

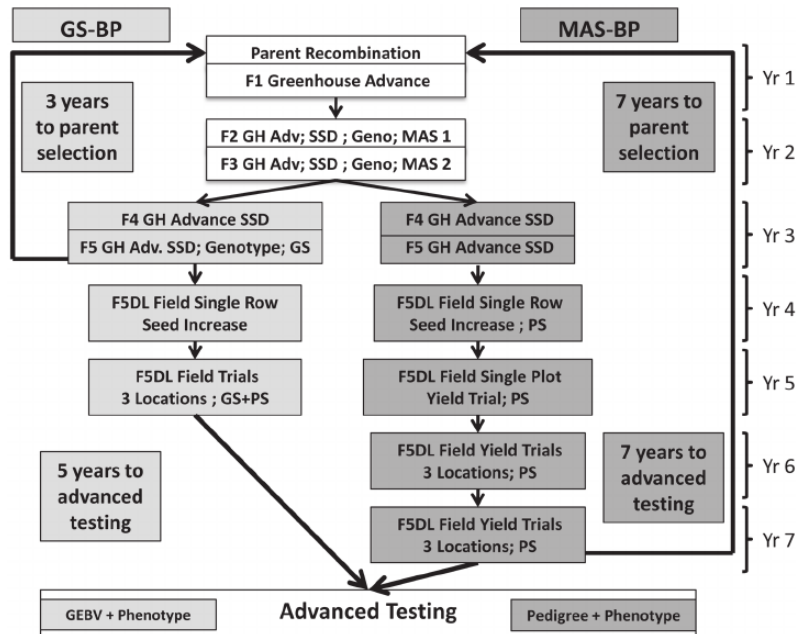
# Modular Breeding Program Simulator

Torsten Pook & Henner Simianer

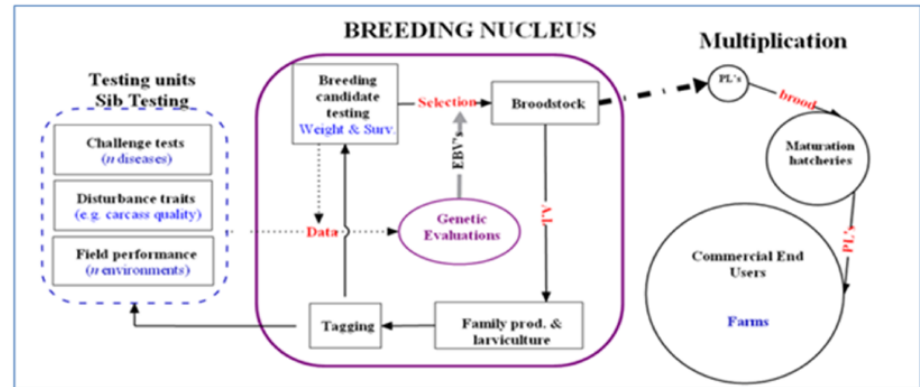




# Modern breeding programs are complex by nature



(Wheat breeding: Heffner et al., 2010)



(Fish breeding: Rye 2012)



# What are we interested in?

- What is our breeding objective?
    - Maintenance of genetic diversity
    - Genetic progress
    - Economic efficiency
    - Risk (variability of the outcome)
  - How to control it?
    - How many animals to use
    - Generate genotype / phenotype data of all animals
    - Mating scheme
    - Selection technique
    - And much more...
- Complex optimization problem!



# Possible ways to answer this?

- Experience of the breeder
- Learn from case studies
- Simulation study
- Cohort-based deterministic (ZPLAN+)
  - Reliant on underlying theory
  - E.g. Breeders equation:

$$R = i \cdot h \cdot \sigma_a$$

R: response to selection

i: selection intensity

h: heritability

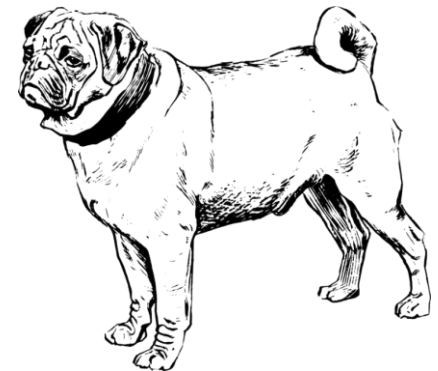
$\sigma_a^2$ : additive genetic variance

- Good approximation but formulas are limited to specific application and are constructed to handle „easy“ scenarios
  - Stochastic individual-based simulation



# What is MoBPS?

- **Modular Breeding Program Simulator**
- R-package (<https://github.com/tpook92/MoBPS>)
  - Highly efficient (Computational relevant parts written in C/C++)
  - Highly flexible (Cohort-based design)
  - Pre-implemented functions
  - Variety of different trait architectures
- Web-based application ([www.mobps.de](http://www.mobps.de))





# About the R-package

- Distributed via github: basically weekly updates
- Design philosophy:
  - Generate a tool that is able to simulate all breeding programs
  - When something is not yet possible and we see a general value in it, we are going to add it

## **Version 1.4.15 (29.10.19)**

Solve issues with running MoBPS without miraculix

Added BayesA,B,C, BL, BRR to BGLR options for BVE

New function: get.genotyped() to export which individuals are genotyped

Renamed ogc\_cAc to ogc.cAc for general uniformity

Updated documentation (especially for web-based application)

Minor update for miraculix (v0.9.7) - not MoBPS related

Added frozen version of the current that (Submission-version)

## **Version 1.4.10 (22.10.19)**

Mostly improvements to reduce memory requirement

New function: add.diag(). R-matrix is not required anymore

Clean-up of memory in ssGBLUP

Better handling of duplicated individuals (generated via copy.individuals)

## **Version 1.4.3 (16.10.19)**

MoBPSmaps 0.1.6 (Including maps for Wheat and Sorghum)

Removed typos in summary.population()

Updates to creating.diploid for trait generation via n.additive/dominant etc.

## **Version 1.4.2 (14.10.19)**

Variety of additions to json.simulation and user-interface

Added direct-mixed-model BVE for individuals without phenotype (vanRaden 2008)

Added use of Parent/Grandparent mean as breeding values

Further updates for miraculix/RandomFieldsUtils for compiler independent computing



# About the R-package

- Internally everything based on single individuals including the simulation of meiosis and recombination
- A variety of implemented functions:
  - Breeding values estimation
  - Single-step GBLUP
  - Optimum Genetic Contribution
  - Genome editing
- Trait architectures:
  - QTL-based: additive, dominance, qualitative/quantitative epistasis
  - Correlated traits



# About the R-package

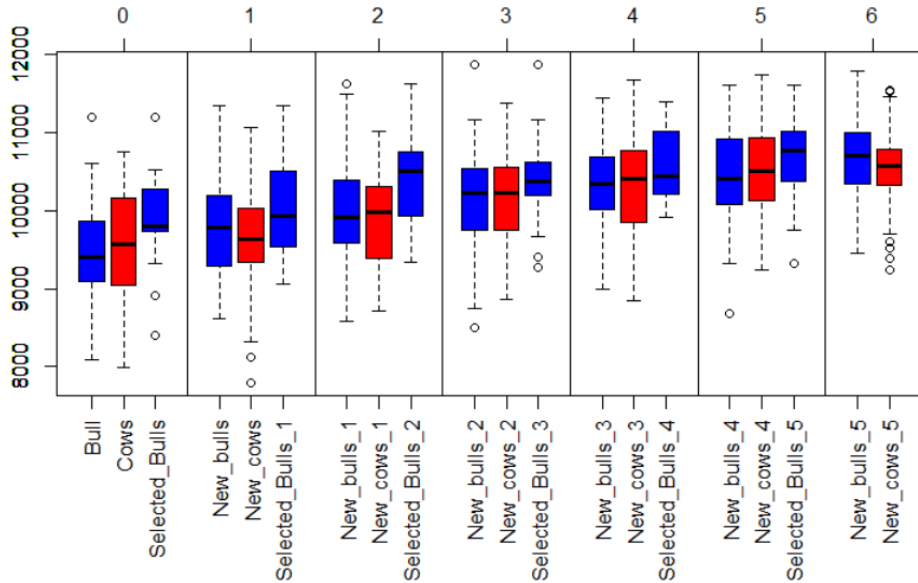
- Flexible storage structure
  - General information
  - Individual specific information
- Bitwise-storing of founder haplotypes
  - On-the-fly computation of genotypes & haplotypes
- R-package miraculix developed by Martin Schlather
  - Bit-wise computation and storage
  - 10 times faster than regular matrix multiplication in R
  - 15 times less memory



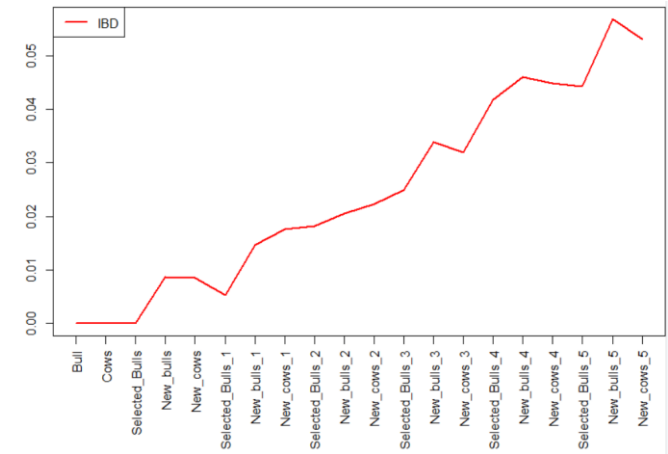


# About the R-package

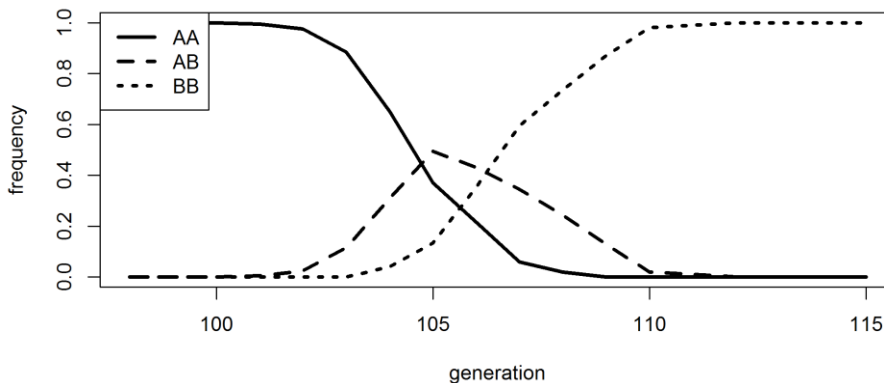
Genetic gain:



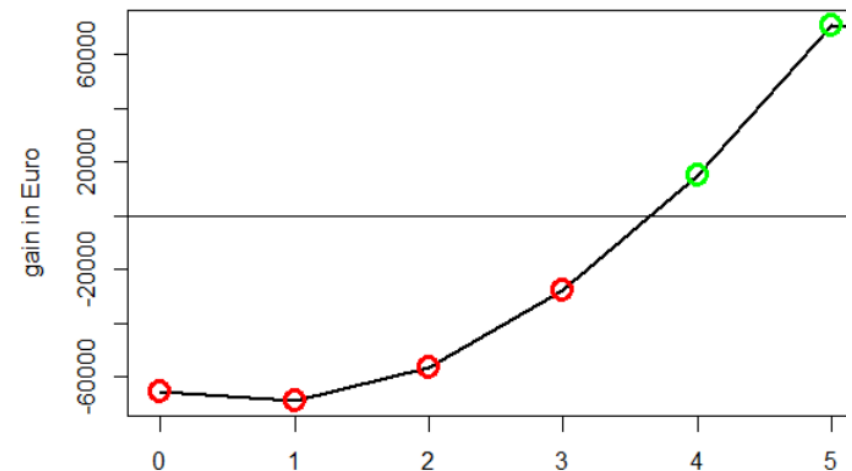
Inbreeding:



Genetic variation:



Economic gain:





# About the R-package

```
breeding.diploid <- function(population, mutation.rate = 10^-5, remutation.rate = 10^-5, recombination.rate = 1,
  selection.m = "random", selection.f = NULL, new.selection.calculation = TRUE, selection.function.matrix = NULL,
  selection.size = 0, ignore.best = 0, breeding.size = 0, breeding.sex = NULL, breeding.sex.random = FALSE,
  used.generations.m = 1, used.generations.f = NULL, relative.selection = FALSE, class.m = 0, class.f = 0,
  add.gen = 0, recom.f.indicator = NULL, recom.f.polynom = NULL, duplication.rate = 0,
  duplication.length = 0.01, duplication.recombination = 1, new.class = 0L, bve = FALSE, sigma.e = NULL, sigma.g = 100,
  new.bv.child = "mean", computation.A = "vanRaden", delete.haplotypes = NULL, delete.individuals = NULL,
  fixed.breeding = NULL, fixed.breeding.best = NULL, max.offspring = Inf, store.breeding.totals = FALSE, forecast.sigma.g = TRUE,
  multiple.bve = "add", multiple.bve.weights = 1, store.bve.data = FALSE, fixed.assignment = FALSE,
  reduce.group = NULL, reduce.group.selection = "random", selection.criteri = c(TRUE, TRUE), selection.criteria.type = c("bve", "bve"),
  same.sex.activ = FALSE, same.sex.sex = 0.5, same.sex.selfing = TRUE, selfing.mating = FALSE, selfing.sex = 0.5,
  praeimplantation = NULL, heritability = NULL, multiple.bve.scale = FALSE, use.last.sigma.e = FALSE,
  save.recombination.history = FALSE, martini.selection = FALSE, BGLR.bve = FALSE, BGLR.burnin = 500,
  BGLR.iteration = 5000, copy.individual = FALSE, dh.mating = FALSE, dh.sex = 0.5, n.observation = 1,
  bve.0isNA = TRUE, phenotype.bv = FALSE, standardize.bv = FALSE, standardize.bv.level = 100,
```

Only two functions are needed to perform all simulations.

You just have to memorize 260 input parameters and read our 58 page user manual and you are good to go!

```
add.architecture = NULL, ncore = 1, ncore.generation = 1, z.integer = FALSE, store.elect.ireq = FALSE,
backend = "doParallel", randomSeed = NULL, randomSeed.generation = NULL, Rprof = FALSE, miraculix = FALSE,
miraculix.mult = NULL, fast.compiler = 0, miraculix.cores = 1, store.bve.parameter = FALSE, miraculix.chol = TRUE,
best.selection.ratio.m = 1, best.selection.ratio.f = NULL, best.selection.criteria.m = "bx", best.selection.criteria.f = NULL,
best.selection.manual.ratio.m = NULL, best.selection.manual.ratio.f = NULL, bve.class = NULL, parallel.generation = FALSE,
name.cohort = NULL, display.progress = TRUE, max.ticks = Inf, combine = FALSE, repeat.mating = 1, time.point = 0,
creating.type = 0, multiple.observation = FALSE, new.bv.observation.gen = NULL,
new.bv.observation.cohorts = NULL, new.bv.observation.database = NULL, bve.gen = NULL, bve.cohorts = NULL,
bve.database = NULL, sigma.e.gen = NULL, sigma.e.cohorts = NULL, sigma.e.database = NULL, sigma.g.gen = NULL,
sigma.g.cohorts = NULL, sigma.g.database = NULL, gwas.gen = NULL, gwas.cohorts = NULL, gwas.database = NULL,
bve.insert.gen = NULL, bve.insert.cohorts = NULL, bve.insert.database = NULL, reduced.selection.panel.m = NULL,
reduced.selection.panel.f = NULL, breeding.all.combination = FALSE, depth.pedigree = Inf, copy.individual.keep.bve = TRUE,
bve.avoid.duplicates = TRUE, report.accuracy = TRUE, share.genotyped = 1, singlestep.active = FALSE,
remove.non.genotyped = TRUE, added.genotyped = 0, fast.what = FALSE, offspring.bve.parents.gen = NULL,
offspring.bve.parents.database = NULL, offspring.bve.parents.cohort = NULL, offspring.bve.offspring.gen = NULL,
offspring.bve.offspring.database = NULL, offspring.bve.offspring.cohort = NULL){
```



# A uniform language for breeding programs

- Every breeding program can be represented as a combination of boxes (nodes) and arrows (edges), (Simianer et al., in prep.)
- Nodes: cohorts of animals with certain properties
- Edges: breeding actions (e.g. selection, aging, reproduction)

## Edit Node

### Node *Young Cows*

Name *i*

Number of individuals *i*

Founder *i* ☒

Genotype generation type *i*

Upload genotypes (plink/vcf) *i*  dataset.vcf

Path genotypes

Phasing required *i* ☒

Sex *i*

Phenotyping Class *i*

Housing Cost Class *i*

Proportion of genotyped individuals *i*

## Edit Edge

### Edge *Young Cows* *Selected Cows*

Breeding Type *i*

Time needed *i*  Month

Selection Type *i*

Calculate reliability *i* ☐

Selection Index *i*

Selection Proportion *i*

Use offspring for BVE *i* ☐

Relationship Matrix *i*

Depth of Pedigree (# generation back) *i*

BVE Method *i*

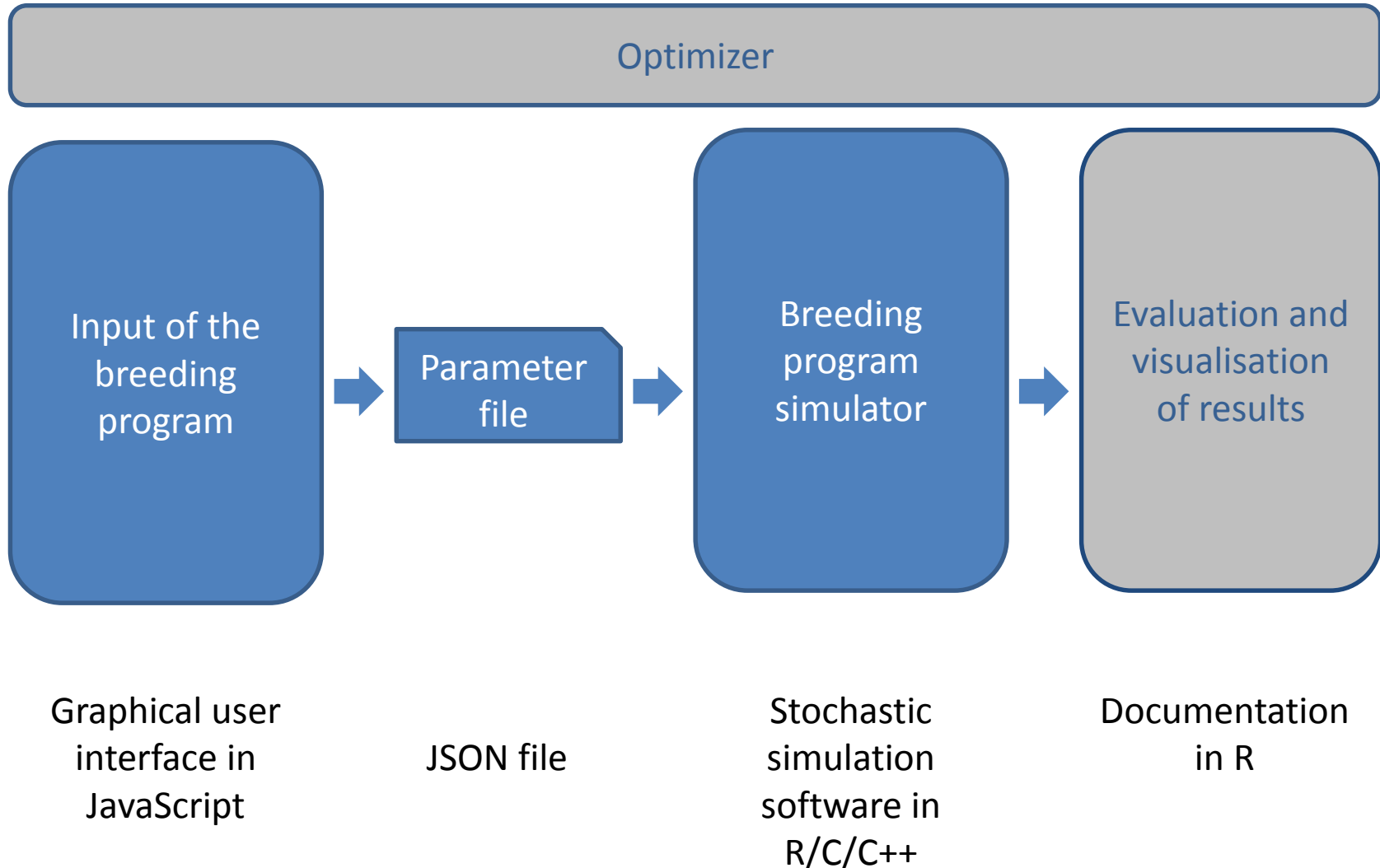
Cohorts used in BVE *i*

Young Cows

Selected Cows



# Web-based application





# Web-based application

- Available at [www.mobps.de](http://www.mobps.de)

## MoBPS Login

<input type="text"/>
<input type="password"/>
<input type="button" value="Login"/>

[Email Me:](#) For Questions and new account generation

Test-account during EAAP:  
User: EAAPguest  
pw: eaap2019

[GitHub:](#) For the R-package and source code

MoBPS was developed in the context of the EU project [IMAGE](#)  
Copyright © 2017 -- 2019 Torsten Pook



GEORG-AUGUST-UNIVERSITÄT  
GÖTTINGEN

**CiBreed**  
Center for Integrated Breeding Research





# Available modules

- Design your Genome
- Design your Traits
- Multiple Subpopulations
- Design your Selection Index
- Reasons for Culling
- Economy Parameters
- Draw your Breeding Scheme
- Analyze your Population



# Design your genome

## General Information

Project Name ⓘ

Maze

Advanced settings

Species ⓘ

Maize ▼

Time Unit ⓘ

Weeks ▼

Genetic Data

☐ Use Ensembl Map

☐ Upload Own Map (vcf/plink)

☒ Create customized Map

Ensembl Dataset ⓘ

Affymetrix Axiom Maize Genotyping Array ▼

Max. Number of SNPs ⓘ

30000

Enter the name of your project.  
You can load  
save/download/delete your  
project via the action bar and  
load different version to return  
to.



# Design your traits

## Phenotype Information ⓘ

[Add new phenotype](#)[Show/Hide 3 phenotypes](#)[Show/Hide QTLs](#)[Show/Hide phenotypic correlation](#)[Show/Hide genetic correlation](#)

Phenotype ⓘ	Unit ⓘ	Pheno. Mean ⓘ	Pheno. SD ⓘ	Heritability ⓘ	# polygenic loci ⓘ	Major QTL ⓘ	Value per unit (€) ⓘ	Show Cor
PH_V4		100	10	0.4	1000	0	0	<input checked="" type="checkbox"/> <input type="button" value="X"/>
PH_V6		100	10	0.5	1000	0	0	<input checked="" type="checkbox"/> <input type="button" value="X"/>
PH_final		100	10	0.7	500	0	0	<input checked="" type="checkbox"/> <input type="button" value="X"/>

## Residual Correlation ⓘ

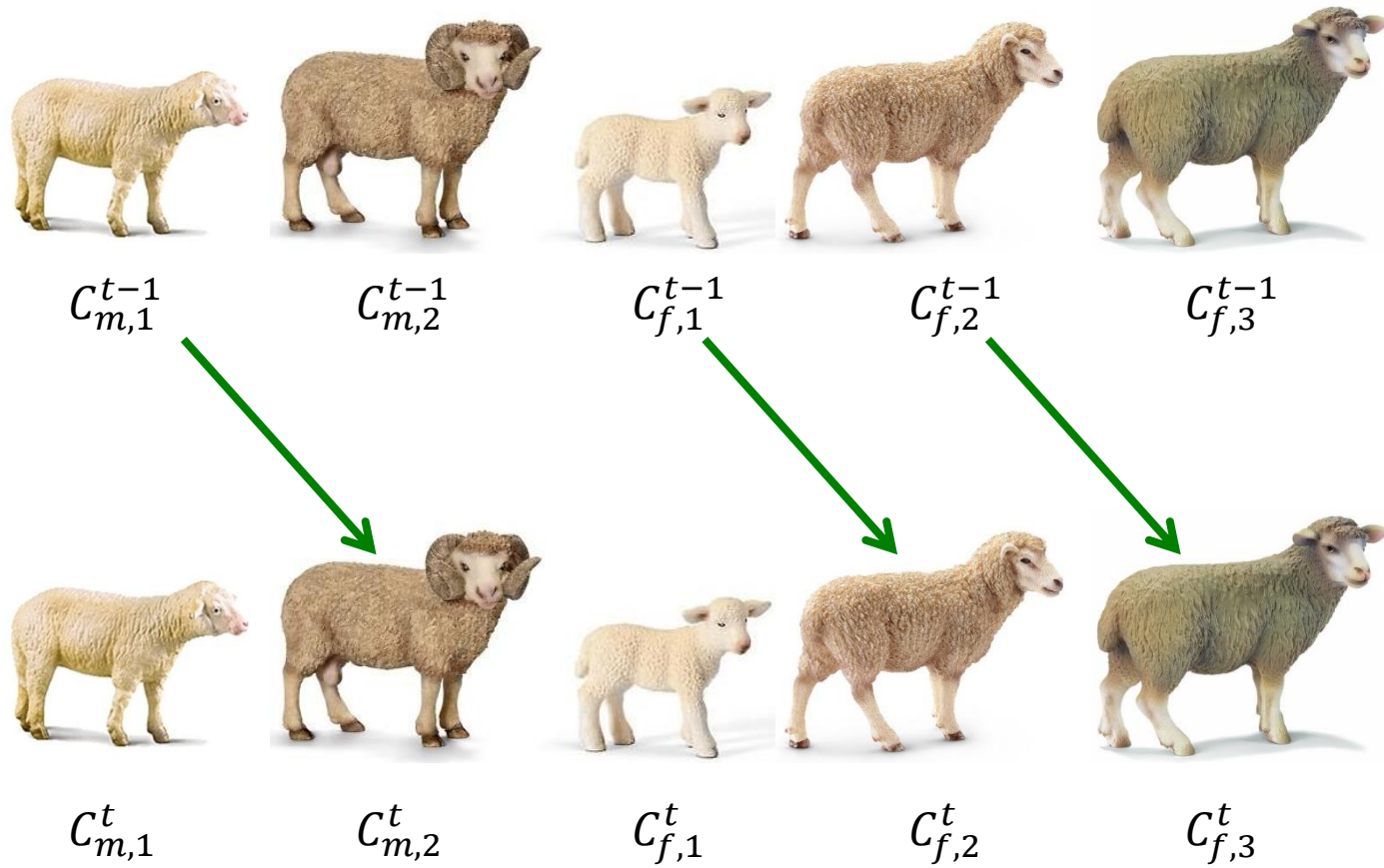
	PH_V4	PH_V6	PH_final
PH_V4	1	0	0
PH_V6	<input type="text" value="0"/>	1	0
PH_final	<input type="text" value="0"/>	<input type="text" value="0"/>	1

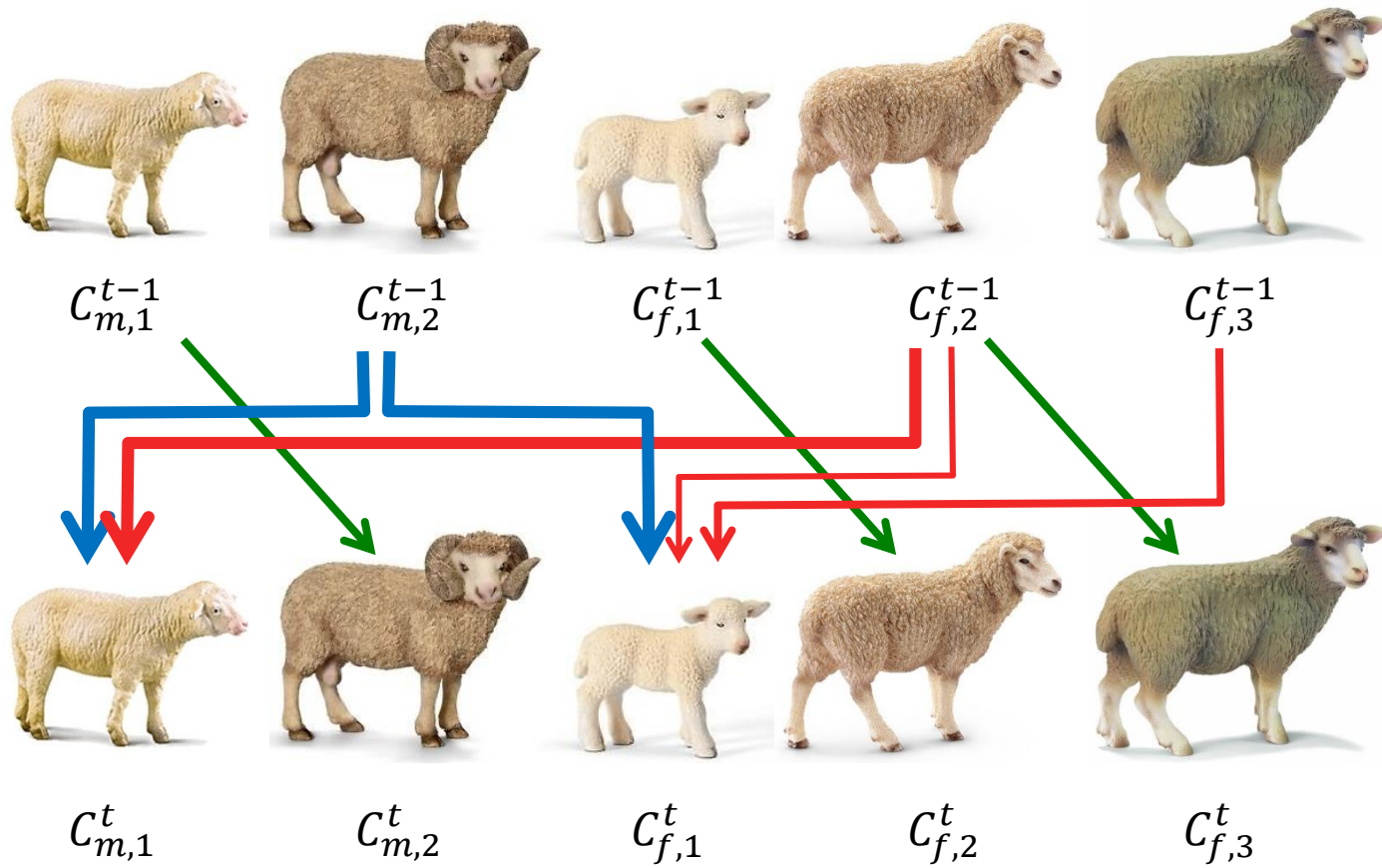
## Genetic Correlation ⓘ

	PH_V4	PH_V6	PH_final
PH_V4	1	0.96	0.54
PH_V6	<input type="text" value="0.96"/>	1	0.65
PH_final	<input type="text" value="0.54"/>	<input type="text" value="0.65"/>	1

Enter Phenotypic correlation instead of residual correlation ☐









# And this is how it looks in MoBPS

## Breeding Scheme ①

*Important: Please AVOID node names with : and \_ . In particular avoid trailing numbers like ABC\_1. Repeated nodes will use this syntax!*

### Legends

Nodes:

□ Male

□ Female

□ Both

Edges:

➤ Selection

➤ Reproduction

➤ Aging

➤ Combine

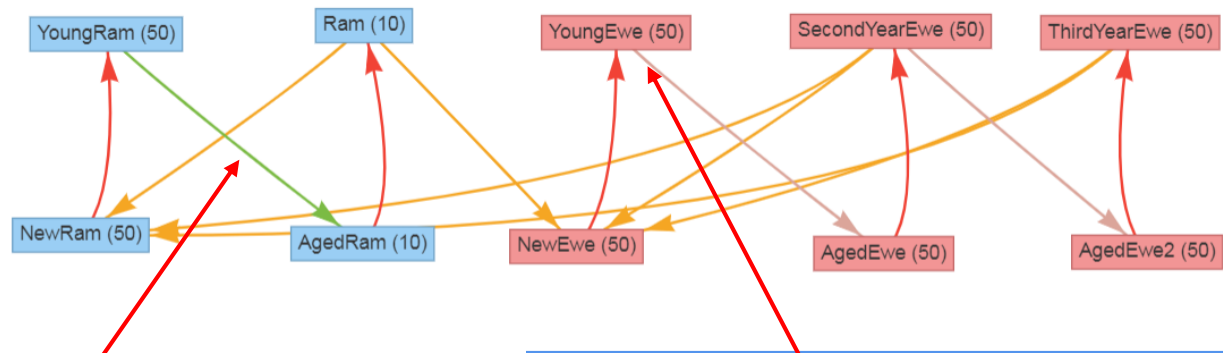
➤ Repeat

➤ Split

➤ Cloning

➤ Selfing

➤ DH-Production



### Edit Edge

#### Edge YoungRam\_AgedRam

Breeding Type ①

Selection

Time needed ①

0 Weeks

Selection Type ①

BVE

BVE Method ①

Direct Mixed-Model

Selection Proportion ①

0.2 (calculated based on # Individuals)

Relationship Matrix ①

VanRaden

Cohorts used in BVE ①

Manual select

YoungRam:Same Repeat

Ram:Same Repeat

YoungEwe:Same Repeat

SecondYearEwe:Same Repeat

Select cohorts for manual select: ①

### Edit Node

#### Node YoungEwe

Name ①

YoungEwe

Number of individuals ①

50

Founder ①

☒

Genotype generation type ①

Random-sampling

Sex ①

Female


Phenotyping Class ①

Fully phenotyped

save cancel



# Task I

- MoBPS interface: [www.mobps.de](http://www.mobps.de)
  - 1.1) Open the template „Sheep\_breeding“
  - 1.2) Simulate the project 
  - 1.3) How did the genomic values of the new rams change over time?
  - 1.4) Look at inbreeding rates for the new rams.



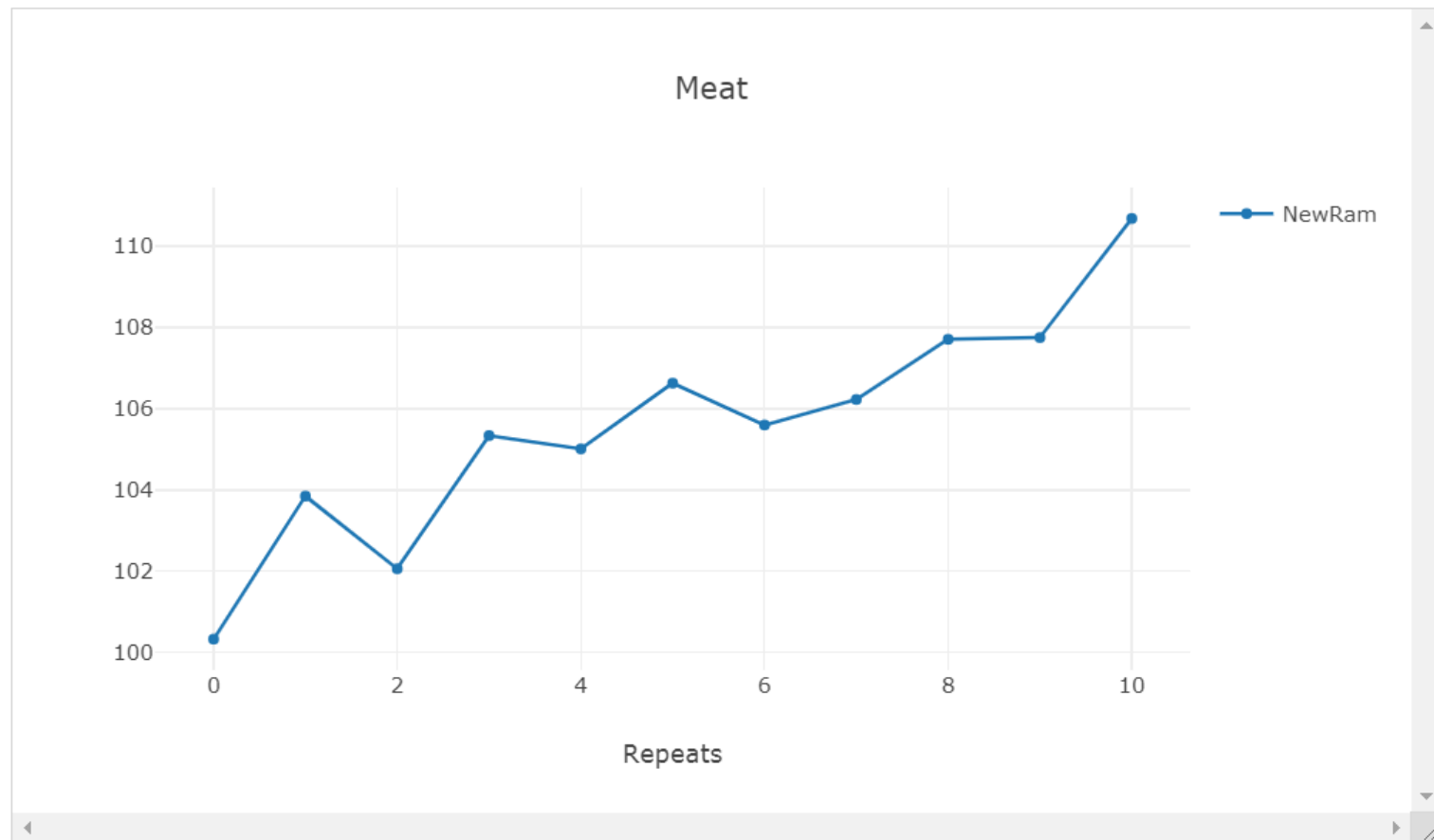
# Solutions – Task I

## *Results: True Breeding Values*

Select plotting type: By Repeats ▼

Select cohorts (multiple selection possible): Plot Results

NewRam (10 Repeats) ×





# Solutions – Task I

## Results: True Breeding Values

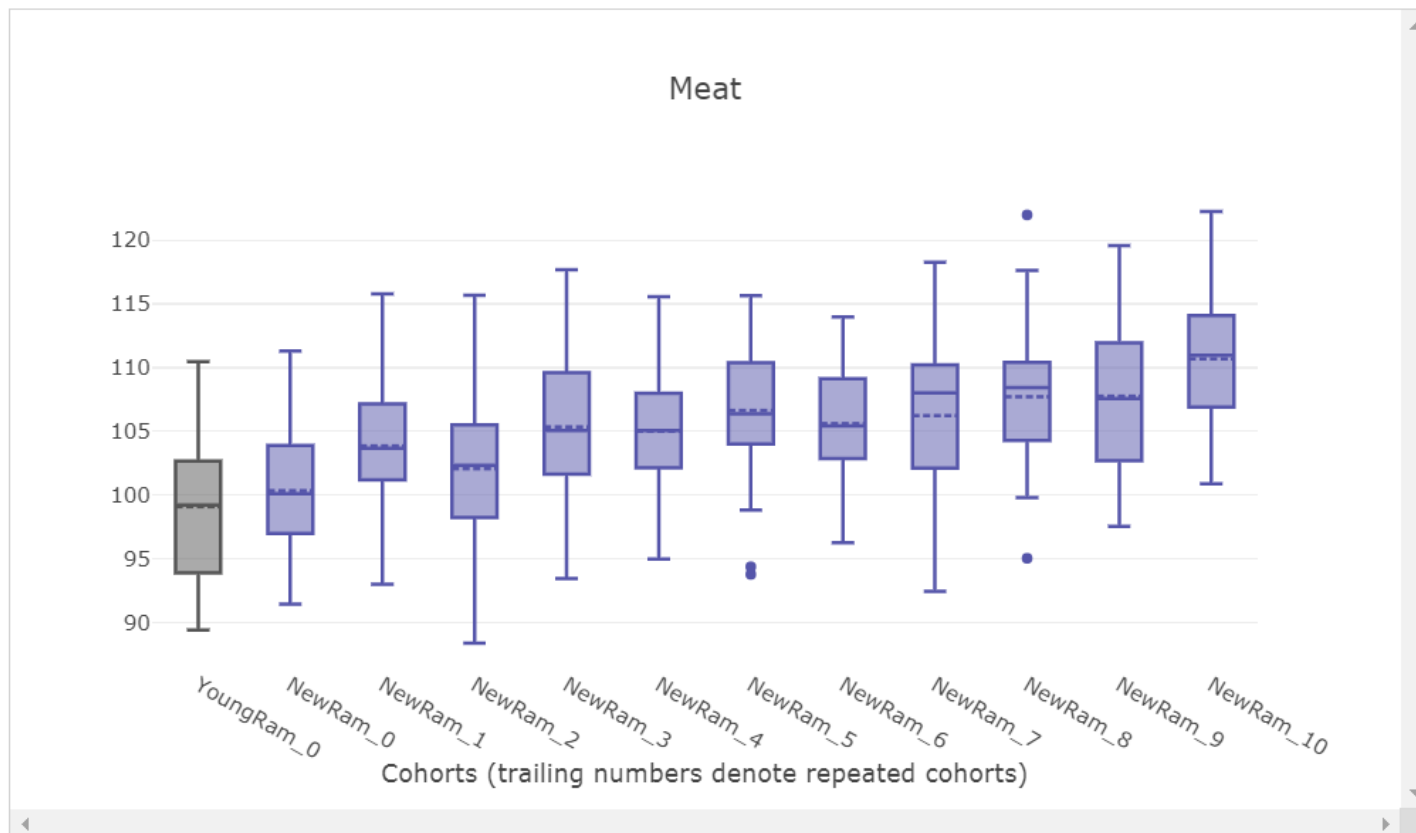
Select plotting type: By Cohorts ▾

Select cohorts (multiple selection possible): Plot Results

YoungRam (0 Repeats) ×

NewRam (10 Repeats) ×

× ▾





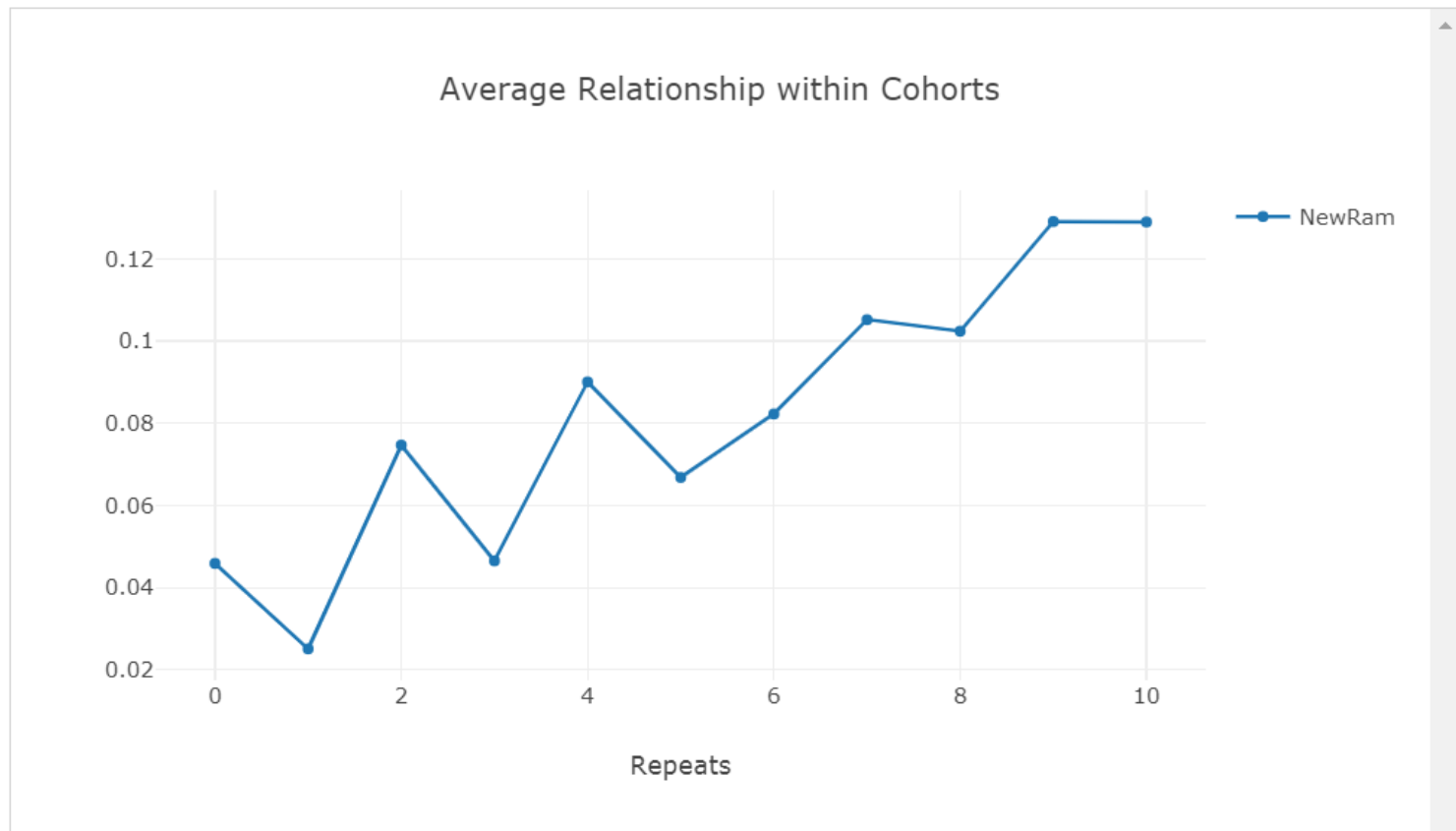
# Solutions – Task I

## *Results: Relationship and Inbreeding within Cohorts*

Select plotting type: By Repeats ▼

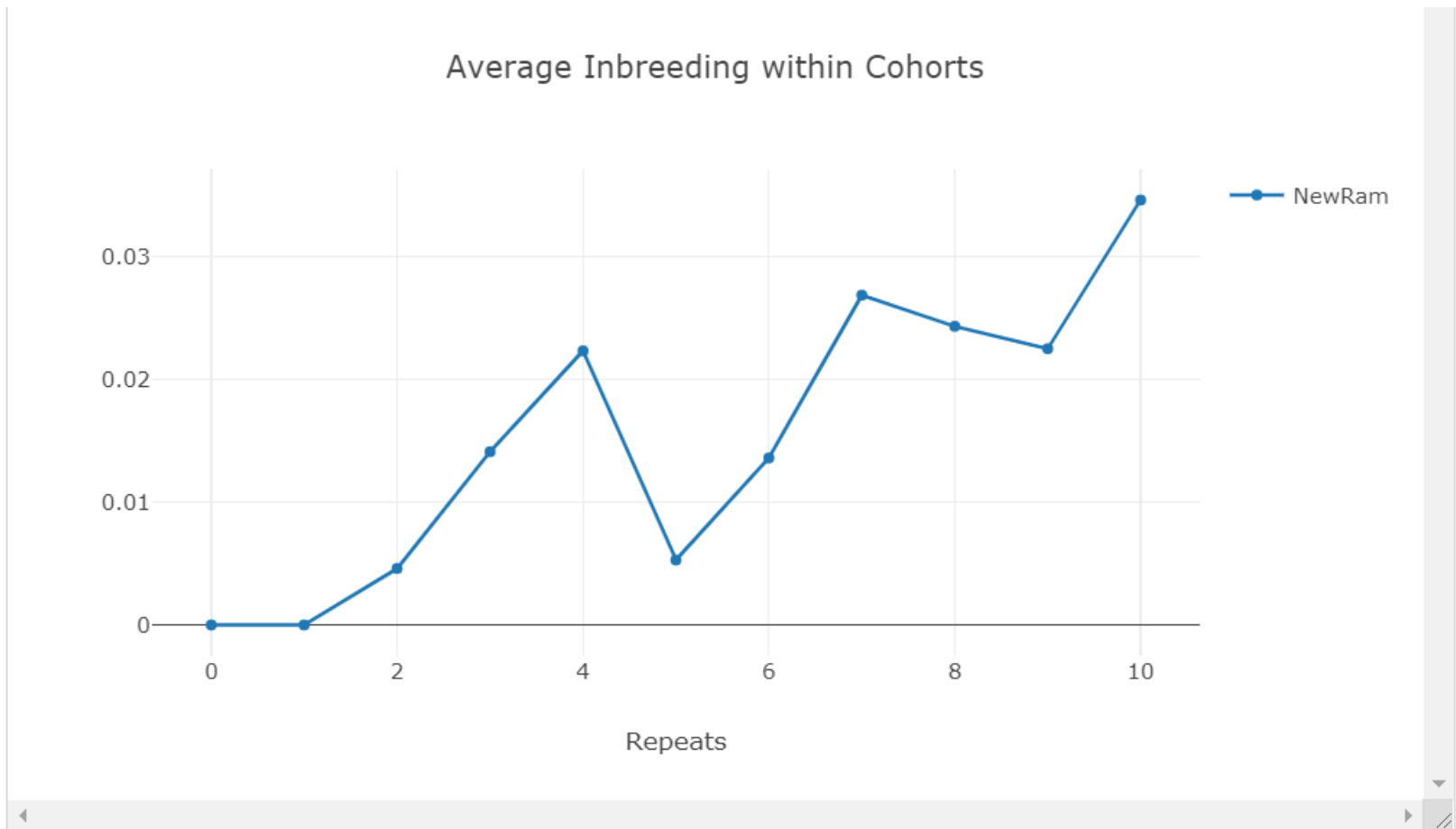
Select cohorts (multiple selection possible): [Plot Results](#)

NewRam (10 Repeats) x





# Solutions – Task I







# Introducing genetic material

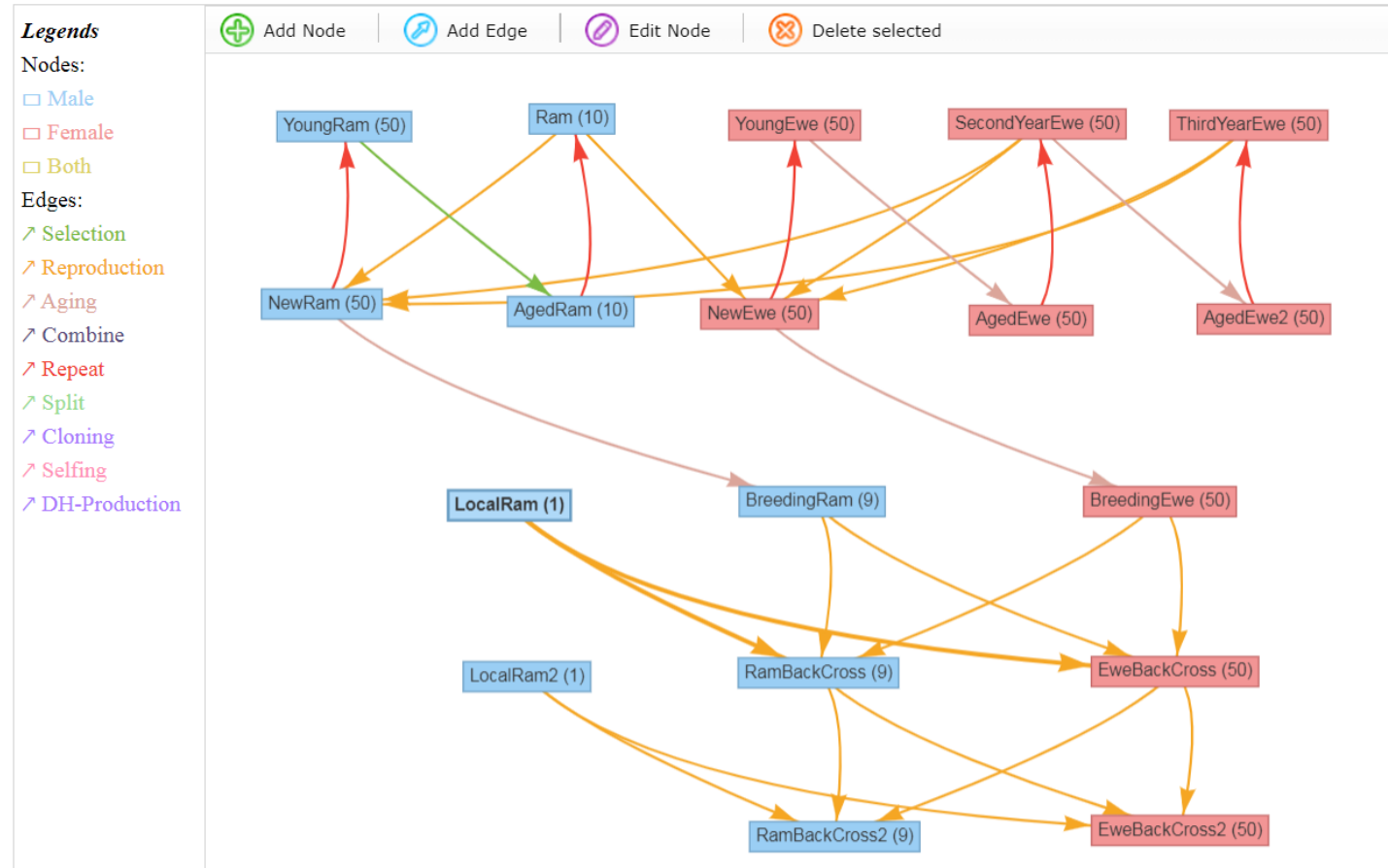
- Open the template „Sheep\_diversity“
- Task:
  - 2.1) Perform two generations of back-crosses. Use 9 rams from the breeding population and 1 local ram as paternal parent.
  - 2.2) Look at the allele frequency of the major QTL in the new cohorts. What allele frequency would you expect?
  - 2.3) Use 10 local rams and 50 rams from the breeding population. Apply selection on the male side perform introgression while at least maintaining the performance level in terms of the meat-trait.
  - 2.4) How does the allele frequency in the major QTL for the fitness trait change over time?



# Solutions – Task II

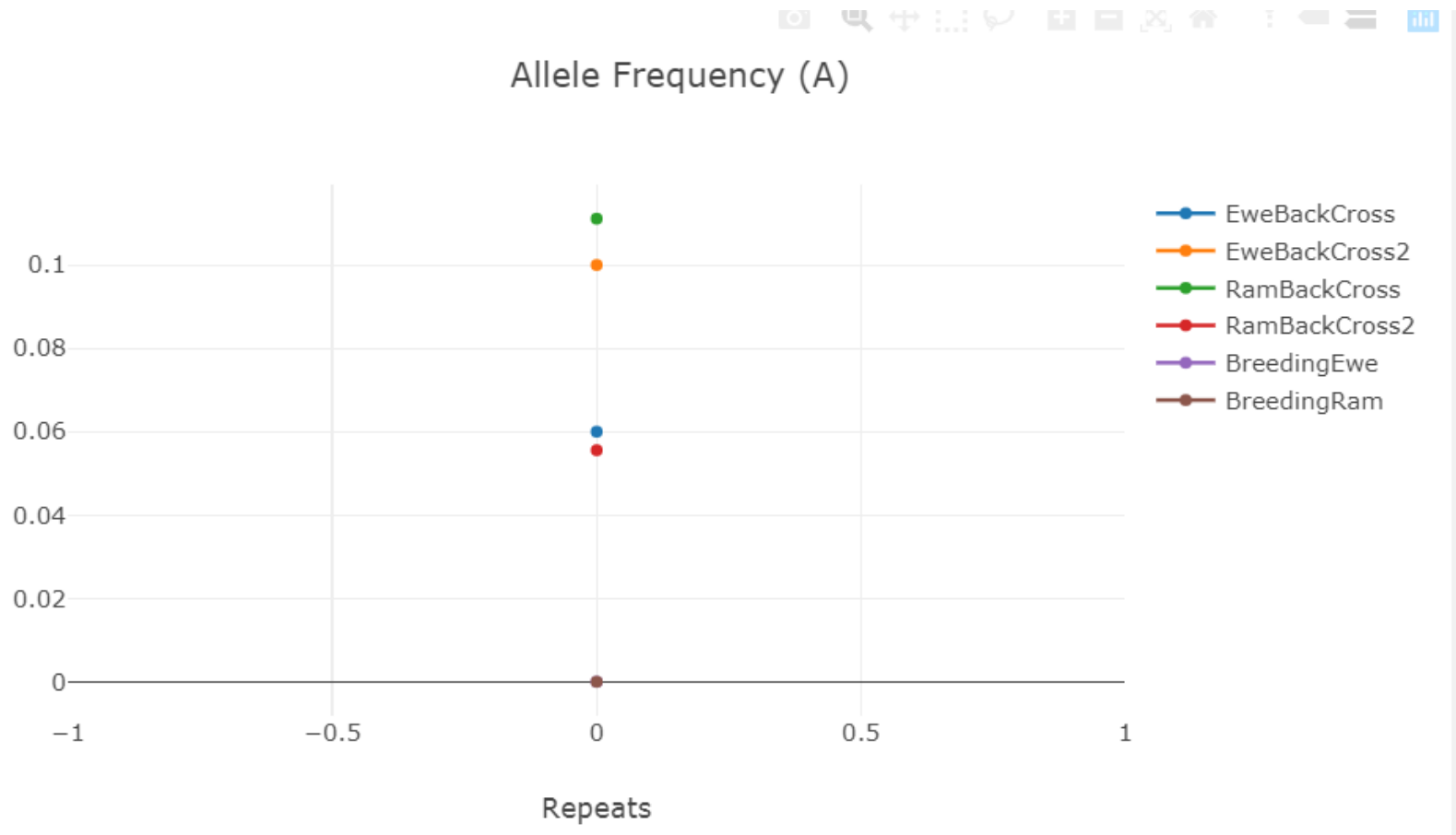
## Breeding Scheme ①

**Important: Please AVOID node names with : and \_ . In particular avoid trailing numbers like ABC\_1. Repeated nodes will use this syntax!**





# Solutions – Task II



- Only 10% of the males carry the allele. Therefore we would expect  $p = 0.05$  in the first back cross.



# Solutions – Task II

## Breeding Scheme ①

**Important: Please AVOID node names with : and \_ . In particular avoid trailing numbers like ABC\_1. Repeated nodes will use this syntax!**

### Legends

Nodes:

□ Male

□ Female

□ Both

Edges:

↗ Selection

↗ Reproduction

↗ Aging

↗ Combine

↗ Repeat

↗ Split

↗ Cloning

↗ Selfing

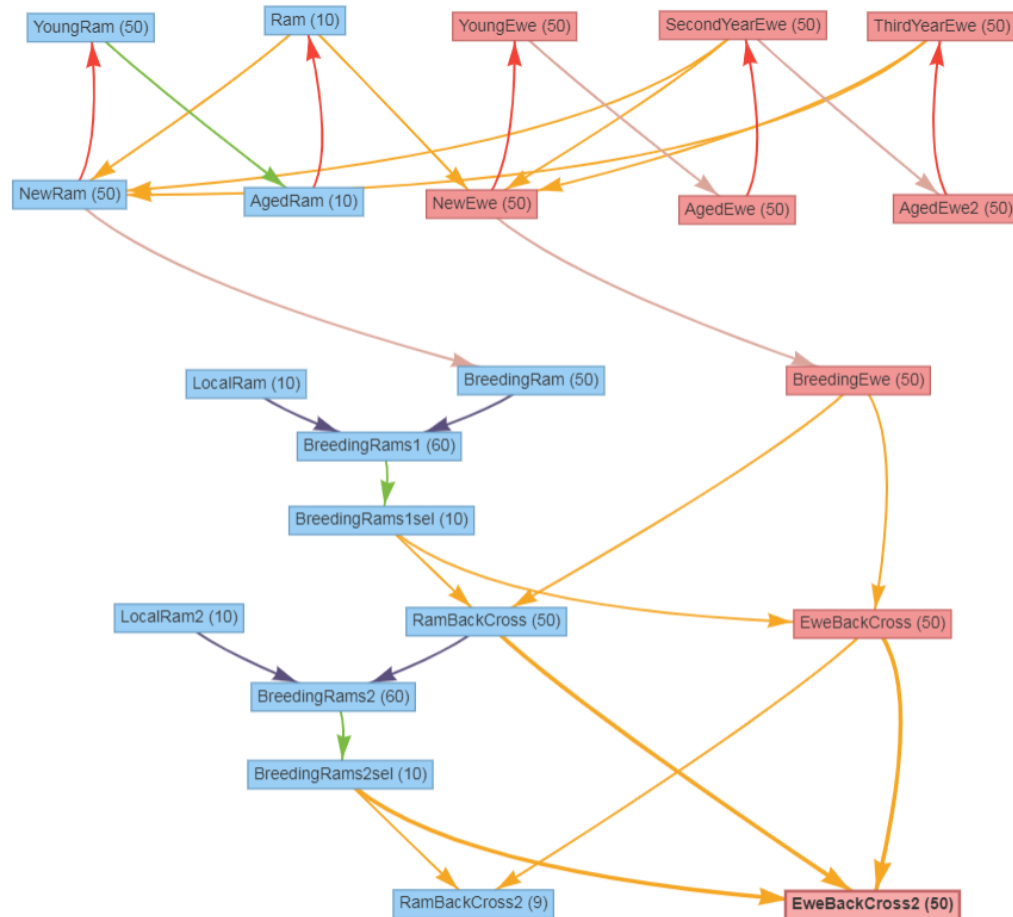
↗ DH-Production



Add Node



Add Edge





# Solutions – Task II

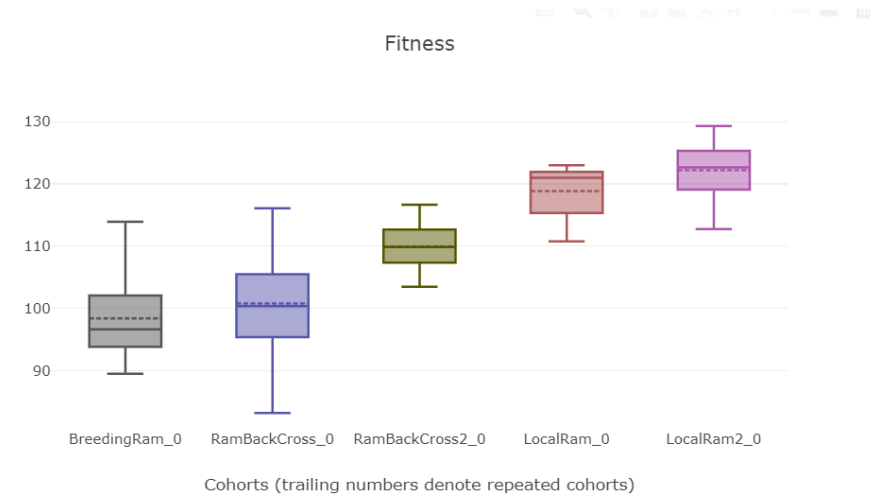
- Selection with equal weight on both traits

**Results: True Breeding Values**

Select plotting type: By Cohorts ▾

Select cohorts (multiple selection possible): Plot Results

BreedingRam (0 Repeats) ✕ RamBackCross (0 Repeats) ✕ RamBackCross2 (0 Repeats) ✕ LocalRam (0 Repeats) ✕  
LocalRam2 (0 Repeats) ✕





# Solutions – Task II

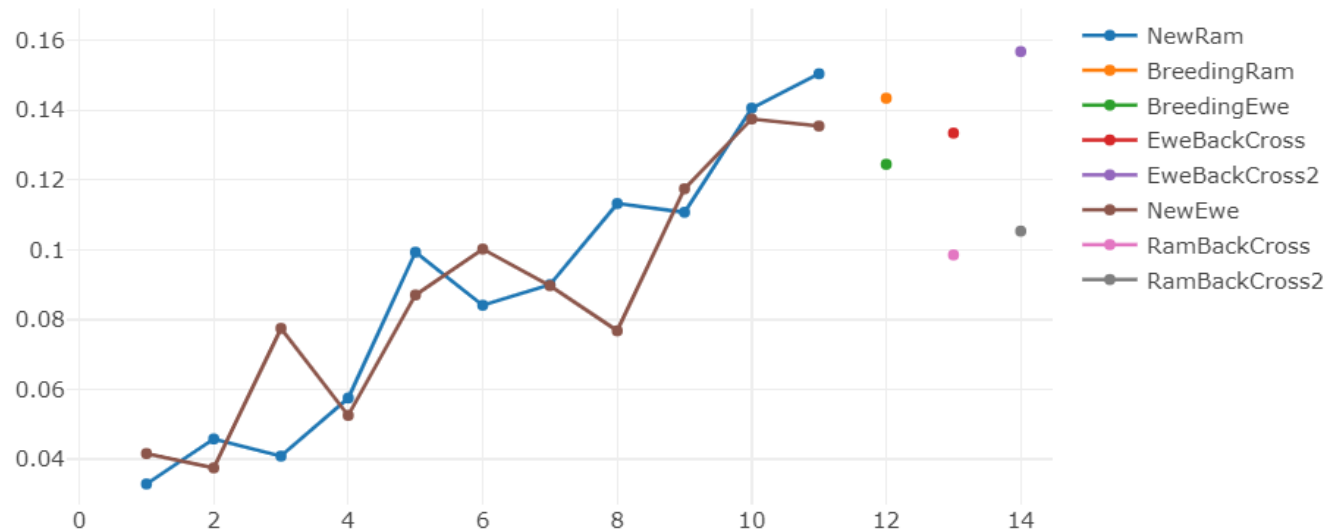
## Results: Relationship and Inbreeding within Cohorts

Select plotting type:

Select cohorts (multiple selection possible):

NewRam (10 Repeats) × BreedingRam (0 Repeats) × BreedingEwe (0 Repeats) × EweBackCross (0 Repeats) ×  
EweBackCross2 (0 Repeats) × NewEwe (10 Repeats) × RamBackCross (0 Repeats) × RamBackCross2 (0 Repeats) ×

Average Relationship within Cohorts





# Solutions – Task II

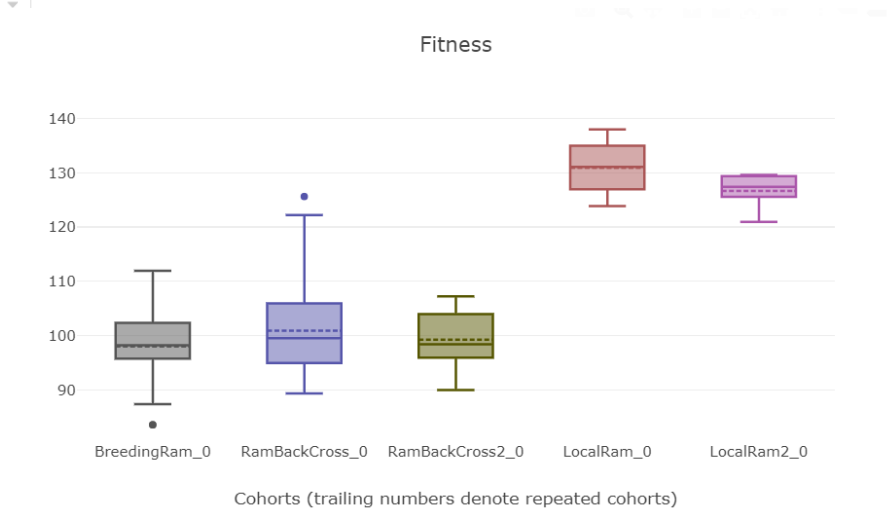
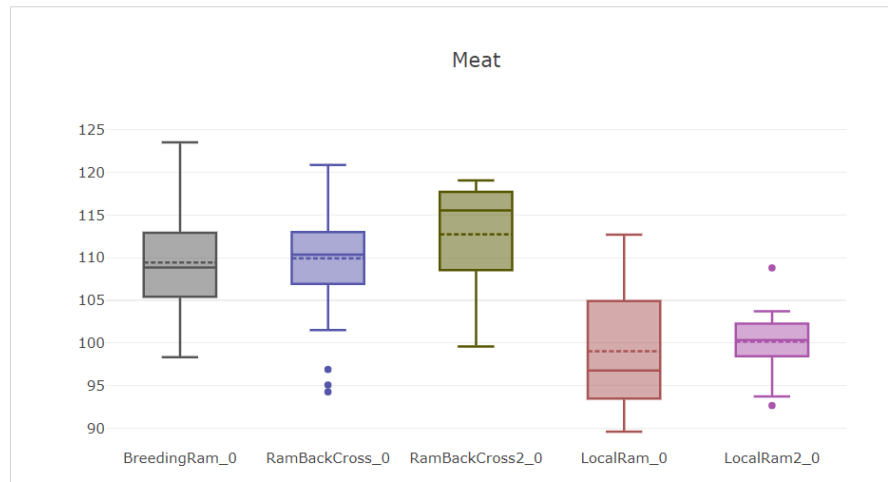
- 3 Index Points on Meat and 1 Index Point on Fitness

## Results: True Breeding Values

Select plotting type:

Select cohorts (multiple selection possible):

BreedingRam (0 Repeats) × RamBackCross (0 Repeats) × RamBackCross2 (0 Repeats) × LocalRam (0 Repeats) ×  
LocalRam2 (0 Repeats) ×





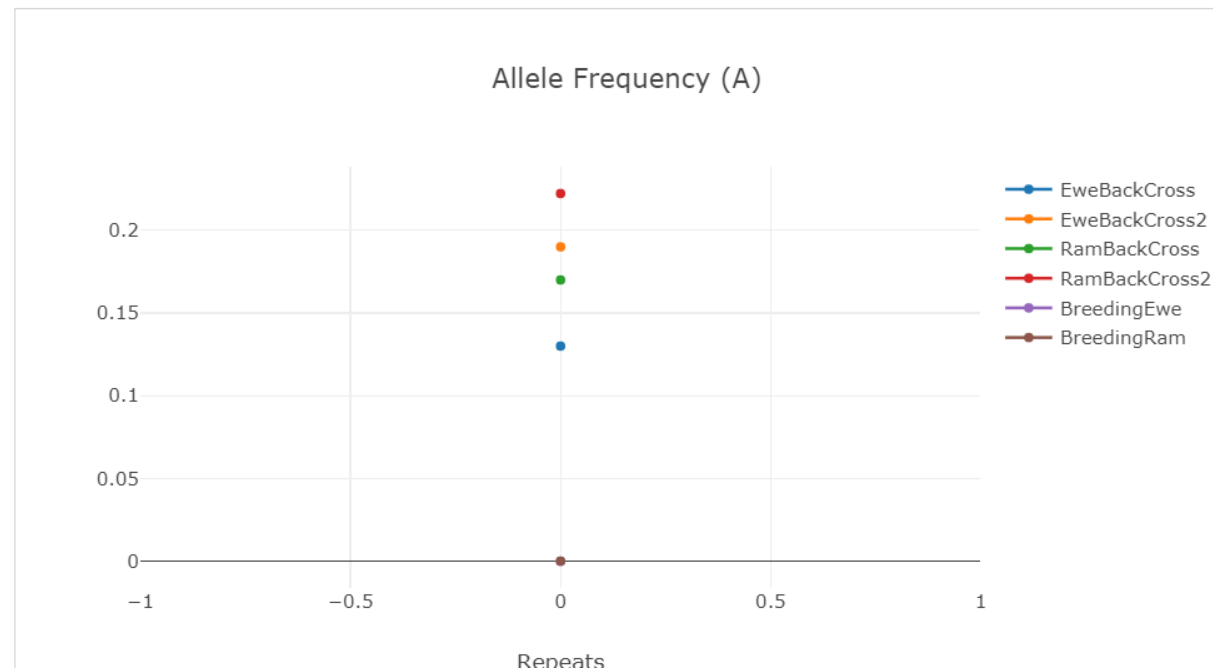
# Solutions – Task II

Results: Major QTLs (Allele Frequency, exp./obs. Heterozygosity)

Select Trait :  Select QTL :

Select Cohorts (multiple selection possible):

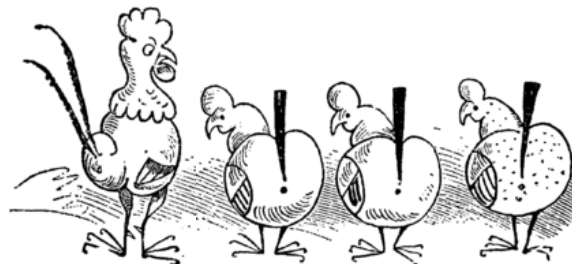
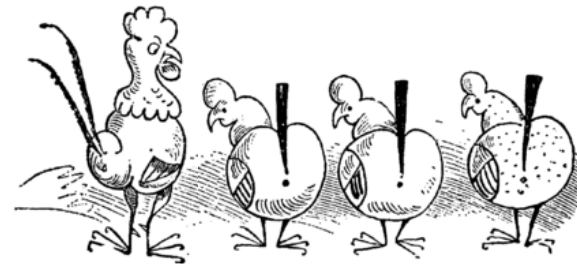
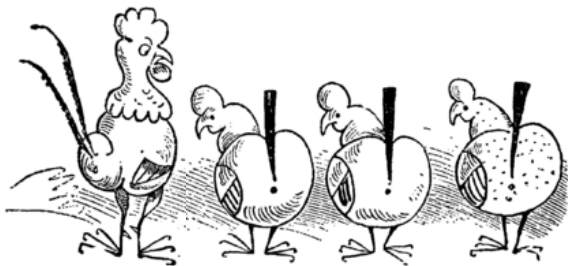
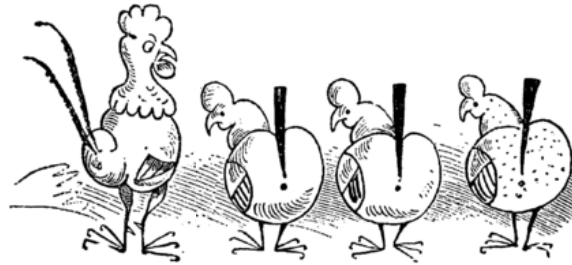
EweBackCross (0 Repeats) × EweBackCross2 (0 Repeats) × RamBackCross (0 Repeats) ×  
RamBackCross2 (0 Repeats) × BreedingEwe (0 Repeats) × BreedingRam (0 Repeats) ×



- More weight on fitness leads to higher allele frequency in the major QTL



# Conservation breeding





# Task III

3.1) Simulate 20 generations of a breeding program

– Assume: 5 Boxes with 5 hen and 1 cock each

3.2) How do inbreeding rates compare to a random mating environment?

3.3) Can you think of further modifications of the breeding program to reduce inbreeding



# Solutions – Task III

## Breeding Scheme ①

*Important: Please AVOID node names with : and \_ . In particular avoid trailing numbers like ABC\_1. Repeated nodes will use this syntax!*

### Legends

Nodes:

□ Male

□ Female

□ Both

Edges:

↗ Selection

↗ Reproduction

↗ Aging

↗ Combine

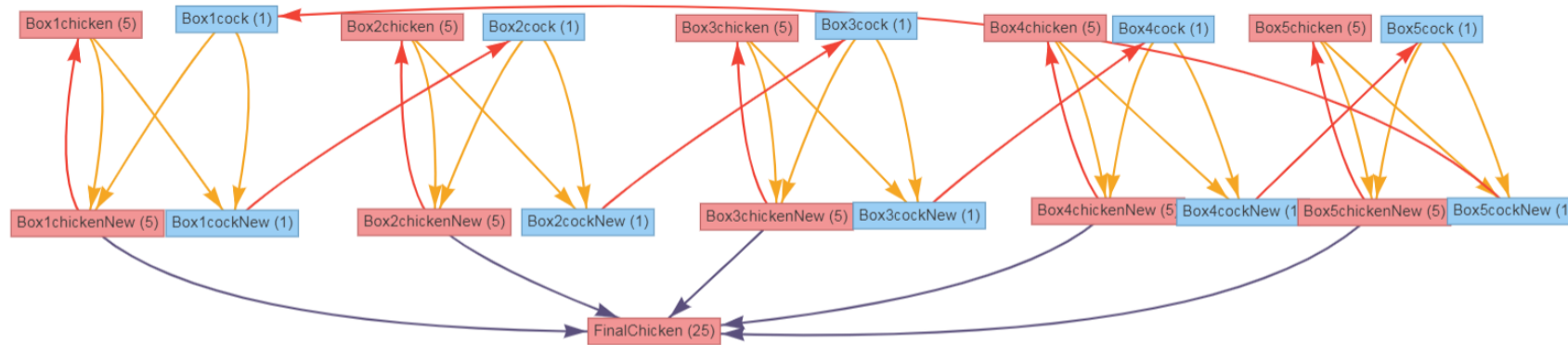
↗ Repeat

↗ Split

↗ Cloning

↗ Selfing

↗ DH-Production





# Solutions – Task III

## *Results: Relationship and Inbreeding within Cohorts*

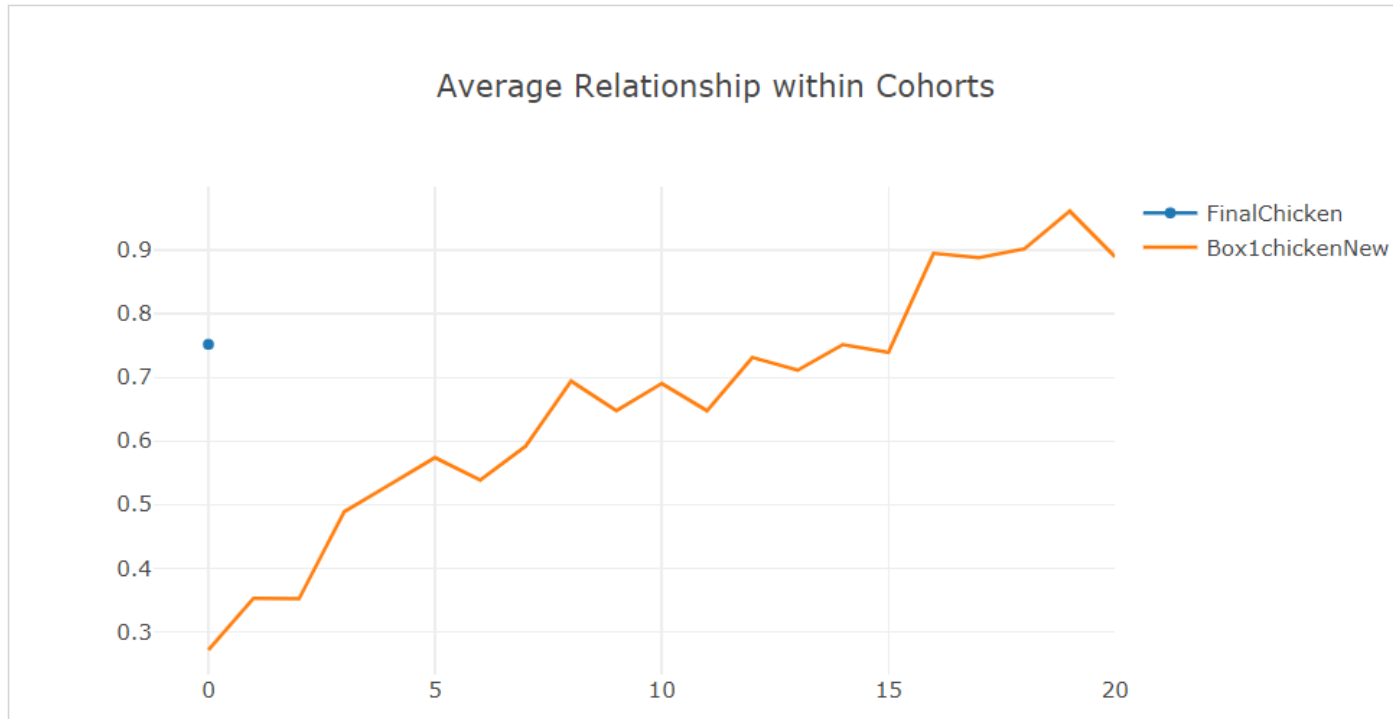
Select plotting type:

Select cohorts (multiple selection possible):

FinalChicken (0 Repeats) ×

Box1chickenNew (20 Repeats) ×

×



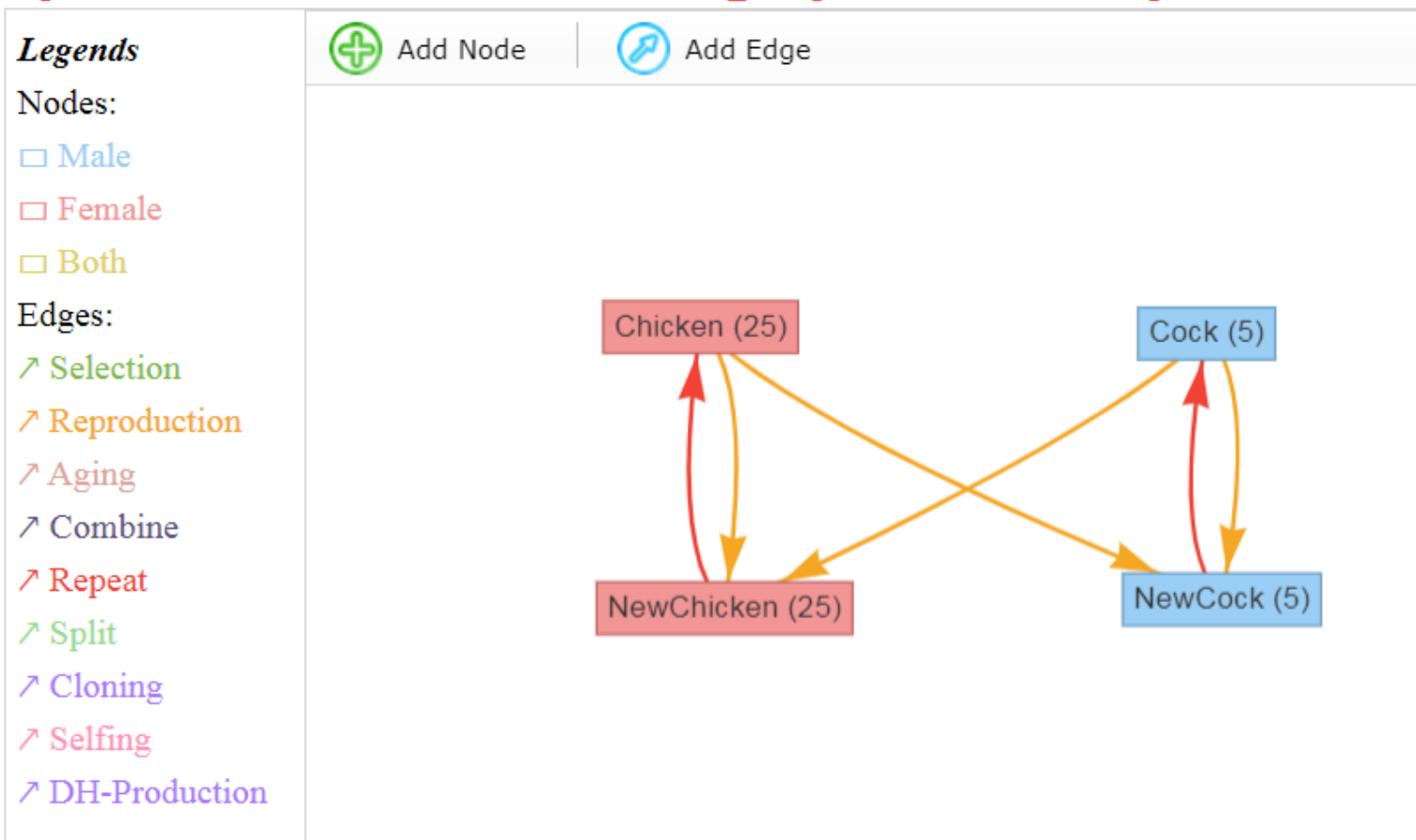
Note that inbreeding between the boxes is lower than for a single cohort!



# Solutions – Task III

## Breeding Scheme ⓘ

**Important:** Please *AVOID* node names with : and \_ . In particular avoid trailing numbers like AB1





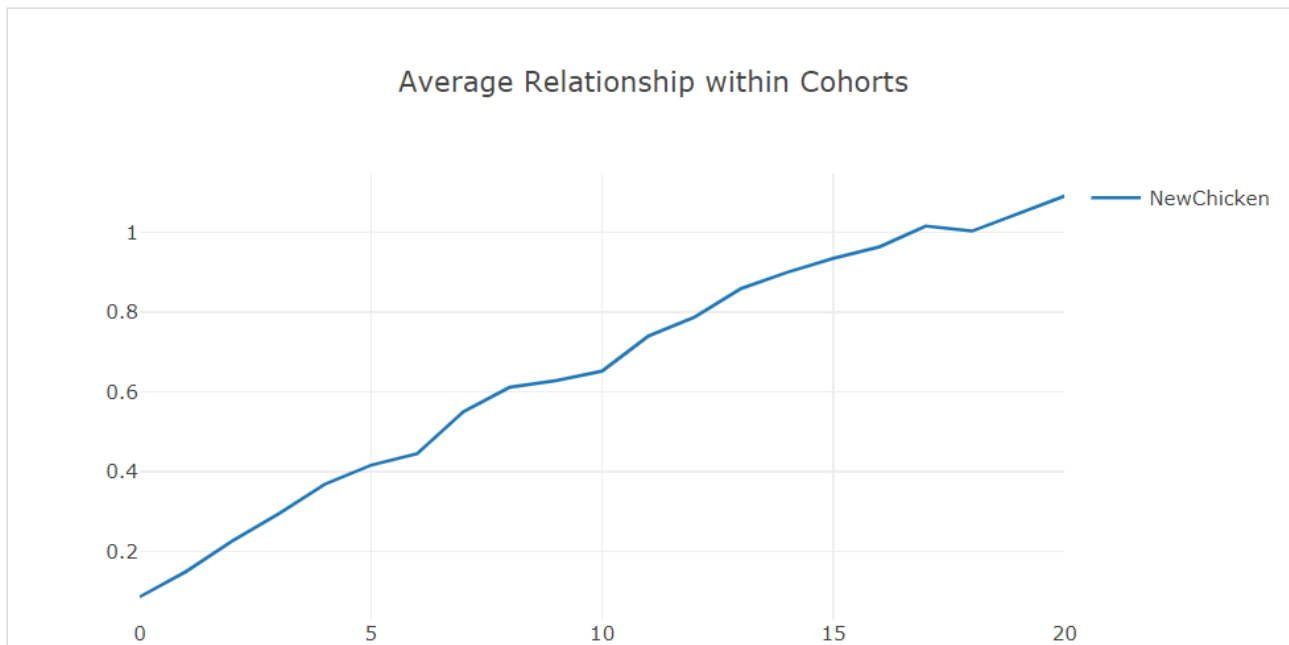
# Solutions – Task III

## *Results: Relationship and Inbreeding within Cohorts*

Select plotting type:

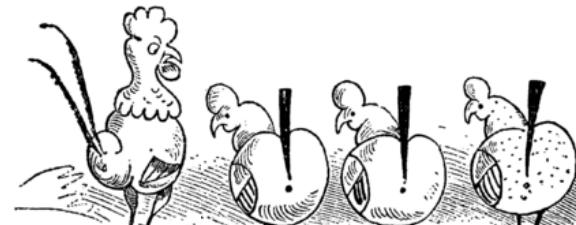
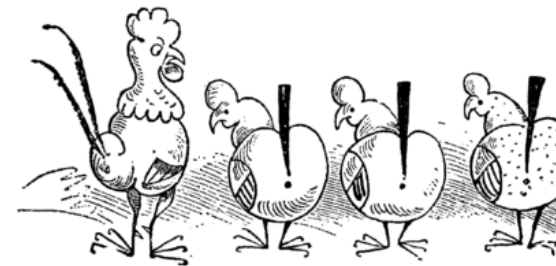
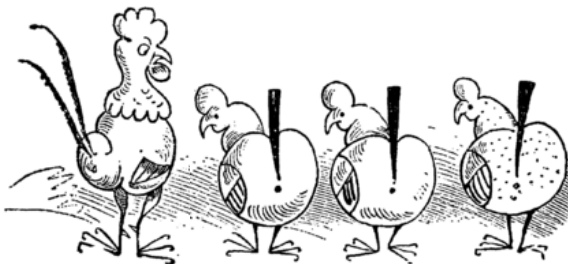
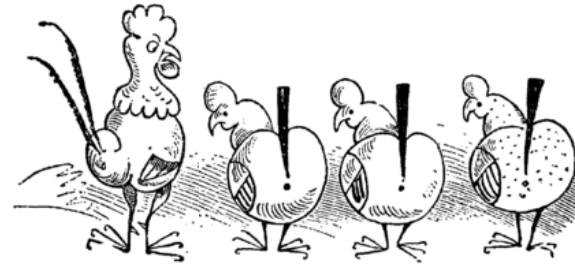
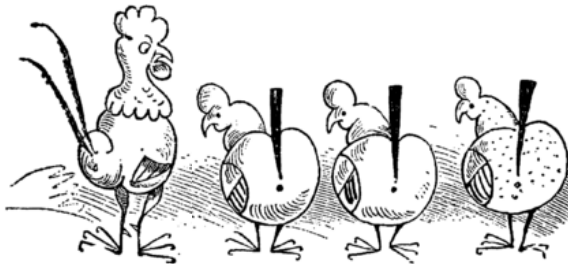
Select cohorts (multiple selection possible):

NewChicken (20 Repeats)



# Solutions – Task III

- Task 3.3)
  - Use each hen exactly once for reproduction of new hens
  - Do not always rotate to the same cohort





# Working outside of the Web-interface

- Our server provides:
  - 64 GB Memory
  - 20 cores

→ Not enough for large scale simulation studies!
- Option 1: Use the R-package directly
- Option 2: Export json-file from the interface





# Task IV

- Export the json-file of the template „Simple\_Cattle“
- Install R (<https://cran.r-project.org/>)
- R-package MoBPS (<https://github.com/tpook92/MoBPS>)
  - Make sure to install version 1.4.22!
  - You also need the R-package jsonlite
- Simulate the breeding program (`(json.simulation() )`)
- Generate a summary of your population (`( summary() )`)
- Generate a histogram of the phenotypes of the cohort: „CowsSecondYear“ (`( get.pheno() )`)
- Compare LD between the cohorts „Bull“, „NewBulls\_5“ and „SelectedBulls“ (`(ld.decay() )`):
- Export the genotypes of the cohort „NewBulls\_5“ and „NewCows\_5“ to PLINK files  
(`ped/map`) (`(write.pedmap() )`)



# Task IV - Solutions



Navigation Open

## MoBPS

Load a new/existent project from your own database:

Project:  Version:   
Exemplary Templates:

You are assigned to User Class Admin. This enables you to use 20 Cores and maximum run time of 120 hours

### General Information

Project Name ⓘ	<input type="text" value="Simple_Cattle"/>
Advanced settings	<input type="checkbox"/>
Species ⓘ	<input type="text" value="Cattle"/>
Time Unit ⓘ	<input type="text" value="Years"/>
	<input type="radio"/> Use Ensembl Map

- Load in the Project „Simple\_Cattle“
- Press the „Export Project“ – button in the navigation
- R-related: task1.R



# Task IV - Solutions

Install Packages

Install from:  
Package Archive File (.zip; .tar.gz) ▼

Package archive:  
C:/Users/pook/Desktop/R-Stuff/MoBPS\_1.4.22.tar.gz Browse...

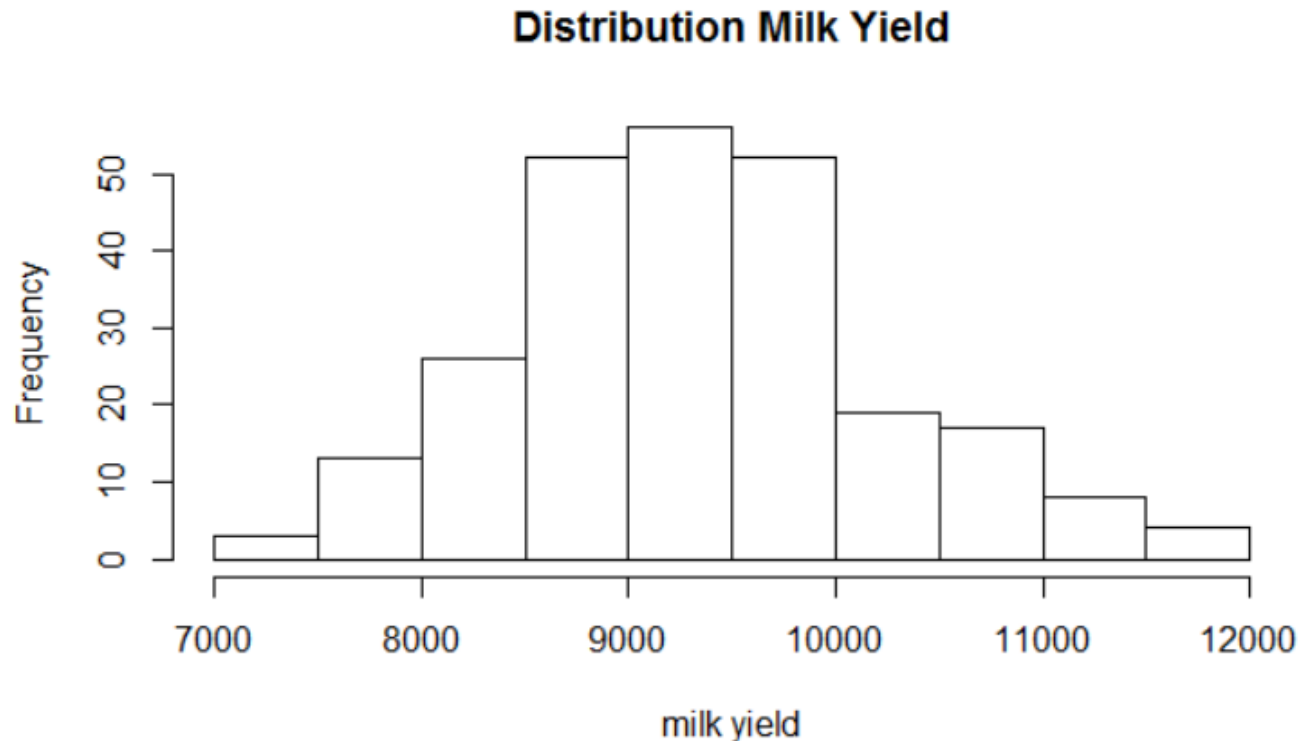
Install to Library:  
C:/Users/pook/Documents/R/R-3.6.0/library [Default] ▼

Install Cancel



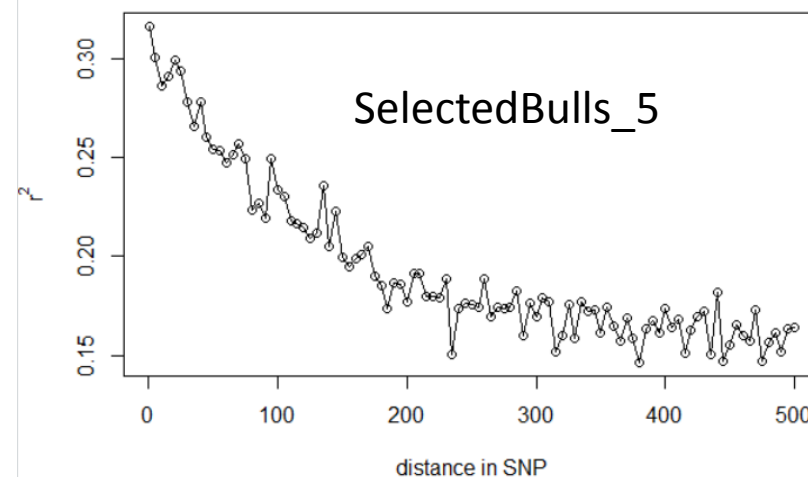
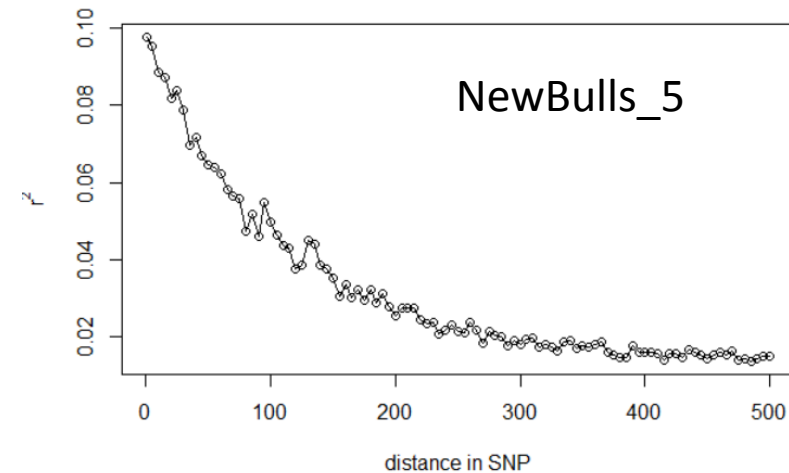
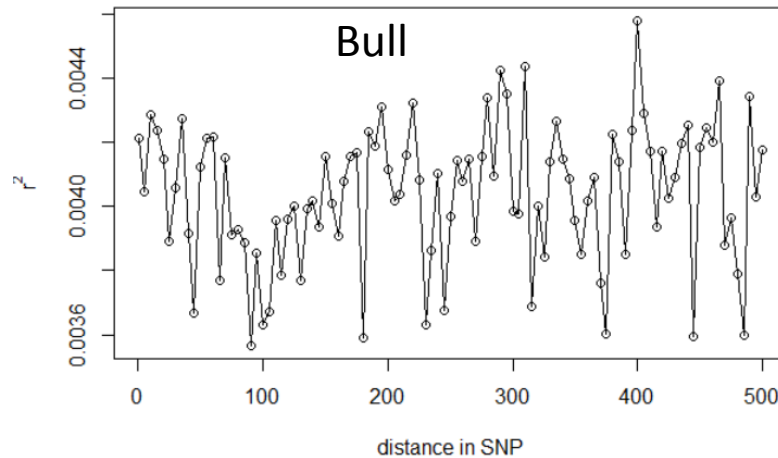
# Task IV - Solutions

- Simulations contain random factors therefore results can deviate slightly!





# Task IV - Solutions



- By far highest LD for the group of selected individuals