# AlphaPart - R implementation of the method for partitioning genetic trends

Jana Obšteter1, Justin Holl2, John M. Hickey3, and Gregor Gorjanc3

1Department of Animal Science, Agricultural Institute of Slovenia, Hacquetova ulica 17, 1000 Ljubljana, Slovenia

*2*The Pig Improvement Company, Genus plc, 100 Bluegrass Commons Blvd., Ste 2200, Hendersonville, TN 37075, USA

*3*The Roslin Institute and Royal (Dick) School of Veterinary Studies, University of Edinburgh, Easter Bush, Midlothian, EH259RG, United Kingdom

*\**Jana Obšteter, Agricultural Institute of Slovenia, Department of Animal Science, Hacquetova ulica 17, 1000 Ljubljana, Slovenia

E-mail addresses:

JO: jana.obsteter@kis.si

JH: justin.holl@genusplc.com

JMH: [john.hickey@roslin.ed.ac.uk](mailto:john.hickey@roslin.ed.ac.uk)

GG: [gregor.gorjanc@roslin.ed.ac.uk](mailto:gregor.gorjanc@roslin.ed.ac.uk)

# Abstract

1. **Background:** In this paper we present the AlphaPart R package, an open-source software that implements a method for partitioning breeding values and genetic trends to identify sources of genetic gain. Breeding programmes improve populations for a set of traits, which can be measured with a genetic trend calculated from averaged year of birth estimated breeding values of selection candidates. While sources of the overall genetic gain are generally known, their realised contributions are hard to quantify in complex breeding programmes. The aim of this paper is to present the AlphaPart R package and demonstrate it with a simulated stylized multi‑tier breeding example, usually applied in poultry and pig breeding.
2. **Results:** The package includes the main partitioning function AlphaPart, that partitions the breeding values and genetic trends by analyst defined paths, and a set of functions for handling data and results. The package is freely available from CRAN repository at <http://CRAN.R-project.org/package=AlphaPart>. We demonstrate the use of the package by examining the sources of genetic gain in a stylized multi‑tier breeding programme We partitioned the nucleus and multiplier genetic gain by tier‑gender variable. For traits measured and selected in the multiplier, the multiplier selection generated additional source of genetic gain causing higher genetic gain in the multiplier compared to nucleus. The increase depended on the accuracy and intensity of selection in the multiplier and the extent of gene flow from the nucleus. For traits measured only in the nucleus, the multiplier selection did not contribute to the genetic gain. The multiplier therefore had comparable or smaller genetic gain than the nucleus depending on the amount of gene flow from the nucleus.
3. **Conclusions:** AlphaPart implements a method for partitioning breeding values and genetic trends and provides a useful tool for quantifying the sources of genetic gain in breeding programmes. The use of AlphaPart will help breeders to improve genetic gain through a better understanding of the key selection points that are driving gains in each trait.

Keywords: genetic trend, partition, Mendelian sampling term, R package, multi‑tier breeding

# Background

In this paper we present the AlphaPart R package that implements a method for partitioning breeding values and genetic trends,. Wedemonstrate the package with a stylized multi‑tier breeding example usually applied in poultry and pig breeding. Breeding programmes improve populations for a set of traits by selecting and mating genetically superior individuals. Population improvement can be measured with a genetic trend calculated from averaged year of birth estimated breeding values of selection candidates [1,2].

## Partitioning method

While sources of the overall genetic gain are generally known, their realised contributions are hard to quantify in complex breeding programmes. García-Cortés *et al.* [3] proposed a method for such analysis. For completeness of the manuscript we briefly summarize the method below.

The infinitesimal model assumes that the breeding values follow a normal distribution with mean zero and variance **A**σ2a, where **A** is the pedigree relationship matrix and σ2a is the genetic variance for the trait in the founder population. We can partition non-founders’ breeding value *a* into parent average (0.5*asire +* 0.5*adam)* and Mendelian sampling term (*w*) [4], which arises from independent segregation and recombination of parental chromosomes. For founders we have no parental information, hence we assign their breeding value completely to Mendelian sampling term.

We can decompose the variance of the breeding values to **A**σ2a = **TDT’**σ2a, where **T** is a triangular matrix relating individuals to their ancestors (it holds information on gene transmission), and **D**σ2a includes the variance of the breeding values of the founders and variance of the Mendelian sampling terms of the non-founders. Following this, we can express additive genetic values **a** = **Tw**, where **w** follows a normal distribution with mean zero and variance **D**σ2a . Hence, we can further partition an individual’s breeding value into the contribution of founders breeding values and non-founders’ Mendelian sampling terms (including the individual itself) as

[Woolliams et al., 1999].

We can then specify different groups generating the Mendelian sampling term and assign individuals to the groups (called “path” in the Implementation). If we summarise the group‑specific contributions, we can partition the breeding values into **a** = **a1**+ **a2**+ … + **am**, where 1 to m are different groups, and quantify group contributions to the overall genetic trend.

The partitioning method has been used in a number of cases. Gorjanc *et al.* [5] and Gorjanc *et al.* [6] estimated contributions of national breeding programmes to Brows-Swiss and Holstein country-specific and global genetic trends. Špehar *et al.* [7] estimated contributions of national selection and importation in Croatian Simmental cattle. Škorput *et al.* [8] estimated the contribution of national selection and importation in two pig breeds in Croatia, and extended the analysis with the quantification of uncertainty [2]. However, these studies used manual implementations of the partitioning method, for which no open-source software exists.

The aim of this paper is to: i) present the AlphaPart R package; and ii) demonstrate it with a simulated stylized multi‑tier breeding example that quantifies nucleus-multiplier gene flow and the contribution of nucleus and multiplier selection on genetic gain in the two tiers.

# Implementation

1. We first demonstrate the AlphaPart package and its functions on an example dataset. Next, we describe the simulation of a stylized multi‑tier breeding example to demonstrate the use of AlphaPart.

## AlphaPart

AlphaPart is an R package available from CRAN repository at <https://CRAN.R-project.org/package=AlphaPart>. It consists of the main function AlphaPart() for partitioning breeding values and auxiliary functions for manipulating data and summarizing, visualizing, and saving results. We install and load the package with:

> install.packages(pkg = “AlphaPart”)  
> library(package = "AlphaPart")

The package includes an example dataset AlphaPart.ped. We show the first few lines below. The dataset includes a four-generation pedigree (individual IId, father FId, and mother MId) and information about the generation (gen), country, gender, and breeding values for trait 1 (bv1) and trait 2 (bv2).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| IId | FId | MId | gen | country | gender | bv1 | bv2 |
| A |  |  | 1 | domestic | F | 100 | 88 |
| B |  |  | 1 | import | M | 105 | 110 |
| C | B | A | 2 | domestic | F | 104 | 100 |

Here we describe and demonstrate the functions with the AlphaPart.ped dataset. We use the AlphaPart() function to partition breeding values for trait 1 (bv1) in the AlphaPart.ped by the country variable into domestic and import contributions:

> data(AlphaPart.ped)  
> part <- AlphaPart(x = AlphaPart.ped,  
 colPath = “country”,  
 colBV = “bv1”)

The output part is an AlphaPart object, which is a list with an info element and an element for each of the partitioned traits. The info element is a list with information on the path variable (path), number of paths (nP), names of paths (lP), number of traits (nT), names of traits (lT), and putative warnings (warn). The trait element bv1 is a table (shown below). It includes the input information, the breeding value for the trait (bv1), the parent average (bv1\_pa), the Mendelian sampling term (bv1\_w), and the contributions of each path to the breeding value (bv1\_domestic, bv1\_import).

> part$bv1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IId | FId | MId | gen | country | gender | bv1 | bv1\_pa | bv1\_w | bv1\_domestic | bv1\_import |
| A |  |  | 1 | domestic | F | 100 | 0 | 100 | 100 | 0 |
| B |  |  | 1 | import | M | 105 | 0 | 105 | 0 | 105 |
| C | B | A | 2 | domestic | F | 104 | 102.5 | 1.5 | 51.5 | 52.5 |

The partitioning function AlphaPart() requires a data frame holding pedigree with animal/sire/dam or animal/sire/maternal-grandsire, a time-ordering variable such as year of birth, partition variable (path), and breeding values. Following the method described in García-Cortés *et al.* [3], the function recurses the pedigree from the oldest to the youngest individual, for each individual calculate parent average and Mendelian sampling terms for any number of traits and assign terms to paths (i.e. groups generating Mendelian sampling term). We partition multiple traits by specifying a vector of variables, say colBV = c(“bv1”, “bv2”). The multiple trait option can also serve to partition samples from a posterior distribution of breeding values to quantify uncertainty [2, 8]. To speed-up calculations we use C++ and trait-vectorised partitioning. The function can also simultaneously partition and summarize path contributions by a grouping variable, which is a useful computational speed-up for huge pedigrees. Alternatively, we subsequently use summary() function to summarize the partitions. The output object of the AlphaPart() function is either AlphaPart or summaryAlphaPart class.

We use the generic summary.AlphaPart() function to summarize an AlphaPart object by a grouping variable, say generation (gen):

> sumPartByGen <- summary(part, by = “gen”)

The output is an summaryAlphaPart object. It is a list with an info element and (as in AlphaPart object) and a bv1, which we can display with print() function (shown below). The column N holds the number of individuals in each generation. We used the default function mean to summarize the data, so columns Sum, domestic, and import respectively hold the mean breeding value, mean contribution of domestic selection, and mean contribution of import in each generation.

> print(sumPartByGen)

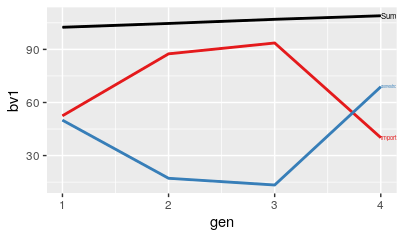
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| gen | N | Sum | domestic | import |
| 1 | 2 | 102.5 | 50 | 52.5 |
| 2 | 3 | 104.6667 | 17.16667 | 87.5 |
| 3 | 2 | 107 | 13.375 | 93.625 |
| 4 | 1 | 109 | 68.875 | 40.125 |

The summary() function summarizes breeding values and their path partitions by levels of grouping variable. By default, we summarize with a mean, but the user can specify any R function via the FUN argument. The summary() function can also summarize only a subset of the object via the subset argument.

We use the generic plot.summaryAlphaPart function to plot summarized partitions:

> plot(sumPartByGen)

The output is an plotSummaryAlphaPart object. It is a list containing one plot for each partitioned trait.



We also provide a number of utility functions that ease partitioning analysis. T pedFixBirthYear() functionimputes missing or fix erroneous years of birth. The pedSetBase() function sets the base population by specifying founders and removing older pedigree records. The AlphaPartSubset() function keeps partitions for specified paths in the AlphaPart or summaryAlphaPart objects. The AlphaPartSum() function sums the partitions of several paths in a summaryAlphaPart object. The AlphaPartSubset() and AlphaPartSum() functions simplify the presentation of partitioning analysis.

|  |  |
| --- | --- |
| Function | Description |
| pedFixBirthYear() | Imputes missing of fixes erroneous years of birth. |
| pedSetBase() | Sets the base population by specifying founders and removing older pedigree records. |
| AlphaPartSubset() | keeps partitions for specified paths in the AlphaPart or summaryAlphaPart objects. |
| AlphaPartSum() | Sums the partitions of several paths in a summaryAlphaPart object. |

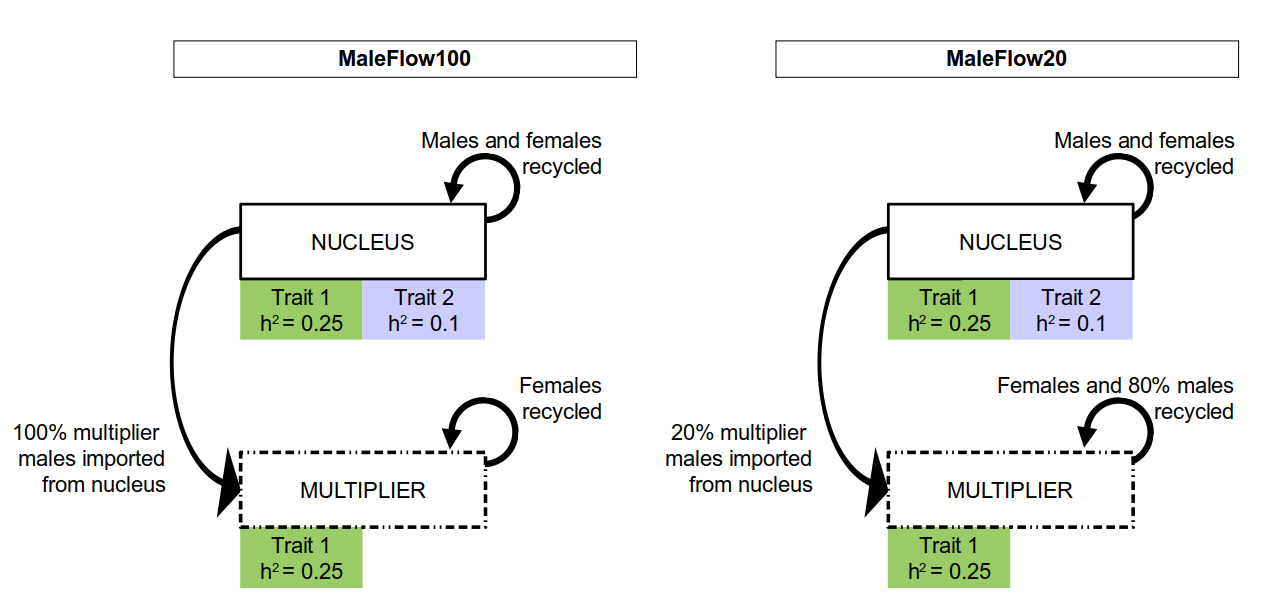
## Stylized multi‑tier breedingexample

We applied the AlphaPart R package on a simulated stylized multi‑tier breeding example, usually applied in poultry and pig breeding. TWe aimed to examine the directional nucleus‑multiplier gene flow and the contribution of nucleus and multiplier selection on genetic gain in each tier. Breeders select in the nucleus and multiply this improvement in the multiplier to supply a large number of commercial animals. The multiplier generally has lower genetic mean than the nucleus due to time-lag. However, animals with very high breeding values are often observed in the multiplier for some traits and we aimed to use AlphaPart to explain the source of this observation. To this end we have first simulated a stylized multi‑tier breeding programme that exposes the drivers of real observations. We have next partitioned the genetic trend of true breeding values by a tier-gender variable to quantify sources of genetic gain in the nucleus and the multiplier.

We used the AlphaSimR package [9] to simulate a multi‑tier breeding programme for a single breed with closed nucleus and directional flow of animals from the nucleus into the multiplier. We simulated 40 generations of selection on two uncorrelated traits. Trait 1 had heritability 0.25 and trait 2 had heritability 0.10. We measured both traits in the nucleus and only trait 1 in the multiplier. We selected on the index of the two traits with equal emphasis. We split the simulation into initial 20 generations of a burn‑in and 20 generations of evaluation.

In the burn-in we simulated only the nucleus and selected animals based on the index of phenotype values for both traits. We selected 25 males and 500 females each generation and randomly crossed them to produce a new generation of 6,000 progeny (12 per cross). At the end of the burn-in we generated 5,000 females to seed the multiplier.

In the evaluation we simulated both the nucleus and the multiplier and selected animals within each tier based on the index of estimated breeding values for both traits. In the nucleus, we selected 25 nucleus males and 500 nucleus females each generation and randomly crossed them to produce a new generation of 6,000 progeny (12 per cross). In the multiplier, we selected 750 multiplier females each generation and randomly crossed them to a set of males to produce a new generation of 9,000 progeny (12 per cross). To quantify the effect of selection in the multiplier on genetic gain we defined the set of males as either 1) the 25 best nucleus males (MaleFlow100 scenario) or 2) the 25 best nucleus males and 100 best multiplier males (MaleFlow20 scenario).



We estimated the breeding values for each trait independently before each nucleus or multiplier selection decision. We ran a pedigree‑based model implemented in blupf90 [10] and used all available data from evaluation generations. The model included the mean as a fixed effect and animal breeding values as a random effect modelled hierarchically with pedigree.

Finally, we partitioned the true breeding values and genetic trends with the AlphaPart as demonstrated above. We used AlphaPart() function to partition standardized true breeding values from the 20 evaluation generations by the tier‑gender variable and summary.AlphaPart() function to summarize the partitions by generations to quantify the contribution of each tier-gender level to genetic trend in the nucleus and the multiplier.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Generation | IId | FId | MId | TierGender | TbvT1\_s | TbvT2\_s | TbvI\_s |
| 20 | 111670 | 107396 | 106535 | GN-M | -0.17 | 0.08 | -0.04 |
| 20 | 111671 | 107396 | 106535 | GN-F | -0.55 | 0.05 | -0.25 |
| 20 | 111672 | 107396 | 106535 | GN-M | -0.31 | -0.17 | -0.24 |
| 20 | 111673 | 107396 | 106535 | GN-F | 0.70 | -0.68 | 0.01 |

# Partition the standardized breeding value for trait 1, trait 2 and index  
Part1g <- AlphaPart(x = as.data.frame(PedEval1), sort = FALSE,  
 colId = "IId", colFid = "FId", colMid = "Mid",  
 colPath = "TierGender",  
 colBV = c("TbvT1\_s", "TbvT2\_s", "TbvI\_s"))

We repeated the simulation 10 times and measured the genetic trend separately in the nucleus and multiplier. We present standardized true breeding values and genetic trends, as well as their partitions with mean set to zero and genetic standard deviation set to one in the generation 20. We chose to present true (instead of estimated) breeding values to assess the true sources of genetic gain. The simulation code for the datasets generated and/or analysed during the current study are available in the GitLab repository, <https://git.ecdf.ed.ac.uk/HighlanderLab_public/jobsteter_alphapart>.

# Results

The results show partitions of true breeding values and genetic trends in the nucleus and multiplier obtained with the AlphaPart for the two simulated stylized multi‑tier breeding scenarios. Partitioning showed that we can explain the situation with very high breeding values in the multiplier by the extent of nucleus-multiplier gene flow as well as accuracy and intensity of multiplier selection. For each scenario we first describe the distribution of true breeding values in the nucleus and multiplier in generation 40 of one replicate. Next we explain the sources of the observations obtained by partitioning the nucleus and multiplier genetic trend and averaging the results across ten replicates.

## MaleFlow100 scenario

### Distribution of breeding values

In Male100 scenario the multiplier had a higher genetic merit than the nucleus for trait 1 and trait 2. We show this in Figure 1 that presents the distribution of true breeding values and their partitions in nucleus and multiplier by trait in generation 40 of one replicate of MaleFlow100 scenario. The multiplier hence produced animals with a higher breeding value for than the nucleus for both traits, which reflected in a higher index value as well.

## Partitioning the true breeding values and genetic trend of MaleFlow100 scenario

The partitioning showed that the higher genetic gain in the multiplier compared to the nucleus for trait 1 was due to additional contribution of selection of multiplier females. We show this in Figure 2 that presents the genetic trends in the nucleus and multiplier by trait and their partitions summarised over 10 replicates. As expected, the nucleus genetic gain stemmed completely from selection of nucleus males and nucleus females, since we did not import the multiplier animals into the nucleus. The selection of nucleus males contributed more to the genetic gain than the selection of nucleus females. The mean genetic gain in generation 40 in the nucleus for trait 1 was 9.75 and 8.34 for trait 2, with male selection contributing 5.65 for trait 1 and 4.92 for trait 2, and female selection contributing 4.10 for trait 1 and 3.42 for trait 2.

In the multiplier, the average genetic gain for trait 1 was higher than in the nucleus. This was partly due to larger contribution of nucleus males to the multiplier than to nucleus genetic gain, since they contributed directly by fathering the multiplier animals and indirectly by fathering the multiplier mothers (via gene flow). Andpartly, this was due to non-zero contribution of multiplier female selection. The mean genetic gain in generation 40 in the multiplier for trait 1 was 10.00 with nucleus males contributing 5.75, nucleus females 4.09, and multiplier females 0.14. The mean genetic gain and path partitions in generation 40 for trait 2 in the multiplier were comparable to the nucleus.

## MaleFlow20 scenario

### Distribution of breeding values

In Male20 scenario the multiplier had a higher genetic merit than the nucleus for trait 1, but lower for trait 2. We show this in Figure 3 that presents the distribution of true breeding values and their partitions in nucleus and multiplier by trait in generation 40 of one replicate of MaleFlow20 scenario. We again observed animals with higher breeding values for trait 1 in the multiplier than in the nucleus, with an even larger difference than in MaleFlow100 scenarios. We did not observe the same phenomena for trait 2.

### Partitioning the true breeding values and genetic trend of MaleFlow20 scenario

The partitioning revealed, that selection of multiplier males and females further increased the genetic gain for trait 1 in the multiplier compared to the nucleus, but decreased the genetic gain for trait 2. We show this in Figure 4 that presents the genetic trends in the nucleus and multiplier by trait and their partitions summarised over 10 replicates. As in MaleFlow100 scenario, the nucleus genetic gain stemmed from selection of nucleus males and females.The mean genetic gain in generation 40 for trait 1 was 10.09 and 8.39 for trait 2, with nucleus males contributing 5.69 for trait 1 and 5.17 for trait 2, and nucleus females contributing 4.40 for trait 1 and 3.22 for trait 2.

In the multiplier the genetic gain was again higher than in the nucleus, but only for trait 1. This higher genetic gain was a result of non‑zero contribution of multiplier female and male selection and reduced contribution of nucleus females. In MaleFlow20 we reduced the use of nucleus males in the multiplier, which reduced the contribution of nucleus females via reduced nucleus-multiplier gene flow. For trait 2, the genetic gain in the multiplier was lower than in the nucleus due to a small average negative contribution of multiplier females and multiplier males and reduced contribution of nucleus females and nucleus males via reduced gene flow. The mean genetic gain in generation 40 in the multiplier was 10.36 for trait 1 and 8.14 for trait 2, with nucleus males contributing 5.70 for trait 1 and 5.09 for trait 2, nucleus females contributing 4.21 for trait 1 and 3.13 for trait 2, multiplier males contributing 0.15 for trait 1 and -0.03 for trait 2, and multiplier females contributing 0.30 for trait 1 and -0.05 for trait 2.

# Discussion

In this paper we present AlphaPart, freely available R package that implements the method for partitioning breeding values and genetic trends. We demonstrate the package on a simulated stylized multi‑tier breeding example with a higher genetic trend for some traits in the multiplier compared to the nucleus. Following this, we organized the discussion into two parts: i) advantages and disadvantages of the AlphaPart R package; ii) partitioning results of the breeding example.

## AlphaPart

AlphaPart is the first free implementation of the method for partitioning breeding values and genetic trends. The method and the package are valuable for deciphering and quantifying the sources of genetic gain in breeding programmes. The package is easy to use, since it streamlines the partitioning analysis into a few lines of R code. AlphaPart presents a holistic tool to perform a partitioning analysis, from preparing the input data - such as manipulating the pedigree data - to handling of results and plotting. The partitioning step is fast, even for large pedigrees, since the main partitioning function is recursive and implemented in C++.

AlphaPart is aimed at the researchers who are interested in quantifying the sources of genetic gain in their breeding programmes either to understand the dynamics of genetic gain, improve efficiency, asses the performance of different breeding actions, optimize investments etc. Users should take into account, that the accuracy of partitioning depends on the accuracy of the estimated breeding values and their Mendelian sampling terms, which are driven by the biology of the trait, breeding programme structure, and choice of the prediction model.

Our future work on AlphaPart will include extending the partitioning method in three areas. The first extension will utilise genomic information to inform which genome regions drive genetic change and what are sources of specific haplotypes or alleles. The second extension will use the partitioning method to analyse changes in genetic variance in addition to the genetic mean. The third extension will simplify handling of uncertainty of path contributions when working with samples from posterior distributions [2, 8].

## Stylized multi‑tier breeding example

The multi‑tier breeding example showed the investigative power of the partitioning method and the free AlphaPart implementation. Here we discuss the sources of genetic gain in the two tiers of a breeding programme.

By partitioning the genetic trend in a simulated multi‑tier breeding programme, we disentangled the observation of some multiplier animals having higher breeding values for some traits compared to the nucleus animals. While larger number of recombinations in the multiplier can potentially reveal more variation and occasional outlying animals, we expect lower breeding values in the multiplier due to time-lag between the nucleus and multiplier. The partitioning revealed that the gene flow from the nucleus into the multiplier was the main source of genetic gain in the multiplier, with the nucleus males contributing the most. This was expected due to nucleus-multiplier gene flow and higher intensity of selection in males.

However, the results also showed that selection in the multiplier can contribute genetic gain in addition to the gene flow from the nucleus. The multiplier outperformed the nucleus for trait 1, because with the 10,500 recorded multiplier animals there was substantial amount of information for accurate multiplier selection that generated additional genetic gain. The partitioning of genetic trend for trait 1 showed that when we used only the nucleus males in the multiplier (MaleFlow100), the multiplier generated additional gain from two sources. First, compared to the nucleus, the contribution of the nucleus males increased because they contributed through the gene flow and through the selection of multiplier females. Second, the selection of multiplier females contributed as well. When we used both the nucleus males and the multiplier males in the multiplier (MaleFlow20), the multiplier generated further gain through a combination of the sources. First was the contribution of the selection of multiplier females and males. In contrast, the contribution of nucleus selection decreased due to the reduced gene flow. This decrease was due to a smaller number of progeny per nucleus male compared to the MaleFlow100 scenario.

On the contrary, trait 2 was not measured in the multiplier and had comparable or smaller genetic trend in the multiplier than in the nucleus. For trait 2 the multiplier animals were selected only on estimated parent average, which resulted in low accuracy selection. In the MaleFlow100 scenario this low accuracy selection resulted in a null contribution of multiplier females to the genetic trend for trait 2 and comparable genetic trends between the nucleus and the multiplier. In the MaleFlow20 scenario with a reduced nucleus-multiplier gene flow this low accuracy selection resulted in the reduced genetic gain for trait 2.

# Conclusion

AlphaPart R package is a freely available software for partitioning breeding values and genetic trends. Use of AlphaPart will help breeders to better understand sources of genetic gain and improve their breeding programmes.

**Declarations**

## Ethics approval and consent to participate

Not applicable

## Consent for publication

Not applicable

## Availability of data and materials

Project name: AlphaPartProject home page: https://cran.r-project.org/package=AlphaPartOperating system(s): Windows, MacOS, LinuxProgramming language: R & C++License:**GPL-2 | GPL-3**Any restrictions to use by non-academics:**-**

## **Competing interests**

Not applicable

## Funding

The authors acknowledge support from the BBSRC to The Roslin Institute (BBS/E/D/30002275) and The University of Edinburgh’s Data-Driven Innovation Chancellor’s fellowship.

## Authors’ contributions

Not applicable

## Acknowledgements

Not applicable

## **Author’s information** (optional)

# References

1. 1. Blair HT, Pollak EJ. Estimation of genetic trend in a selected population with and without the use of a control population. J Anim Sci. 1984;58:878–86.
2. 2. Sorensen D, Wang C, Jensen J, Gianola D. Bayesian analysis of genetic change due to selection using Gibbs sampling. Genet Sel Evol. 1994;26:333.
3. 3. García-Cortés LA, Martínez-Ávila JC, Toro MA. Partition of the genetic trend to validate multiple selection decisions. Animal. 2008;2:821–4.
4. 4. Thompson R. Sire Evaluation. Biometrics. 1979;35:339–53.
5. 5. Gorjanc G, Potočnik K, García-Cortés LA, Jakobsen J, Dürr J. Partitioning of international genetic trends by origin in Brown Swiss bulls. In: Proceedings of the 2011 Interbull meeting: ; August 27-29 2011; Stavanger, Norway. <https://journal.interbull.org/index.php/ib/article/view/1196>.
6. 6. Gorjanc G, Hely F, Amer P. Partitioning international genetic trends by origin in Holstein bulls. Presented at: ICAR 38th Annual Meeting: 28 May - 1 June 2012; Cork, Ireland.
7. 7. Špehar M, Ivkić Z, Bulić V, Barać Z, Gorjanc G. Partitioning of Genetic Trends by Origin in Croatian Simmental Cattle. Agriculturae Conspectus Scientificus. 2011;76:301-304.
8. 8. Škorput D, Gorjanc G, Kasap A, Luković Z. Partition of genetic trends by origin in Landrace and Large-White pigs. animal. 2015;9:1605–9.
9. 9. Gaynor RC, Gorjanc G, Wilson DL, Money D, Hickey JM. AlphaSimR: An R Package for Breeding Program Simulations. Manuscript in preparation. 2019.
10. 10. Misztal I, Tsuruta S, Strabel T, Auvray B, Druet T, Lee DH. BLUPF90 and related programs (BGF90). In: Proc 7th World Congress on Genetics Applied to Livestock Production: August 19-23 2002; Montpellier, France. <http://www.wcgalp.org/system/files/proceedings/2002/blupf90-and-related-programs-bgf90.pdf>.

# Figures

**Figure 1** **Distribution of true breeding values and their partitions by trait, generation, and tier in MaleFlow100 scenario.**

We show scaled densities of partitions in generations 23 and 40 of one simulation replicate. MalerFlow100 uses only nucleus males in the multiplier. Trait 1 is measured in the nucleus and the multiplier, while trait 2 is measured only in the nucleus. Black vertical lines represent the nucleus mean breeding value for a trait in a generation.

## Figure 2: Partitioning of genetic trend by tier-gender in MaleFlow100 scenario.

## The scenario uses nucleus males in the multiplier. Trait 1 is measured in the nucleus and the multiplier, while trait 2 is measured only in the nucleus.

**Figure 3** **Distribution of true breeding values and their partitions by trait, generation, and tier in MaleFlow20 scenario.**

We show scaled densities of partitions in generations 23 and 40 of one simulation replicate. MalerFlow20 uses nucleus and multiplier males in the multiplier. Trait 1 is measured in the nucleus and the multiplier, while trait 2 is measured only in the nucleus. Black vertical lines represent the nucleus mean breeding value for a trait in a generation.

**Figure 4** **Partitioning of the genetic trend by tier-gender in MaleFlow20 scenario.**

The scenarios uses nucleus and multiplier males in the multiplier. Trait 1 is measured in the nucleus and the multiplier, while trait 2 is measured only in the nucleus.