# 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 the relative contributions of selection pathways to genetic gain. Breeding programmes improve populations for a set of traits, which can be measured with a genetic trend calculated from estimated breeding values averaged by year of birth. 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 programme mimicking a pig or poultry breeding programme.
2. **Results:** The package includes the main partitioning function AlphaPart, that partitions the breeding values and genetic trends by pre-defined selection paths, and a set of functions for handling data and results. The package is freely available from the CRAN repository at <http://CRAN.R-project.org/package=AlphaPart>. We demonstrate the use of the package by partitioning the nucleus and multiplier genetic gain of the stylized breeding programme by tier‑gender paths. For traits measured and selected in the multiplier, the multiplier selection generated additional genetic gain. By using AlphaPart we show that the increase depends on an intricate relationship between accuracy and intensity of selection in the multiplier and the extent of gene flow from the nucleus. AlphaPart proved as a valuable tool in understanding the sources of genetic gain in the nucleus and especially the multiplier, as well as the relationship between the sources and parameters affecting them.
3. **Conclusions:** AlphaPart implements the 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. We demonstrate the package with a stylized multi‑tier breeding example usually applied in pig and poultry 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 by averaging estimated breeding values by year of birth [1,2].

While sources of the overall genetic gain are generally known, their realised contributions are hard to quantify in complex breeding programmes due to many interacting selection parameters. García-Cortés *et al.* [3] proposed a method for such analysis. In summary, the method uses pedigree information to first partition the breeding values into a parent average and a Mendelian sampling term: *ai = ½as +* *½ad + wi* [4], where *ai, as,* and *ad* are the breeding value of the individual, its sire, and its dam, respectively, and *wi* is the individual’s Mendelian sampling term. The parent average captures the progeny inheriting half of its genetic material from each parent and the Mendelian sampling term captures the variability arising from recombination and segregation of parental chromosomes. The partitioning method then allocates Mendelian sampling terms to the selection “path” generating it. For example, assume a small trio pedigree with two parents and a female progeny. Specifying gender with two levels (males, females) as the selection path variable generating Mendelian sampling term, we can write *ai = ½as +* *½ad + wi.* Since individual *i* is a female, we assign her Mendelian sampling term *wi* to the contribution of females as we do with half of the dam’s breeding value (½ *ai*). We assign half of the sire’s breeding value to the contribution of males. The term reduces to *(½ad + wi) + ½as  = ai,f + aim*, where the term *ai,f* denotes the contribution of females and the second term *ai,m* denotes the contribution of males. Alternatively, assume that the sire is imported, then an interesting path specification can be made separating contributions from domestic versus imported sources, which can be partitioned similarly (yields the same result as the partition by gender above, since only the one male is imported??? - WOULD THIS BE BETTER?) to the gender example.. In general, we can write a vector of breeding values as a linear combination of Mendelian sampling terms of individuals and their ancestors’ contributions: **a** = **Tw**, where **T** is a triangular matrix of expected gene flow between ancestors and individuals [4,5] and w is a vector of ancestors’ Mendelian sampling terms.. The method of García-Cortés *et al.* [3] uses a path variable to partition the gene flow matrix **T** = **T**1 + **T**2 + … + **T**p and with this partitions breeding values by paths **a** = (**T**1 + **T**2 + … + **T**p)**w** = **a**1 + **a**2 + … + **a**p, where Tp combines (SELECTS?) the contributions of ancestors of a specific level of the path variable (e.g. females and males). Aggregating these partitions by other variables (such as year of birth, insemination centres, countries, gender etc.) is a powerful way to analyse sources of genetic gain.

The partitioning method has been used in a number of cases. Gorjanc *et al.* [6] and Gorjanc *et al.* [7] estimated contributions of breeding programmes in different countries to country specific and global genetic trends in the Brows-Swiss and Holsteinpopulations . Špehar *et al.* [8] estimated contributions of domestic and foreign selection paths to genetic gain of Croatian Simmental cattle whereas Škorput *et al.* [9] estimated such contributions to genetic gain in two pig breeds in Croatia. The latter study also extended the analysis by accounting for the uncertainty of the estimated breeding values and partitions [2]. However, these studies used bespoke implementations of the partitioning method, for which no open-source software exists.

The aim of this paper is to present the AlphaPart R package that implements a method for partitioning breeding values and genetic trends. We demonstrate the usage of the package with an example session in which we use sample data to partition genetic gain of a small population importing genetic material. We next demonstrate the power / value / usefulness of the package by partitioning the genetic trends of a more complex and real-life a stylized multi‑tier breeding example usually applied in pig and poultry breeding .

# Implementation

1. AlphaPart is an R package available from the CRAN repository at <https://CRAN.R-project.org/package=AlphaPart> and thus easily installed via install.packages(“AlphaPart”) and loaded with library(AlphaPart) command inside R. We developed the package with the aim to create a user-friendly implementation of the method for partitioning breeding values and genetic trends.
2. The package is user-friendly, since all the functions have self-explanatory names. The only input required from the used is the initial data frame. All the subsequent functions in the analysis pipeline accept the output of a preceding function and require no additional input.
3. We first demonstrate the standard AlphaPart analysis pipeline and the usage of function on a sample dataset in an example session. Next, we demonstrate the usefulness of AlphaPart as a tool for resolving real-life observations by analysing the dynamics of genetic gain in a more realistic and complex multi-tier breeding programme.

## Example session

Here, we demonstrate the functionality of the software on a simulated example. Consider a case of a small population (population 1) that imports genetic material from two larger populations (population 2 and population 3) with a higher genetic mean. Populations 2 and 3 achieve a higher genetic mean due to higher accuracy of selection (h2 = .9 vs. h2=0.7 in population 1). Population 3 additionally benefits from higher intensity of male selection (2.89 vs 2.67 in populations 1 and 2). Assume presence of genotype-environment interactions and the genetic correlation of trait in population 1 with trait in population 2 of 0.9 and with trait in population 3 of 0.8. Assume 20 generations of selection of males on phenotypic values. In generation 1 – 10 populations perform selection within the population and do not exchange genetic material. In generations 10 - 20 population 1 imports 20% of their male genetic material from population 2 and population 3, 10% from each. To optimize the breeding strategy the population 1 would like to know how much of the observed genetic gain comes from selection in each of the populations. Hence we would like to partition the breeding values by the population variable. To analyse such a situation we simulated a dataset and we give the details of the simulation in Additional file 1.

**Preparing the data**

The first step is preparing the input data. The main input for the analysis is a data frame holding pedigree with individual/sire/dam or individual/sire/maternal-grandsire identification, a time-ordering variable such as year of birth, partition variable (path), and breeding values for one or multiple traits. The package provides functions to pre-process the input data, such as for correcting missing or incorrect years of birth and setting the base population of the pedigree.

We prepare the input data frame PedEval containing the identification for each individual and its parents (individual IId, father FId, and mother Mid), generation of birth (Generation), population (Population), and breeding values (Tbv) for the correlated (OBSERVED/MEASURED?) trait (Figure 1).

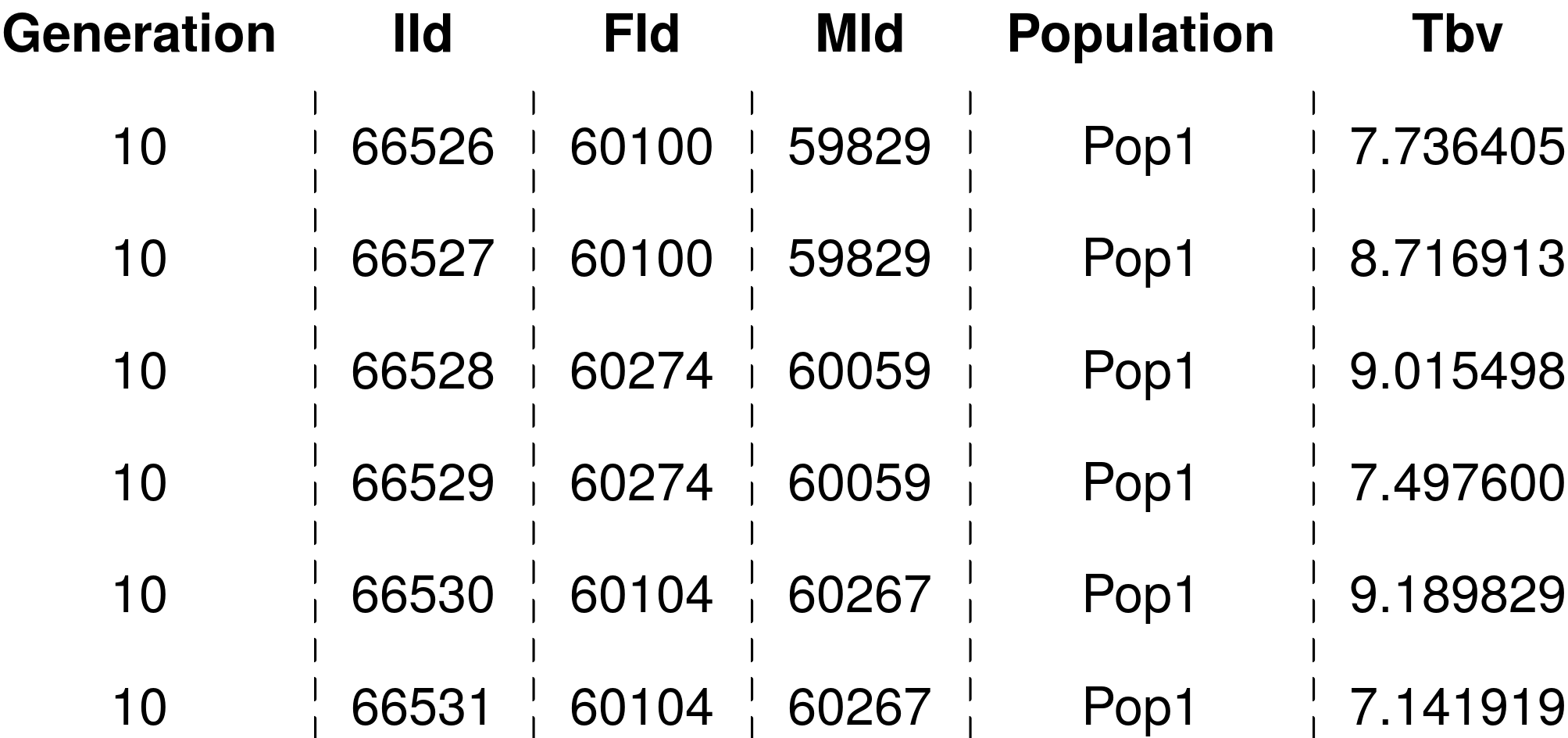


Figure 1: Example input data frame for partitioning analysis. The data frame holds individual’s generation, identification (IId), father’s identification (Fid), mother’s identification (MId), population and true breeding value (Tbv).

If the input data contains missing or erroneous years of birth, this affects the accuracy of partitioning. The package allow us to impute missing or correct erroneous years of birth with pedFixBirthYear() function. Since we are using simulated data in this example, we have no missing or erroneous generations of birth. If we had, we would run the following line:

PedEval ← pedFixBirthYear(x = PedEval,

interval=1,

colId = "IId", colFid = "FId", colMid = "MId")

The pedFixBirthYear() requires the input data frame as described above (Figure 1) holding at least individual, father and mother identification, and the year of birth. It also requires the generation interval via interval parameter. The function computes the missing birth years either based on i) offspring by adding the generation interval to the birth year of the oldest offspring to obtain the parent birth year (parameter down set to FALSE); or ii) parents by subtracting the generation interval from the birth year of the youngest parent to obtain offspring birth year (parameter down set to TRUE). The output of the function is a data frame with corrected years of birth.

Our data frame contains information on 20 years of selection, 10 of those selection within each of the populations and 10 years of import into population 1. If we want to consider only the generations in which we perform import of genetic material, i.e. generations 10 to 20, we would use function pedSetBase() to rebase the pedigree:

PedEval ← pedSetBase(x = PedEval,

keep = PedEval$Generation > 10,

colId = "IId", colFid = "FId", colMid = "MId")

The pedSetBase() function requires the input data frame as described above (Figure 1) holding at least individual, father and mother identification, and the year of birth. It also requires instructions for how to rebase the pedigree via the keep parameter. The function removes all the individuals that do not meet the condition as well as their presence as parents. The output of the function is a data frame with adjusted set of individuals.

**Partitioning analysis**

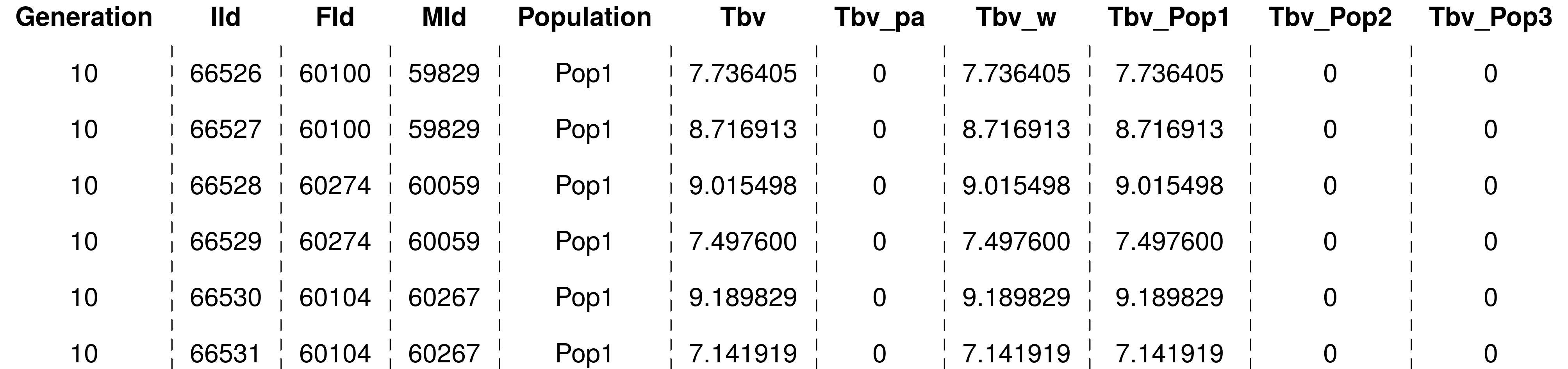
The second step is partitioning the breeding values. We perform this with the main function of the package, AlphaPart(), that partitions the individual breeding values into contributions of pre-defined paths. We apply the AlphaPart() function to partition breeding values for the observed trait (Tbv) by the population (Population) variable into contributions of selection in population 1 (domestic source), population 2 (imported source) and population 3 (imported source):

> part <- AlphaPart(x = PedEval,  
 colPath = “Population”,  
 colBV = “Tbv1”)

The function AlphaPart() requires the input data frame as described above and shown on Figure 1. Following the method described in García-Cortés *et al.* [3], the function recurses the pedigree from the oldest to the youngest individual and calculates for each individual its parent average and Mendelian sampling terms for the trait. It then assigns half of the parent average term to each of the paths that generate the parents and Mendelian sampling term to the path that generates the individual. For the founders, the function assigns their entire breeding value to the path that generates the individual founder. The function can also partition multiple traits simultaneously by specifying a vector of variables, say colBV = c(“Tbv1”, “Tbv2”). This outputs separate partitions for each of the traits in a single data frame. The multiple trait option can also serve to partition samples from a posterior distribution of breeding values to quantify uncertainty when estimating breeding values within a Bayesian framework [2, 9]. 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 (i.e. generation) provided via colBy parameter, which is a useful computational speed-up for huge pedigrees. Alternatively, we subsequently use summary.AlphaPart() function to summarize the partitions.

The output of the AlphaPart() function is an object of either a data frame with partitioned breeding values (AlphaPart class) or partitioned and summarized breeding values (summaryAlphaPart class(. The output is a list with an info element and one element with partitioned breeding values for each of the 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 is a data frame, named Tbv for the example above (Figure 2), holding for each individual the information from the original dataset as well as parent average (Tbv1\_pa), Mendelian sampling term (Tbv\_w), and breeding value partitions (Tbv\_Pop1, Tbv\_Pop2 and Tbv\_Pop3).

3

Figure 2: Example output data frame of partitioning analysis with AlphaPart() function. The data frame holds individual’s generation, ID (IId), father’s IDs (Fid), mother’s ID (MId), population, true breeding value (Tbv), parent average (Tb\_pa) and Mendelian sampling term (Tbv\_w) for the trait, and contributions of the three populations (Tbv\_Pop1, Tbv\_Pop2, Tbv\_Pop3).

**Analysing the results**

The third step is analysing the results. The output of the partitioning analysis is a partitioned breeding value for each individual, which is usually not what we are interested in. We usually want to compare the total or mean contributions of paths or how the contributions change through time. To facilitate such analysis, the package includes functions to summarize and/or combine the contributions of paths (by a grouping variable), subset the partitioning results and visualise the results.

An interesting measure is the trend of mean breeding values (genetic trend) and path partitions (partial genetic trends) through time. To summarize the breeding values and the partitions by generation we use the summary.AlphaPart().: We are interested in the genetic gain of population 1, hence we use the subset option to filter out population 1 individuals.

> sumPartByGen <- summary(part, by = “Generation”,  
 subset = part$Tbv$Population == “Pop1”)

The input for the summary.AlphaPart() is the output of the partitioning analysis (AlphaPart class). The function also requires a variable by which to summarize the breeding values and their partitions via by parameter. In the case above, we are summarizing by generation, but depending on the analysis this could be any variable (i.e. gender, insemination centre, lineage, population …) or a concatenated variable (e.g. generation-gender). By default, the function summarizes with the mean of the breeding values of the analysed trait, but the user can specify any R function via the FUN argument. The function can also summarize only a subset of the object by specifying which records to keep. The user passes the condition as a vector of logical values, i.e. true of false, via the subset argument.

The output of the summary.AlphaPart() function is a data frame holding summarized breeding values and their partitions for the trait(summaryAlphaPart class). It is a list with an info element and one element for each partitioned trait.. The summary element, named Tbv for the example above, contains the grouping variable levels (Generation), number of individuals per level (N), and output of the summary function applied to the breeding values (Sum) and its partitions (, Pop1, Pop2, and Pop3).

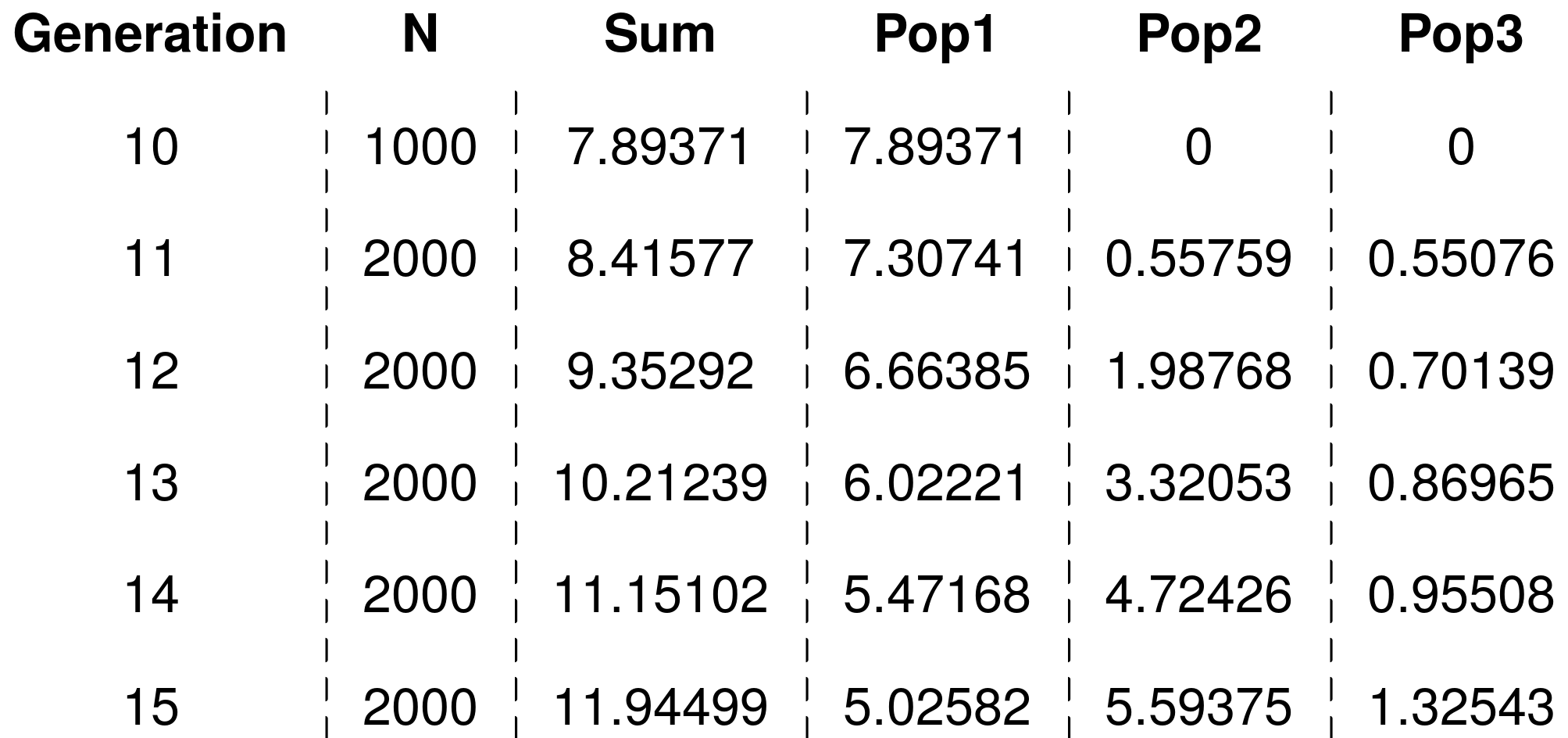


Figure 3: Example output data frame of summarized partitions obtained with summary.AlphaPart() function. The data frame holds information on generation, number of data points (N), the total genetic gain (Sum) and mean contributions of the three populations (Pop1, Pop2, Pop3).

We can use package functions to further manipulate the partitioned or summarized results. If we are only interested in the contributions of populations 1 and 2, we can filter out the summary results with AlphaPartSubset() function:

AlphaPartSubset(sumPartByGen, paths = c("Pop1", "Pop2"))

AlphaPartSubset() takes the output of either partitioning (AlphaPart class) or summarizing (summaryAlphaPart class) analysis, and a character vector of paths to keep via paths parameter. The output of the function is a truncated dataset of the input class.

We could also be interested only in the contribution of domestic vs. imported sources and not of specific populations. To this end we can combine the contributions of imported sources, i.e. populations 2 and 3, with AlphaPartSum() function and compare the contributions of population 1 (named Domestic) vs combined contributions of imported sources (named Import):

AlphaPartSum(sumPartByGen,

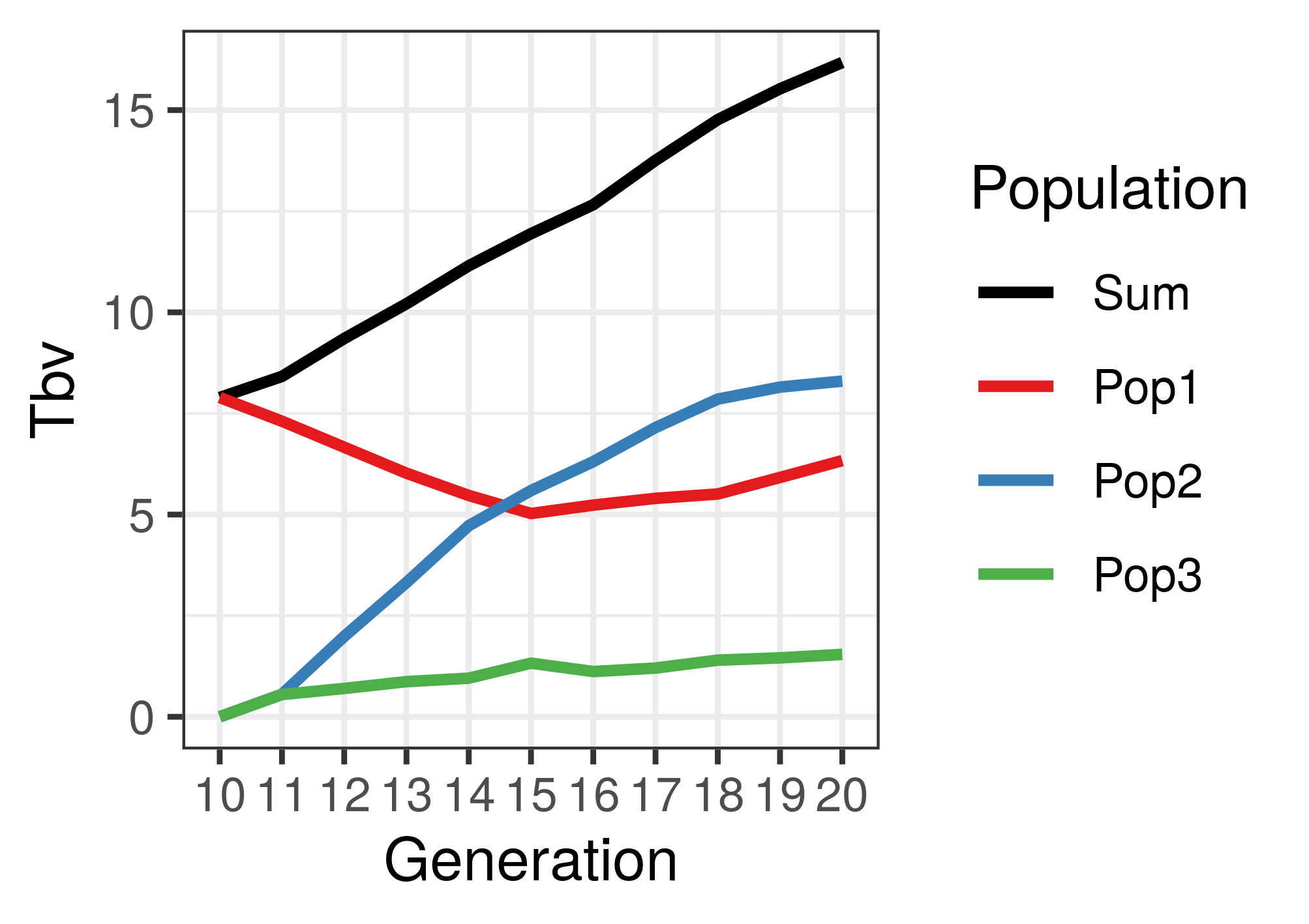
map=list(c("Domestic", "Pop1"), c("Import", "Pop2", "Pop3")))

The input for AlphaPartSum() can be the output of either partitioning (class AlphaPart) or summarizing (class summaryAlphaPart) analysis. The function also takes a list of paths to sum via the map parameter. Each element of this list contains the name of the newly created combined path followed by names of the paths to combine.

Lastly, we want to plot the summarized partitions for which we use the plot.summaryAlphaPart function:

> plot(sumPartByGen)

The input for the plotting function must be a summarized dataset (class summaryAlphaPart) that could have been additionally subset or in which some of paths have been combined. The output is a list ( plotSummaryAlphaPart class) containing one plot for each partitioned trait. We show the output plot in Figure 4.



## Simulated multi‑tier breeding example

We applied the AlphaPart R package on a simulated example of a multi‑tier pig breeding programme. We aimed to examine the directional gene flow between nucleus and multiplier 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 breeding animals for commercial purposes. The multiplier generally has lower genetic mean than the nucleus due to a time-lag in the gene flow. However, animals with very high breeding values for some traits could be observed in the multiplier tier and we aimed to use AlphaPart to explain the source of this observation. We used stochastic simulation to simulate a multi‑tier breeding programme. We have next used AlphaPart to partition 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 [10] to simulate the stylized multi‑tier breeding programme for a single breed with closed nucleus and directional flow of animals from the nucleus into the multiplier (Figure 2). 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 an index with equal weights on the estimated breeding values for the two traits. We split the simulation into an initial burn-in period of twenty generations to achieve a population equilibrium and a subsequent twenty-year period of genetic evaluation and selection.

In the burn-in period 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 (Figure 2). 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 mating). 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 – 100% of males are from the nucleus) or 2) the 25 best nucleus males and 100 best multiplier males (MaleFlow20 scenario – 20% of males are from the nucleus).

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 [11] 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.

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 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 by partitioning the nucleus and multiplier genetic trend and averaging the results across ten replicates. Additional files 1 and 2 show distributions of partitioned true breeding values for one replicate.

## MaleFlow100 scenario

### Distribution of breeding values

In MaleFlow100 scenario the multiplier had a higher genetic merit on average than the nucleus for trait 1 and trait 2. We show this in Figure 3 with the distribution of true breeding values in the nucleus and the multiplier by trait in generation 40 of one replicate. The multiplier had a higher genetic merit on average and hence produced animals with a higher breeding value than the nucleus for both traits, which reflected in a higher index value as well.

## *Partitioning the true breeding values and genetic trend*

The partitioning showed that the higher genetic gain in the multiplier compared to the nucleus for trait 1 was due to additional contribution from selection of multiplier females. We show this in Figure 4 with 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. The selection of nucleus males contributed more to the genetic gain than the selection of nucleus females. The mean genetic gain at 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 increase was driven by two sources. First, nucleus males made a larger contribution to the multiplier than to nucleus genetic gain, since they contributed directly by fathering the multiplier animals and indirectly through subsequent selection of their genes in future generations. Second, multiplier female selection made a non-zero contribution. The mean genetic gain at 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 its path partitioning at generation 40 for trait 2 were similar in the in the multiplier and the nucleus. Additional files 1 shows distributions of partitioned true breeding values for one replicate.

## MaleFlow20 scenario

### Distribution of breeding values

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

### Partitioning the true breeding values and genetic trend

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 5 with 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 at 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 a 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 at 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. Additional file 2 shows distributions of partitioned true breeding values for one replicate.

# Discussion

In this paper we present AlphaPart, a 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 a 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 researchers who are interested in quantifying the sources of genetic gain in their breeding programmes in order to understand the dynamics of genetic gain, improve selection efficiency in certain partitions, asses the performance of different breeding actions, or to optimize investment. The accuracy of partitioning depends on the accuracy of the estimated breeding values and their Mendelian sampling terms, which are driven by the genetic parameters of the trait, the information available in the breeding programme structure used, and choice of the prediction model.

Future development of AlphaPart will include an extesniosn of the partitioning method in three areas. The first extension will utilise genomic information to inform which genome regions and which specific haplotypes or alleles drive genetic change. The second extension will use the partitioning method to analyse contributions to 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, 9].

## Stylized multi‑tier breeding example

The multi‑tier breeding example illustrated 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 numbers 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. We emphasise that this result is also due to a limited time-lag between the nucleus and multiplier as we used the nucleus males in nucleus and multiplier concurrently assuming artificial insemination. 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. In both scenarios we observed a trend of decreasing contribution of multiplier selection over generations, though the average multiplier contribution was always above zero. Since we partitioned breeding values with generation 20 as a base generation, the parent average and Mendelian sampling terms for multiplier animals in the generation 20 were assigned to the multiplier path. Over the generations the nucleus and multiplier contributions converged since the used pedigree in next generations accounted for the origin of the nucleus males. This shows the importance of proper base population specification (including unknown parent groups) for meaningful partitioning. This long-term dynamic of contributions is related to the dynamic of “long-term genetic contributions” in the context of genetic gain and inbreeding [12, 13], but note that the “long-term genetic contributions” are trait agnostic (depend only on the pedigree). On a related note, with the implemented method in AlphaPart we can evaluate (long-term) genetic contributions by setting breeding values to the value one for all animals and partitioning the breeding values by analyst defined paths [6].

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. Henderson C. A Simple Method for Computing the Inverse of a Numerator Relationship Matrix Used in Prediction of Breeding Values. Biometrics. 1976;32:69–83.
6. 6. 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>.
7. 7. 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.
8. 8. 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.
9. 9. 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.
10. 10. Gaynor RC, Gorjanc G, Hickey JM. AlphaSimR: An R Package for Breeding Program Simulations. bioRxiv. 2020;doi: https://doi.org/10.1101/2020.08.10.245167.
11. 11. Misztal I, Tsuruta S, Strabel T, Auvray B, Druet T, Lee DH. BLUPF90 and related programs (BGF90). In: Proceedings of the 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>.
12. 12. Woolliams JA, Bijma P, Villanueva B. Expected Genetic Contributions and Their Impact on Gene Flow and Genetic Gain. Genetics. 1999;153;2:1009-1020.
13. 13. Howard DM, Pong-Wong R, Knap PW, Kremer V, Woolliams J. Selective advantage of implementing optimal contributions selection and timescales for the convergence of long-term genetic contributions. Genet Sel Evol. 2018;50:24.

# Figures

Figure 1 The output of the plot.summaryAlphaPart() function.

The plot shows the overall genetic trend and its partition into the contributions of domestic selection and import by generation.

Figure 2 Design of the simulated stylized multi-tier breeding programmes.

We simulated two scenarios with a closed nucleus and a directional flow of animals from the nucleus into the multiplier. The scenarios differ in the percentage of multiplier males imported from the nucleus.

Figure 3 Distribution of true breeding values and their partitions by trait and tier in MaleFlow100 scenario.

We show scaled densities of partitions in generation 40 of one simulation replicate. MaleFlow100 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.

* 1. Figure 4 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 5 Partitioning of the genetic trend by tier-gender in MaleFlow20 scenario.

The scenario 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.

# Additional files

**Additional file 1 Figure S1** (file: Additional\_File\_1.docx)

Format: Figure in a .docx document

Title: Distribution of true breeding values and their partitions by trait, year, and tier in MaleFlow100 scenario.

Description: We show scaled densities of partitions in years 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 year.

**Additional file 2 Figure S2** (file: Additional\_File\_2.docx)

Format: Figure in a .docx document

Title: Distribution of true breeding values and their partitions by trait, year, and tier in MaleFlow20 scenario.

Description: We show scaled densities of partitions in years 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 year.