population,cohorts = c(paste0("Cow_", gen-2))))

    population = breeding.diploid(population,
                                   breeding.size = c(n_bull, n_cow),
                                   selection.size = c(n_bull_sel, n_bull + n_cow),
                                   selection.m.cohorts = paste0("Bull_", gen-1:3),
                                   selection.f.cohorts = paste0("Cow_", gen-1:3),
                                   max.offspring = c(Inf, 1),
                                   multiple.bve.weights.m = c(1, 0, 1.79, 0, 0),
                                   display.progress = FALSE,
                                   name.cohort = c(paste0("Bull_", gen), paste0("Cow_", gen)))

  }
```

```
  bv = matrix(0, nrow = 5, ncol = get.ngen(population))
  kin = matrix(0, nrow = 2, ncol =get.ngen(population))

  for(index in 1:get.ngen(population)){
    bv[,index] = rowMeans(get.bv(population, gen =index))
  }

  for(index2 in 1:get.ngen(population)){
  kin[,index2] = kinship.emp.fast(population, gen = index2)
  }

  save(file=paste0("heat_analysis_", "2_", nr, ".RData"), list = c("bv", "kin"))

  }
```

R-script for the simulation of scenario 3 (see Table 4 for details):

```
library(MoBPS)
library(MoBPSmaps)
library(msm)
rm(list = ls())
args = commandArgs(TRUE)
args = c(1:50)
setwd("/Heat_stress_3/")
mixblup.path = "/shared/legacyapps/mixblup/current/MiXBLUP.exe"

for (args in 1:50){
  nr = as.numeric(args)
  seed = as.numeric(args)
  set.seed(seed)
  n_gen = 20 # number of generation to simulate
  h2 = c(0.046, 1, 0.24, 1, 1) # heritability of the traits
  n_cow = 5000 # number of cows generated per generation
  n_bull = 5000 # number of bulls generated per generation
  n_bull_sel = 250 # number of bulls used as sires
  trait_name = c("C.Rate", "C.R_slope", "Protein yield", "Protein_slope", "Protein yield_2")
  trait_cor = matrix(c(1, -0.04, 0.01, 0.48, -0.40,
                       -0.04, 1,-0.61,-0.76,0.70,
                       0.01,-0.61,1,0.59,-0.51,
                       0.48,-0.76,0.59,1,-0.47,
                       -0.40,0.70,-0.51,-0.47,1), nrow = 5)

  # N.B For cows, we assume one offspring per animal (i.e no embryo transfer, euro donors etc)
  # Animals from the last 3 generations are used as selection candidates

  # Generation of the founder population and traits
  map = MoBPSmaps::map_cattle1

  population = creating.diploid(nindi = 1000, map=map,
                                name.cohort = "Founder",
                                progress.bar = FALSE)

  population = creating.trait(population, n.additive = rep(1000,5),
                              trait.name = trait_name,
                              mean.target = c(46, 0, 991, 0, 0),
                              var.target = c(112, 80, 1.8, 0.016, 0.028),
                              trait.cor = trait_cor)

  population = breeding.diploid(population, heritability = h2 )

  # Generation of initial cohorts from random mating of founder population
  for(cycle in (-2):0){
    population = breeding.diploid(population, breeding.size = c(0, n_cow),
```

```
                                  selection.m.cohorts = "Founder_M",
                                  selection.f.cohorts = "Founder_F",
                                  name.cohort = c(paste0("Cow_", cycle)),
                                  add.gen = 2,
                                  time.point = 0,
                                  display.progress = FALSE)

    population = breeding.diploid(population, breeding.size = c(n_bull, 0),
                                  selection.m.cohorts = "Founder_M",
                                  selection.f.cohorts = "Founder_F",
                                  name.cohort = c(paste0("Bull_", cycle)),
                                  add.gen = 2,
                                  display.progress = FALSE,
                                  time.point = 0)
  }

  for(gen in 1:n_gen){


    # phenotyping of animals that have completed their first lactation (2 years back)
    population <- breeding.diploid(population,
                                   phenotyping.cohorts = paste0("Cow_", gen-1:2))

    phenotypes_raw = get.pheno(population, cohorts = paste0("Cow_", gen-1:2))
    bv_raw = get.bv(population, cohorts = paste0("Cow_",gen-1:2))


      thi = rtnorm(dim(phenotypes_raw)[2], 53, sd =9.76, lower = 14, upper = 78)
      thi2 = (thi-(83+10)/2)/((83-10)/2)

      cov1 = thi-60
      cov1[cov1<0] = 0
      cov1 = cov1/13.75
      cov2 = rep(sqrt(0.5), times = 10000)
      cov3 = sqrt(1.5)*(thi2)
      cov4 = sqrt(1.5)*sqrt(2.5)*0.5*(3*(thi2^2)-1)

      phenotypes_raw[1,] = bv_raw[1,] + bv_raw[2,] * cov1 + rnorm(10000, mean = 0, sd =
sqrt(4757.565))

        phenotypes_raw[3,] = bv_raw[3,]*cov2 + bv_raw[4,]*cov3 +
        bv_raw[5,]*cov4 + rnorm(10000, mean = 0, sd = sqrt(5.7))

    new_pheno = cbind(colnames(phenotypes_raw), t(phenotypes_raw))

    ## Convert the phenotype for C.rate to 100 and 0
    index1 = order(as.numeric(new_pheno[,2]), decreasing = TRUE)
    new_pheno = new_pheno[index1,]
    pointer = as.numeric(length(new_pheno[,2])* (46/100))
    new_pheno[,2][1:pointer] = 100
    new_pheno[,2][(pointer + 1): length(new_pheno[,2])] = 0

    index2 = order(new_pheno[,1])
    new_pheno = new_pheno[index2,]

    population = insert.pheno(population, phenos = new_pheno)


    # Breeding value estimation is skipped
    population = breeding.diploid(population, bve = TRUE, mixblup.bve = TRUE, mixblup.path =
mixblup.path, mixblup.files = "MiXBLUP_files", mixblup.skip = TRUE,
                                  bve.ignore.traits = c(2,4,5), #mixblup.multiple.records = TRUE,
                                  bve.cohorts =  c(paste0("Bull_", gen-1:3),
                                                   paste0("Cow_", gen-1:3)))
```

```
    setwd("/Heat_stress_3/MiXBLUP_files")
    data = read.table("data.txt")
    data = data[c(20001:30000),]
    data$V5 = cov1
    data$V6 = cov2
    data$V7 = cov3
    data$V8 = cov4

    write.table(data, "data.txt", row.names = F, col.names = F)

    setwd("/Heat_stress_3/")
    system("/shared/legacyapps/mixblup/current/MiXBLUP.exe MiXBLUP_files/InpMiXBLUP_1.txt")

    id = get.id(population, cohorts = c(paste0("Bull_", gen-1:3),
                                        paste0("Cow_", gen-1:3)))

    tbv = get.bv(population, cohorts = c(paste0("Bull_", gen-1:3),
                                         paste0("Cow_", gen-1:3)))
    tbv = t(tbv)
    id_tbv = merge(id, tbv, by = "row.names", sort =F)[,c(1,2)]

    ebv = as.matrix(read.table("Solani.txt"))
    ebv = ebv[,-c(2,3)]
    ebv = ebv[,c(1,2,4,3,5,6)] # changing the order of the breeding values from mixblup
    ebv = merge(id_tbv, ebv, by.x = "x", by.y = "V1", sort = F)[,-1]

    ## coef below the threshold
    cof = matrix(c(1, 0, 0, 0, 0,
                   0, 0, 0.7071068, 0.1174413, -0.9415369) ,nrow = 5, ncol = 2)

    bves_matrix = as.matrix(ebv[,c(2:6)]) %*% cof
    bves = cbind(ebv[,-c(2,4)], bves_matrix)
    bves = bves[, c(1, 5, 2, 6, 3, 4)]
    colnames(bves)= c(" ", "C.Rate", "C.R_slope", "Protein yield", "Protein_slope", "Protein
yield_2")
    population = insert.bve(population, bves = bves)

    print("accuracies of the breeding value estimation:")

    print(analyze.bv(population,cohorts = c(paste0("Bull_", gen-1:3),
                                            paste0("Cow_", gen-1:3))))

    print(analyze.bv(population,cohorts = c(paste0("Cow_", gen-2))))

    population = breeding.diploid(population,
                                  breeding.size = c(n_bull, n_cow),
                                  selection.size = c(n_bull_sel, n_bull + n_cow),
                                  selection.m.cohorts = paste0("Bull_", gen-1:3),
                                  selection.f.cohorts = paste0("Cow_", gen-1:3),
                                  max.offspring = c(Inf, 1),
                                  selection.criteria = 'bve',
                                  multiple.bve.weights.m = c(1, 0, 1.79, 0, 0),
                                  display.progress = FALSE,
                                  name.cohort = c(paste0("Bull_", gen), paste0("Cow_", gen)))

  }



  bv = matrix(0, nrow = 5, ncol = get.ngen(population))
  kin = matrix(0, nrow = 2, ncol =get.ngen(population))

  for(index in 1:get.ngen(population)){
    bv[,index] = rowMeans(get.bv(population, gen =index))
```

```
  }

  for(index2 in 1:get.ngen(population)){
    kin[,index2] = kinship.emp.fast(population, gen = index2)
  }

  save(file=paste0("heat_analysis_", "3_", nr, ".RData"), list = c("bv", "kin"))

}
```

R-script for the simulation of scenario 4 (see Table 4 for details)

```
library(MoBPS)
library(MoBPSmaps)
library(msm)
rm(list = ls())
args = commandArgs(TRUE)
args = c(1:50)
setwd("/Heat_stress_4/")
mixblup.path = "/shared/legacyapps/mixblup/current/MiXBLUP.exe"

for (args in 1:50){
  nr = as.numeric(args)
  seed = as.numeric(args)
  set.seed(seed)
  n_gen = 20 # number of generation to simulate
  h2 = c(0.046, 1, 0.24, 1, 1) # heritability of the traits
  n_cow = 5000 # number of cows generated per generation
  n_bull = 5000 # number of bulls generated per generation
  n_bull_sel = 250 # number of bulls used as sires
  trait_name = c("C.Rate", "C.R_slope", "Protein yield", "Protein_slope", "Protein yield_2")
  trait_cor = matrix(c(1, -0.04, 0.01, 0.48, -0.40,
                       -0.04, 1,-0.61,-0.76,0.70,
                       0.01,-0.61,1,0.59,-0.51,
                       0.48,-0.76,0.59,1,-0.47,
                       -0.40,0.70,-0.51,-0.47,1), nrow = 5)

  # N.B For cows, we assume one offspring per animal (i.e no embryo transfer, euro donors etc)
  # Animals from the last 3 generations are used as selection candidates

  # Generation of the founder population and traits
  map = MoBPSmaps::map_cattle1

  population = creating.diploid(nindi = 1000, map=map,
                                name.cohort = "Founder",
                                progress.bar = FALSE)

  population = creating.trait(population, n.additive = rep(1000,5),
                              trait.name = trait_name,
                              mean.target = c(46, 0, 991, 0, 0),
                              var.target = c(112, 80, 1.8, 0.016, 0.028),
                              trait.cor = trait_cor)

  population = breeding.diploid(population, heritability = h2 )

  # Generation of initial cohorts from random mating of founder population
  for(cycle in (-2):0){
    population = breeding.diploid(population, breeding.size = c(0, n_cow),
                                  selection.m.cohorts = "Founder_M",
                                  selection.f.cohorts = "Founder_F",
                                  name.cohort = c(paste0("Cow_", cycle)),
                                  add.gen = 2,
                                  time.point = 0,
                                  display.progress = FALSE)
```

```
    population = breeding.diploid(population, breeding.size = c(n_bull, 0),
                                  selection.m.cohorts = "Founder_M",
                                  selection.f.cohorts = "Founder_F",
                                  name.cohort = c(paste0("Bull_", cycle)),
                                  add.gen = 2,
                                  display.progress = FALSE,
                                  time.point = 0)
  }

  for(gen in 1:n_gen){


    # phenotyping of animals that have completed their first lactation (2 years back)
    population <- breeding.diploid(population,
                                   phenotyping.cohorts = paste0("Cow_", gen-1:2))

    phenotypes_raw = get.pheno(population, cohorts = paste0("Cow_", gen-1:2))
    bv_raw = get.bv(population, cohorts = paste0("Cow_",gen-1:2))


      thi = rtnorm(dim(phenotypes_raw)[2], 53, sd =9.76, lower = 14, upper = 78)
      thi2 = (thi-(83+10)/2)/((83-10)/2)

      cov1 = thi-60
      cov1[cov1<0] = 0
      cov1 = cov1/13.75
      cov2 = rep(sqrt(0.5), times = 10000)
      cov3 = sqrt(1.5)*(thi2)
      cov4 = sqrt(1.5)*sqrt(2.5)*0.5*(3*(thi2^2)-1)

      phenotypes_raw[1,] = bv_raw[1,] + bv_raw[2,] * cov1 + rnorm(10000, mean = 0, sd =
sqrt(4757.565))

      phenotypes_raw[3,] = bv_raw[3,]*cov2 + bv_raw[4,]*cov3 +
        bv_raw[5,]*cov4 + rnorm(10000, mean = 0, sd = sqrt(5.7))

    new_pheno = cbind(colnames(phenotypes_raw), t(phenotypes_raw))

    ## Convert the phenotype for C.rate to 100 and 0
    index1 = order(as.numeric(new_pheno[,2]), decreasing = TRUE)
    new_pheno = new_pheno[index1,]
    pointer = as.numeric(length(new_pheno[,2])* (46/100))
    new_pheno[,2][1:pointer] = 100
    new_pheno[,2][(pointer + 1): length(new_pheno[,2])] = 0

    index2 = order(new_pheno[,1])
    new_pheno = new_pheno[index2,]

    population = insert.pheno(population, phenos = new_pheno)

    # Breeding value estimation is skipped
    population = breeding.diploid(population, bve = TRUE, mixblup.bve = TRUE, mixblup.path =
mixblup.path, mixblup.files = "MiXBLUP_files", mixblup.skip = TRUE,
                                  bve.ignore.traits = c(2,4,5), #mixblup.multiple.records = TRUE,
                                  bve.cohorts =  c(paste0("Bull_", gen-1:3),
                                                   paste0("Cow_", gen-1:3)))


    setwd("/Heat_stress_4/MiXBLUP_files")
    data = read.table("data.txt")
    data = data[c(20001:30000),]
    data$V5 = cov1
    data$V6 = cov2
    data$V7 = cov3
```

```
    data$V8 = cov4

    write.table(data, "data.txt", row.names = F, col.names = F)

    setwd("/Heat_stress_4/")
    system("/shared/legacyapps/mixblup/current/MiXBLUP.exe MiXBLUP_files/InpMiXBLUP_1.txt")

    id = get.id(population, cohorts = c(paste0("Bull_", gen-1:3),
                                        paste0("Cow_", gen-1:3)))

    tbv = get.bv(population, cohorts = c(paste0("Bull_", gen-1:3),
                                         paste0("Cow_", gen-1:3)))
    tbv = t(tbv)
    id_tbv = merge(id, tbv, by = "row.names", sort =F)[,c(1,2)]

    ebv = as.matrix(read.table("Solani.txt"))
    ebv = ebv[,-c(2,3)]
    ebv = ebv[,c(1,2,4,3,5,6)] #changing the order of the estimated breeding value from mixblup
    ebv = merge(id_tbv, ebv, by.x = "x", by.y = "V1", sort = F)[,-1]

    ## coef above the threshold
    cof = matrix(c(1, 0.7272727, 0, 0, 0,
               0, 0, 0.7071068, 0.7885344, 0.2358384) ,nrow = 5, ncol = 2)

    bves_matrix = as.matrix(ebv[,c(2:6)]) %*% cof
    bves = cbind(ebv[,-c(2,4)], bves_matrix)
    bves = bves[, c(1, 5, 2, 6, 3, 4)]
    colnames(bves) = c(" ", "C.Rate", "C.R_slope", "Protein yield", "Protein_slope", "Protein
yield_2")

    population = insert.bve(population, bves = bves)

    print("accuracies of the breeding value estimation:")

    print(analyze.bv(population,cohorts = c(paste0("Bull_", gen-1:3),
                                            paste0("Cow_", gen-1:3))))

    print(analyze.bv(population,cohorts = c(paste0("Cow_", gen-2))))

    population = breeding.diploid(population,
                                  breeding.size = c(n_bull, n_cow),
                                  selection.size = c(n_bull_sel, n_bull + n_cow),
                                  selection.m.cohorts = paste0("Bull_", gen-1:3),
                                  selection.f.cohorts = paste0("Cow_", gen-1:3),
                                  max.offspring = c(Inf, 1),
                                  selection.criteria = 'bve',
                                  multiple.bve.weights.m = c(1, 0, 1.79, 0, 0),
                                  display.progress = FALSE,
                                  name.cohort = c(paste0("Bull_", gen), paste0("Cow_", gen)))

  }



  bv = matrix(0, nrow = 5, ncol = get.ngen(population))
  kin = matrix(0, nrow = 2, ncol =get.ngen(population))

  for(index in 1:get.ngen(population)){
    bv[,index] = rowMeans(get.bv(population, gen =index))
  }

  for(index2 in 1:get.ngen(population)){
    kin[,index2] = kinship.emp.fast(population, gen = index2)
  }
```

```
  save(file=paste0("heat_analysis_", "4_", nr, ".RData"), list = c("bv", "kin"))
}
```