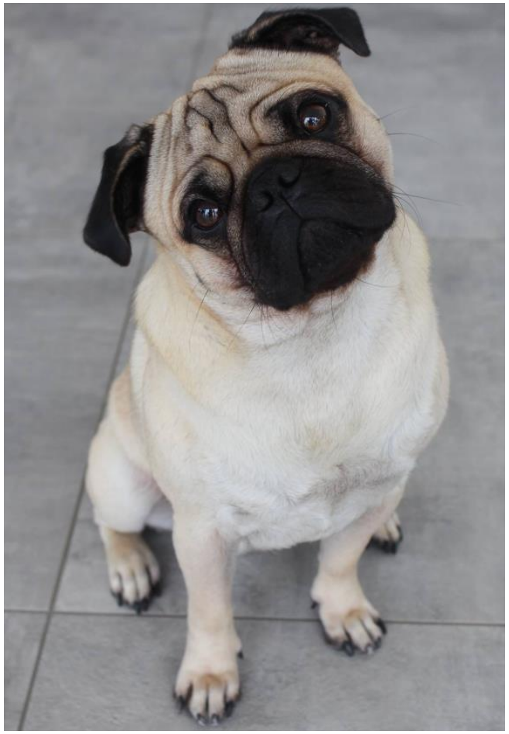
CiBreed
Center for Integrated Breeding Research



MoBPS – Modular Breeding Program Simulation

Torsten Pook[1], Martin Schlather[2], Amudha Ganesan[1], Henner Simianer[1]

[1] University of Goettingen, Animal Breeding and Genetics Group, Center for Integrated Breeding Research, Goettingen, Germany

[2] University of Mannheim, School of Business Informatics and Mathematics, Mannheim, Germany

Common question when designing breeding programs

- How many animals to use?
- Generate genotype/phenotype data of all animals?
- Mating scheme?
- Selection technique?
- And much more …

Possible ways to answer this?

- Experience of the breeder
- Simulation
- Problems with simulation studies & available tools:
  - Reality is far more complex
  - Too little flexibility to account for specific needs
    - ➔ Infinite number of homemade simulation tools/script

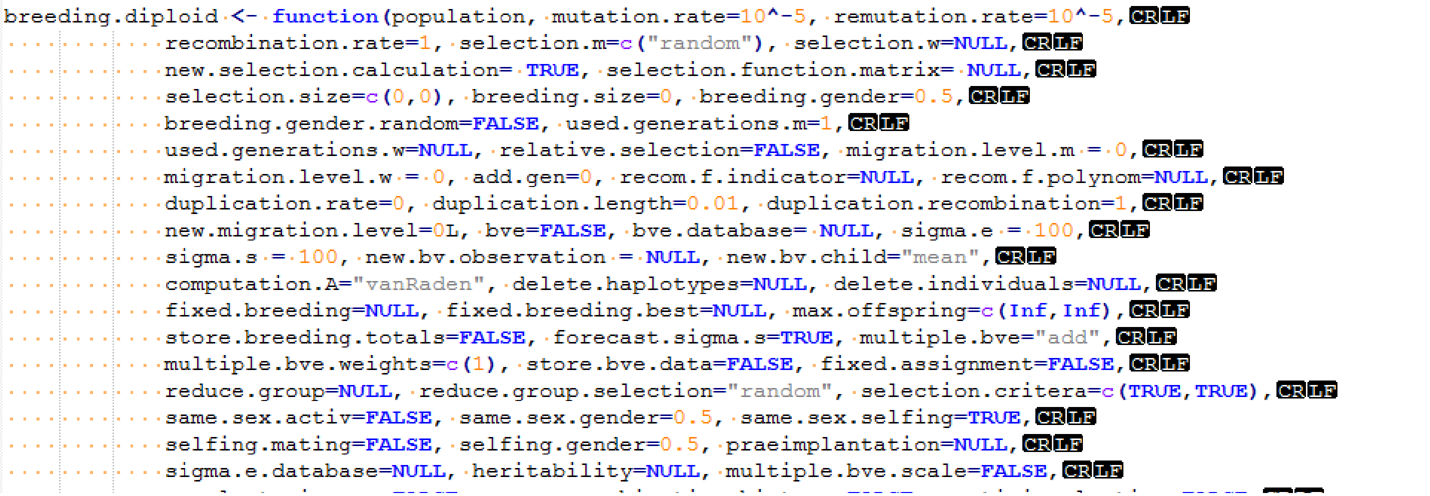
„One ~~ring~~ tool to ~~rule~~ handle them all“

R-package: MoBPS (RekomBre)

- Simulation based on single individuals
- Extremely flexible structure:
  - Whenever someone needed something new we could add it to the tool so far
- Storage technique:
  - General information
    - Allelic variants / Genetic maps
    - Traits and causal loci
  - Data for each individual
    - Points of recombination & founders for each segment
    - Mutations/Duplications
    - On-the-fly computation of genotypes & haplotypes
    - Pedigree

Ways to analyze the obtained data

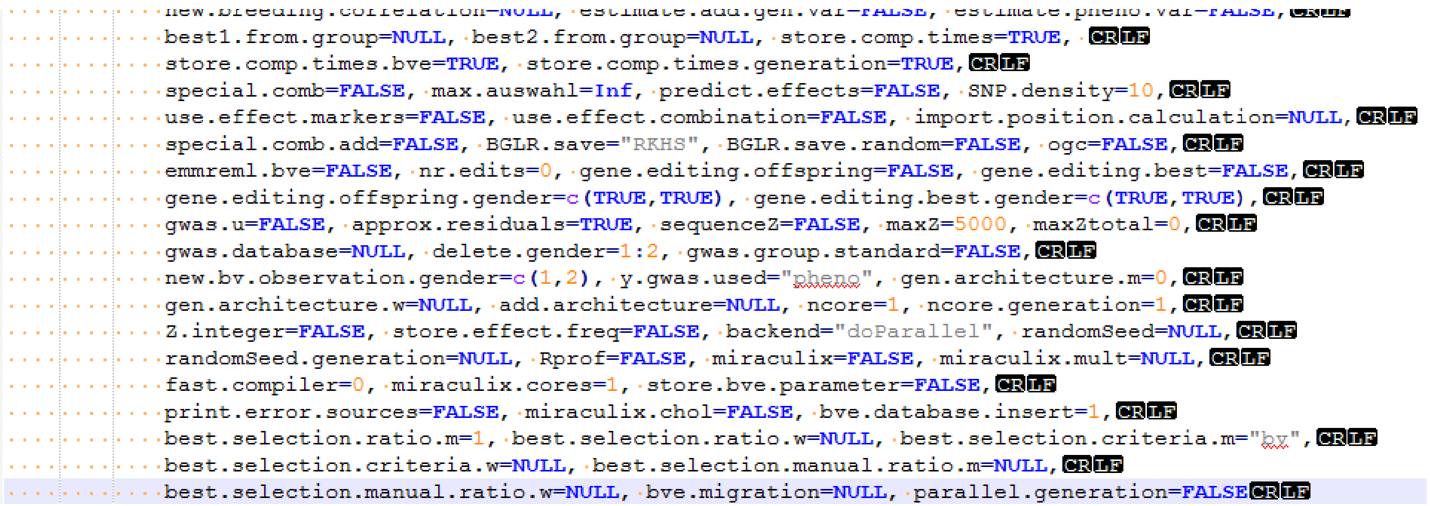
- Compute the costs and gains of a breeding program
  - Fixed costs
  - Cost of genotyping/phenotyping
  - Gains for each individual based on its trait values
- Genotypic analysis:
  - Perform analysis of a dataset with known underlying true settings
  - Genetic values, IBD, sweeps-detection
- Predefined output functions: Genetic trend, allelic frequencies



```
breeding.diploid <- function(population, mutation.rate=10^-5, remutation.rate=10^-5,
            recombination.rate=1, selection.m=c("random"), selection.w=NULL,
            new.selection.calculation= TRUE, selection.function.matrix= NULL,
            selection.size=c(0,0), breeding.size=0, breeding.gender=0.5,
            breeding.gender.random=FALSE, used.generations.m=1,
            used.generations.w=NULL, relative.selection=FALSE, migration.level.m = 0,
            migration.level.w = 0, add.gen=0, recom.f.indicator=NULL, recom.f.polynom=NULL,
            duplication.rate=0, duplication.length=0.01, duplication.recombination=1,
            new.migration.level=0L, bve=FALSE, bve.database= NULL, sigma.e = 100,
            sigma.s = 100, new.bv.observation = NULL, new.bv.child="mean",
            computation.A="vanRaden", delete.haplotypes=NULL, delete.individuals=NULL,
            fixed.breeding=NULL, fixed.breeding.best=NULL, max.offspring=c(Inf,Inf),
            store.breeding.totals=FALSE, forecast.sigma.s=TRUE, multiple.bve="add",
            multiple.bve.weights=c(1), store.bve.data=FALSE, fixed.assignment=FALSE,
            reduce.group=NULL, reduce.group.selection="random", selection.critera=c(TRUE,TRUE),
            same.sex.activ=FALSE, same.sex.gender=0.5, same.sex.selfing=TRUE,
            selfing.mating=FALSE, selfing.gender=0.5, praeimplantation=NULL,
            sigma.e.database=NULL, heritability=NULL, multiple.bve.scale=FALSE,
```

Only two functions are needed to perform all simulations.

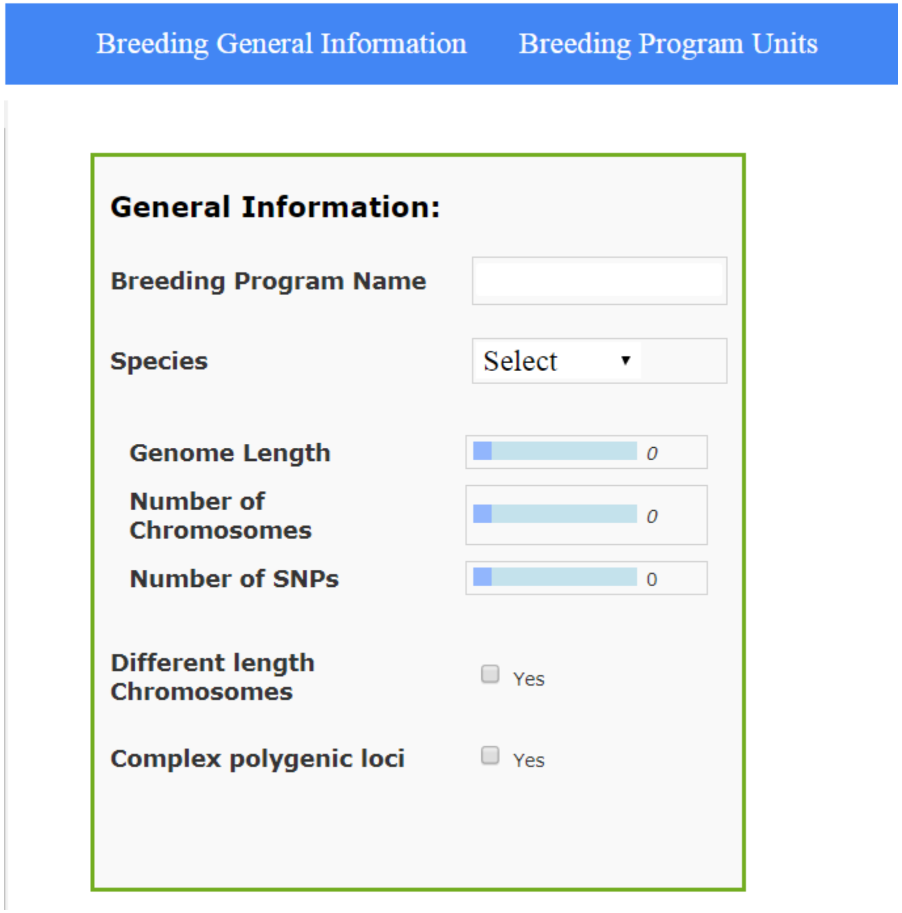
You just have to learn 200 input parameters as input options.



```
            new.breeding.correlation=NULL, estimate.add.gen.var=FALSE, estimate.pheno.var=FALSE,
            best1.from.group=NULL, best2.from.group=NULL, store.comp.times=TRUE, 
            store.comp.times.bve=TRUE, store.comp.times.generation=TRUE,
            special.comb=FALSE, max.auswahl=Inf, predict.effects=FALSE, SNP.density=10,
            use.effect.markers=FALSE, use.effect.combination=FALSE, import.position.calculation=NULL,
            special.comb.add=FALSE, BGLR.save="RKHS", BGLR.save.random=FALSE, ogc=FALSE,
            emmreml.bve=FALSE, nr.edits=0, gene.editing.offspring=FALSE, gene.editing.best=FALSE,
            gene.editing.offspring.gender=c(TRUE,TRUE), gene.editing.best.gender=c(TRUE,TRUE),
            gwas.u=FALSE, approx.residuals=TRUE, sequenceZ=FALSE, maxZ=5000, maxZtotal=0,
            gwas.database=NULL, delete.gender=1:2, gwas.group.standard=FALSE,
            new.bv.observation.gender=c(1,2), y.gwas.used="pheno", gen.architecture.m=0,
            gen.architecture.w=NULL, add.architecture=NULL, ncore=1, ncore.generation=1,
            Z.integer=FALSE, store.effect.freq=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,
            print.error.sources=FALSE, miraculix.chol=FALSE, bve.database.insert=1,
            best.selection.ratio.m=1, best.selection.ratio.w=NULL, best.selection.criteria.m="bv",
            best.selection.criteria.w=NULL, best.selection.manual.ratio.m=NULL,
            best.selection.manual.ratio.w=NULL, bve.migration=NULL, parallel.generation=FALSE
```

User interface

- Mainly developed by Amudha Ganesan
- Plan: Host a webserver (website) for anyone to access
- Computations itself have to be performed locally (PC or server)

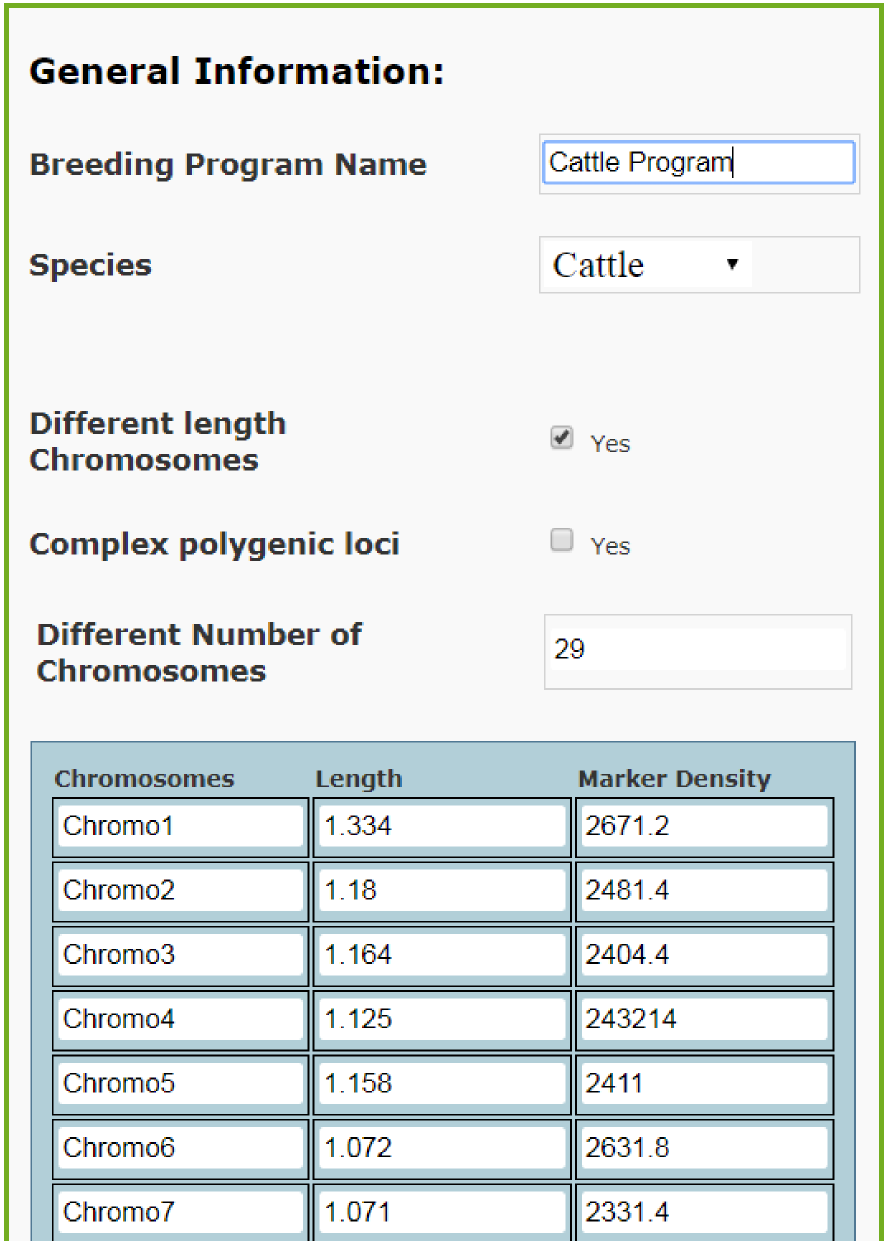


Breeding General Information   Breeding Program Units

General Information:

Breeding Program Name	
Species	Select ▾
Genome Length	0
Number of Chromosomes	0
Number of SNPs	0
Different length Chromosomes	□ Yes
Complex polygenic loci	□ Yes

Breeding General Information Breeding Program Units



General Information:

Breeding Program Name: Cattle Program

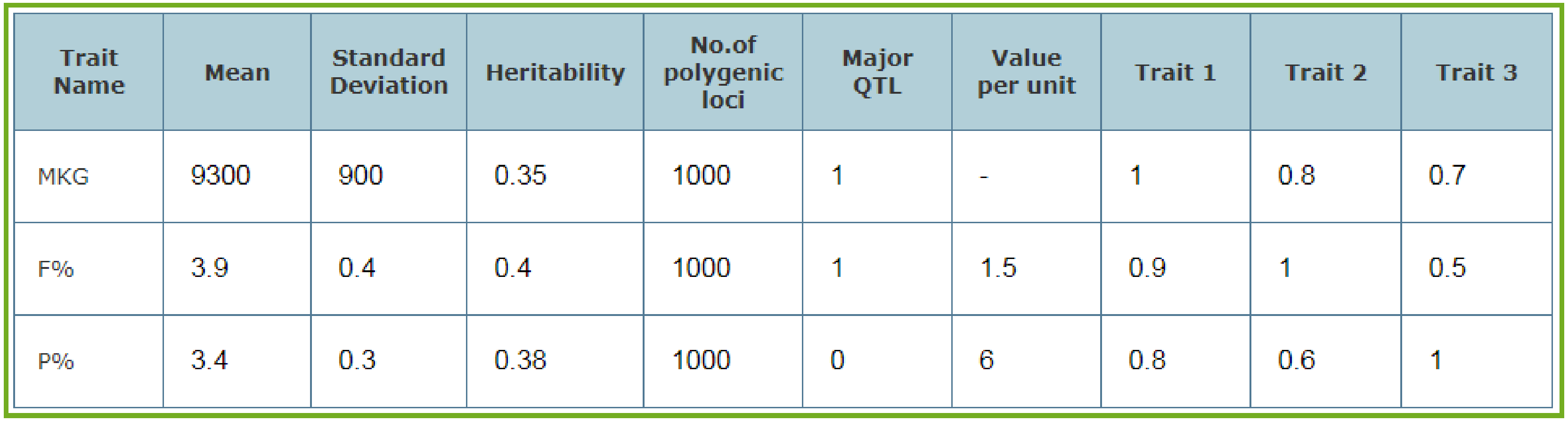
Species: Cattle ▾

Different length Chromosomes: ☑ Yes

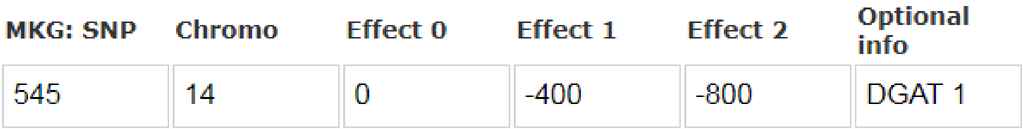
Complex polygenic loci: □ Yes

Different Number of Chromosomes: 29

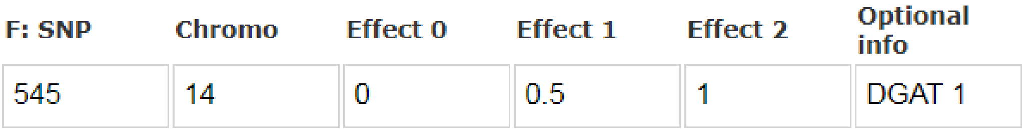
Chromosomes	Length	Marker Density
Chromo1	1.334	2671.2
Chromo2	1.18	2481.4
Chromo3	1.164	2404.4
Chromo4	1.125	243214
Chromo5	1.158	2411
Chromo6	1.072	2631.8
Chromo7	1.071	2331.4



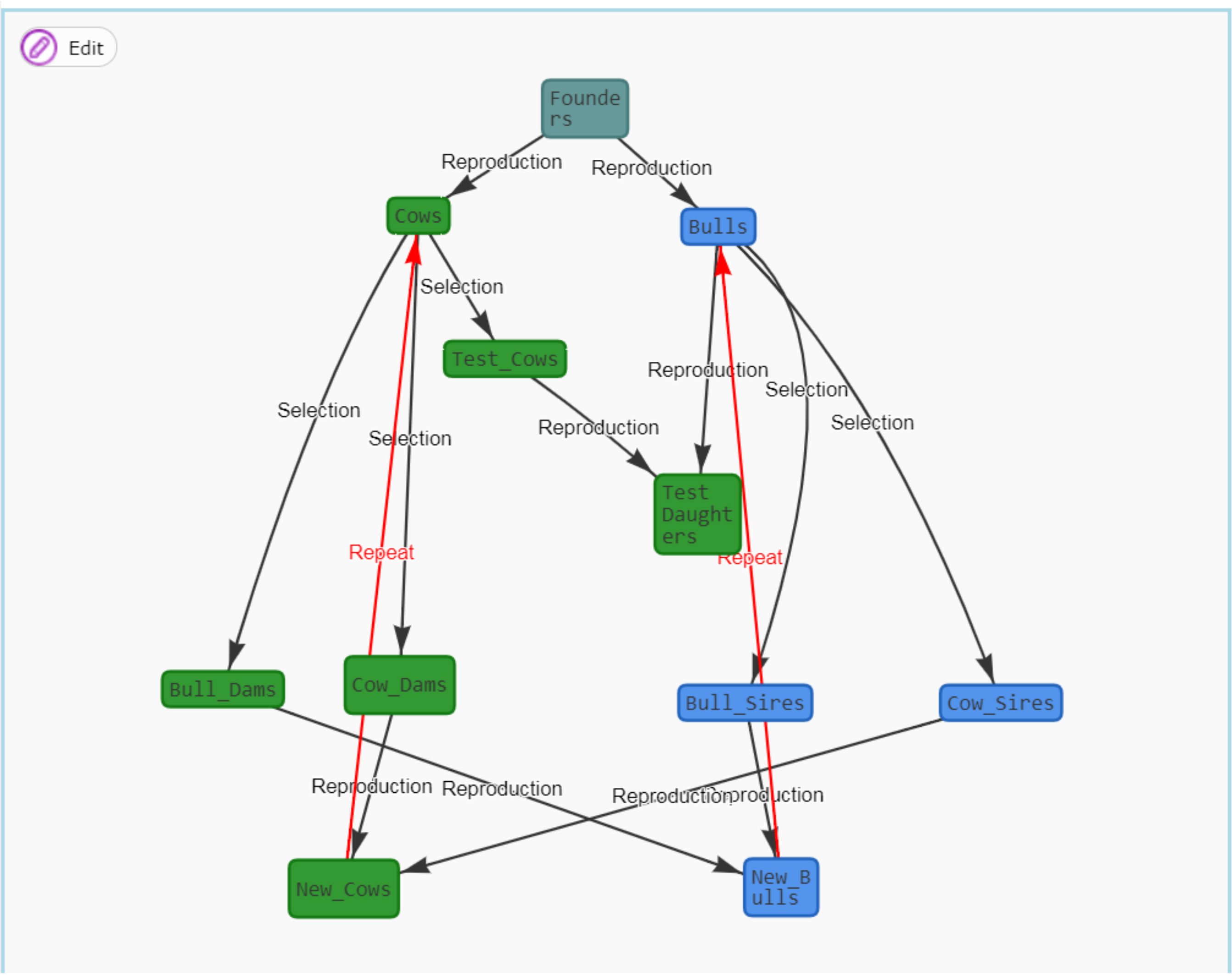
Trait Name	Mean	Standard Deviation	Heritability	No.of polygenic loci	Major QTL	Value per unit	Trait 1	Trait 2	Trait 3
MKG	9300	900	0.35	1000	1	-	1	0.8	0.7
F%	3.9	0.4	0.4	1000	1	1.5	0.9	1	0.5
P%	3.4	0.3	0.38	1000	0	6	0.8	0.6	1



MKG: SNP	Chromo	Effect 0	Effect 1	Effect 2	Optional info
545	14	0	-400	-800	DGAT 1

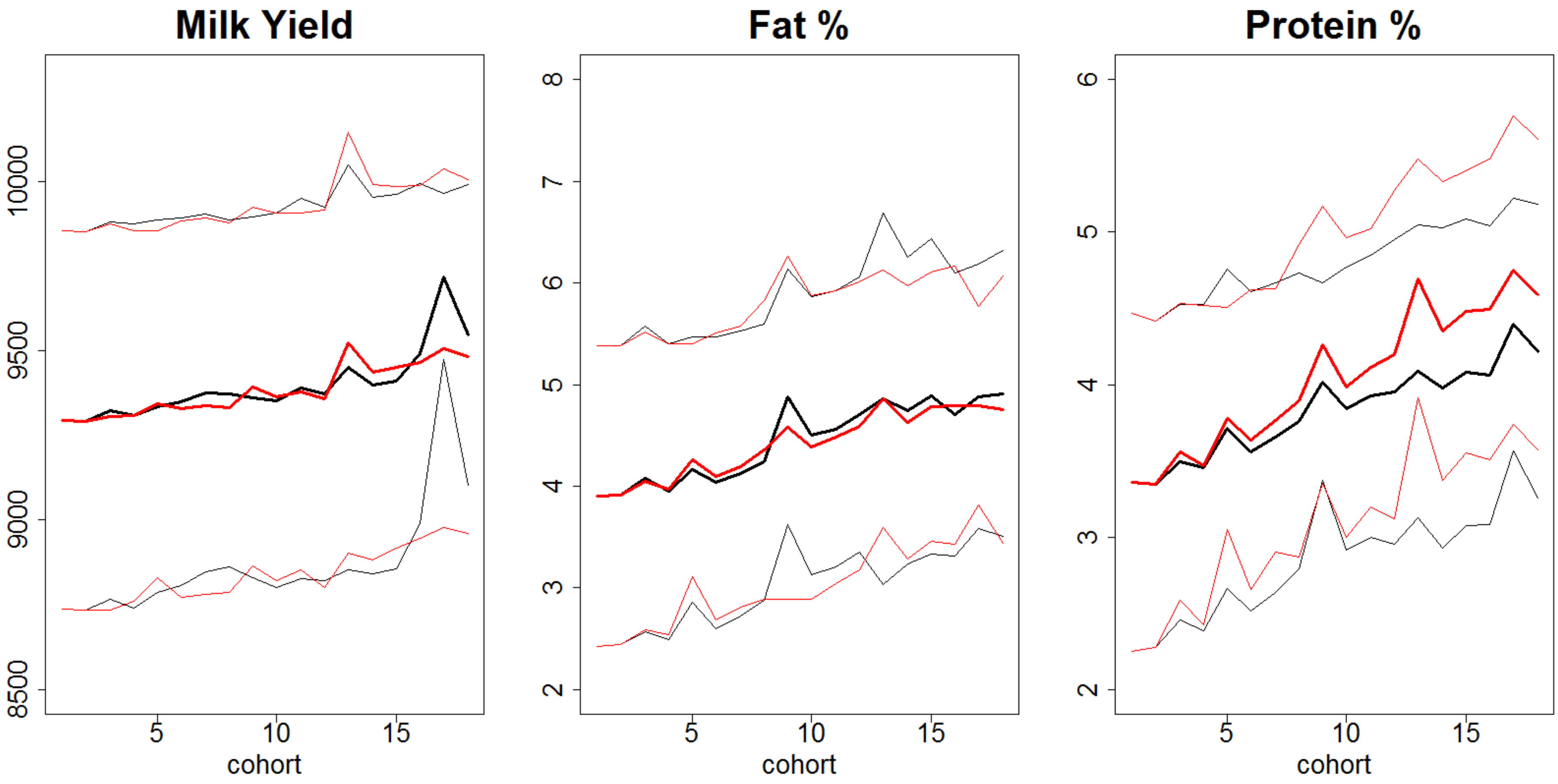


F: SNP	Chromo	Effect 0	Effect 1	Effect 2	Optional info
545	14	0	0.5	1	DGAT 1


Edit
Founders
Reproduction
Reproduction
Cows
Bulls
Selection
Test_Cows
Reproduction
Selection
Selection
Selection
Selection
Reproduction
Selection
Test Daughters
Repeat
Repeat
Bull_Dams
Cow_Dams
Bull_Sires
Cow_Sires
Reproduction
Reproduction
Reproduction
Reproduction
New_Cows
New_Bulls

Scenario-Comparison

- Equal Weights vs. 0.5 Milk Yield, 1 F%, 3 P%

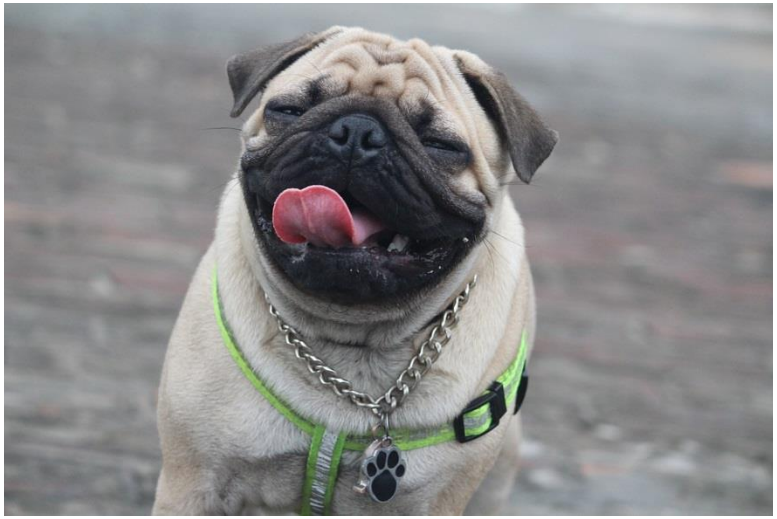


Memory & computational performance

- Most computational relevant parts are written in C/C++
- R-package miraculix developed by Martin Schlather
- Bitwise-storing of founder haplotypes:
  - Each marker needs 2 bits per individual (00, 01, 10, 11)
  - Traditional R: integer (32 bits) & numeric (64 bits)
- Bitwise matrix & vector multiplications (scalar products)
  - Bitwise operations on whole register (128/256 bit)
  - SSE2/AVX2 shuffle
- What it comes down to:
  - Same speed as PLINK with forth of memory usage
  - 10 times faster than regular matrix multiplication in R
  - 15 times less memory than

Summary

- New simulation tool: MoBPS
  - Breeding programs on an individual base
  - Flexibility to incorperate personal needs
  - Computational efficiency to simulate thousands of generations
- Soon openly available
  - R-package at https://github.com/tpook92/
  - Web-based application
- Beta-version on request:
  - Torsten.pook@uni-goettingen.de
    or talk to me here



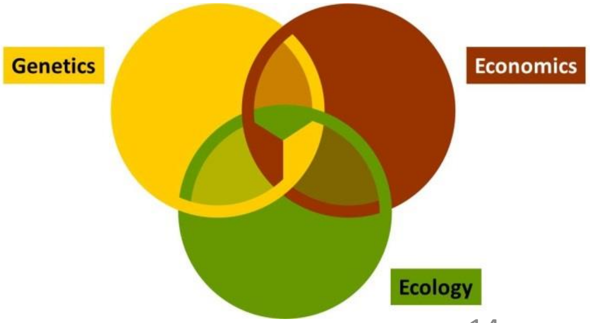
Acknowledgements

- My Project: MAZE („Accessing the genomic and functional diversity of maize to improve quantitative traits“)
  - Funding: BMBF (Grant ID 031B0195)
- The research leading to these results has received funding from the European Union‘s Horizon 2020 Research and Innovation Program under the grant agreement n° 677353 IMAGE
- Research Training Group 1644 „Scaling Problems in Statistics“
- Animal Breeding and Genetics Group University of Goettingen

IMAGE
Innovative Management of
Animal Genetic Resources

European Maize

Bundesministerium
für Bildung
und Forschung



Simulation of genetic traits

- For each individual store:
  - True underlying genetic values (not known in practice)
  - Phenotype (add some non-genetic variance)
  - Estimated breeding value (based on GBLUP or similar)
- Effects caused by: Single Marker, Two Markers, Networks, No direct QTL
- For multiple traits:
  - Assign effects to linked markers to obtain correlations
- Without underlying QTL:
  - Trait correlation by usage of formulas according to multidimensional Gaussian distributions:

$$X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{22} & \Sigma_{22} \end{pmatrix}\right)$$

Then:

$$X_1 | X_2 \sim N\left(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(X_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right)$$

Some simulation we have performed

- Gen editing for quantitative traits
  - 20 generations á 50.000 cows including GBLUP & GWAS/rrBLUP
- Simulation of selection sweeps
  - 5.000 generations under different selection intensities
- Comparison of IBD & BVE methods
- Targeted Mating to use breeding in scenarios with mainly epistatic interactions
- Mating strategies to improve breeding for single QTL traits (blue egg shell)