IGOR FERREIRA COELHO

OPTIMIZING A SWEET CORN BREEDING PROGRAM: IMPLEMENTING GENOMIC SELECTION AND DOUBLED HAPLOID TECHNOLOGY

Thesis submitted to the Genetics and Breeding Graduate Program of the Universidade Federal de Viçosa in partial fulfillment of the requirements for the degree of *Doctor Scientiae*.

Adviser: Leonardo Lopes Bhering

VIÇOSA - MINAS GERAIS
2022

Ficha catalográfica elaborada pela Biblioteca Central da Universidade Federal de Viçosa - Campus Viçosa

T
C672o
2022

Coelho, Igor Ferreira, 1992-
Optimizing a sweet corn breeding program: implementing genomic selection and doubled haploid technology / Igor Ferreira Coelho. – Viçosa, MG, 2022.
1 tese eletrônica (129 f.): il. (algumas color.).

Texto em inglês.
Orientador: Leonardo Lopes Bhering.
Tese (doutorado) - Universidade Federal de Viçosa, Departamento de Biologia Geral, 2022.
Referências bibliográficas: f. 42-49.
DOI: https://doi.org/10.47328/ufvbbt.2022.374
Modo de acesso: World Wide Web.

1. Milho- doce - Melhoramento genético. 2. Genética quantitativa. 3. Biometria. 4. Interação genótipo-ambiente. 5. Custo-benefício. I. Bhering, Leonardo Lopes, 1980-. II. Universidade Federal de Viçosa. Departamento de Biologia Geral. Programa de Pós-Graduação em Genética e Melhoramento. III. Título.

CDD 22. ed. 635.6722

Bibliotecário(a) responsável: Alice Regina Pinto CRB6 2523

IGOR FERREIRA COELHO

OPTIMIZING A SWEET CORN BREEDING PROGRAM: IMPLEMENTING GENOMIC SELECTION AND DOUBLED HAPLOID TECHNOLOGY

Thesis submitted to the Genetics and Breeding Graduate Program of the Universidade Federal de Viçosa in partial fulfillment of the requirements for the degree of *Doctor Scientiae*.

APPROVED: April 1st, 2022.

Assent:

Igor Ferreira Coelho
Author

Leonardo Lopes Bhering
Adviser

To my parents, Danilo Teixeira Coelho and Maria José Ferreira Coelho, and the unique support in all spheres of my life.

To Luana Matsuoka, my wife, future mother of our kids, and partner of this journey called life, supporting me every day.

DEDICATION

ACKNOWLEDGMENTS

Firstly, to God for guide me in all steps I make, to give me the opportunity to be born and reborn, the image of someone which guide me and watch for me is always in my heart.

To my parents, Danilo Teixeira Coelho and Maria José Ferreira Coelho, just you know the journey you had to give me the condition to be here today. I know how you sacrifice yourselves to provide the best for me. *"Obrigado por serem o que são, exemplo de pessoas boas e pais excepcionais."*

To my brother, Marconi Ferreira Coelho, and all my cousins which I consider each one as a sibling, always there with me, celebrating the achievements or helping me on the difficulties.

To my whole family, in the figures of my grandparents Terezinha e Ferreirinha, e Zizinha (*in memorian*) e Dadinho (*in memorian*). I always say that your generations were the key to make us, the grandchildren, full of opportunities.

To my wife, which was the "future" on the acknowledgments of my master, and today is a gift in my life. I hope you keep being this partner who pushes me up everyday and make me a better man and person. I want to add another "future" in here, and adjective you as the "future" mother of our kids. Thank you for this journey and everything.

To my advisor professor Leonardo Bhering, it has been over 6 years of friendship, and you helped me to grow up as a person and as a professional, thank you for the support and all the advices you gave through these years.

To my coadvisor which I spent the last year of PhD at his lab. It was a great opportunity to see how an international lab works, and how the companies are looking for the post graduate students. As I told you, I am glad to have met you and became your friend as well.

To the professors Flaviane de Oliveira Ribeiro, Kaio Olímpio das Graças Dias, Paulo Eduardo Teodoro, and, Larissa Pereira Ribeiro Teodoro, for being part of my commitee and your contribution to the improvement of this work.

To all my colleagues of the biometrics lab, which are friends for me, you made the work environment very good, and the partnerships on the projects made me grow up and learn a lot with you. In special to Marco Antonio Peixoto, a good friend and an example of dedication and effort, your life history is an inspiration, my friend. And all friends I made at the UFV maize breeding program, Sweet Corn Genomics and Breeding – UF, GenMelhor group, and, all the other colleagues and friends who I will never forget.

To my friends of Viçosa, which were always with me, trying to help me when necessary and celebrating my achievements.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

To the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), for granting the scholarship.

To the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), to granting the scholarship.

Thank you.

MUITO OBRIGADO!

The present work was financially supported by the Coordination for the Improvement of Higher Education Personnel (CAPES) - Finance Code 001.

“If you want to change the world, first try to improve and bring about change within yourself. That will help change your family. From there it just gets bigger and bigger. Everything we do has some effect, some impact.”

(Dalai Lama)

ABSTRACT

COELHO, Igor Ferreira, D.Sc., Universidade Federal de Viçosa, April, 2022. **Optimizing a Sweet Corn Breeding Program: implementing genomic selection and doubled haploid technology.** Adviser: Leonardo Lopes Bhering.

Several tools have been adopted to optimize the breeding programs performance, in terms of breeding cycle length and number of field plots. In this way, tools as genomic selection (GS) and doubled haploid technologies (DH) have been adopted because they shorten the breeding cycle, by predicting the best materials without needing for trying in field (GS) or by the generation of lines faster (DH); diminish the number of field plots, by bringing just the most potential materials (GS) or skipping successive cycles of autopollination (DH); and others. Moreover, many models and packages were developed to simulate breeding programs following the advancement of computational efficiency. With reliability of biological process and robust statistical principles. This fact enables the researchers to investigate the breeding methods and strategies, avoiding the need to implement everything in field, which would take long time and have high cost, to choose the most potential strategy(ies) to be adopted in the program. This work adopted the AlphaSimR package with the goal of optimize a sweet corn breeding program, by including the GS and DH tools, through the evaluation of the genetic parameters and general costs. It was observed that the adoption of these technologies inflates the budget of the program and increase the number of field plots. However, these strategies bring higher genetic gains of the programs and reduce the breeding cycle length. As conclusion, the financial/genetic recompense of adopting these technologies is given by the generation of lines/hybrids faster, which is an intangible gain, but it is very important in a long-term commercial breeding program.

Keywords: Plant Breeding. Quantitative Genetics. Biometric Analyses. Genotype-by-Environment Interaction. Cost Efficiency.

RESUMO

COELHO, Igor Ferreira, D.Sc., Universidade Federal de Viçosa, abril de 2022. **Otimização de um programa de melhoramento genético de milho doce: implementando seleção genômica e técnica de duplo-haploide.** Orientador: Leonardo Lopes Bhering.

Diversas ferramentas têm sido adotadas para otimização de programas de melhoramento, em termos de tempo de ciclo e quantidade de experimentos de campo. Nesse sentido, análises como seleção genômica (GS) e utilização de tecnologia duplo haplóide (DH) tem sido adotada em programas pelo fato de encurtarem o tempo do ciclo, seja por predição (GS) ou pela geração de linhagens mais rapidamente (DH); diminuir o número de plots em campo, predizendo os materiais potencias para serem testados (GS) ou evitando os sucessivos ciclos de auto-fecundação (DH); dentre outros. Além disso, com o avanço da eficiência computacional, diversos modelos e pacotes foram desenvolvidos no intuito de simularem programas de melhoramento. Esses modelos permitem a investigação de métodos visando a escolha dos estágios adequados para introdução de tecnologias, a fim de selecionar as melhores estratégias e não necessitar ter que implementar no programa para realmente ver sua potencialidade, perdendo tempo e dinheiro em plots teste. O trabalho desenvolvido utilizou o pacote AlphaSimR com o intuito de otimizar um programa de melhoramento de milho doce, com a adoção de GS e DH, por meio da avaliação de parâmetros genéticos, e custos gerais. Notou-se que as adoções das tecnologias supracitadas inflam o orçamento do programa convencional atualmente adotado e aumentam o número de parcelas em campo. No entanto, essas estratégias trazem ganhos genéticos e ainda reduzem o tempo de ciclo de melhoramento. Conclui-se que o retorno financeiro/genético da adoção de novas tecnologias ao programa conseguirá aumentar o número de materiais lançados e consequentemente a participação no mercado de milho doce do programa, o que são ganhos intangíveis que são peça chave em um programa de melhoramento comercial com uma visão de longo prazo.

Palavras-chave: Melhoramento de Plantas. Genética Quantitaiva. Analyses Biométricas. Interação Genótipo por Ambientes. Eficiência De Custos.

LIST OF ILLUSTRATIONS

LIST OF TABLES

SUMMARY

1 Introduction

Maize (*Zea mays L.*) is a grass in the Poaceae family. One of the most planted cereals in the world, with high versatility, being used for both, human and animal nutrition, in addition to having numerous by-products in the industries (Lima and Borém, 2018). As a result of several mutation in genes in the starch biosynthesis pathway, the sweet corn represents a variety of maize with higher sugar content in the endosperm while reducing starch content (Hu et al., 2021). It represents an important vegetal in EUA and Canada countries, and has the capability of growing all over the word (Lertrat and Pulam, 2007).

To date, just recently a detailed study involving the genome of sweet corn was available (Hu et al., 2021). Combining the importance of sweet corn with the needed to investigate the implementation of recent tools, such as the double haploid technology and genomic selection, simulation studies stand out as largely indicated to guide the implementation and optimization them in a breeding program.

According to Faux et al. (2016), simulation has the potential to evaluation and develop new strategies for plant breeding. In this way, new breeding strategies could be compared without the needed of the field implementation of all of them, which reduces the costs to optimize a breeding program. Just recently programs and packages has been developed to implement those inferences (Faux et al., 2016). Furthermore, many *in silico* models have been developed over the last years to simulate the dynamics of a breeding program, mimicking genomic and phenotypic data. Thus, the breeders can evaluate the best strategies based on different population structures and genetic parameters, leading to access the long-term genetic gains in different scenarios, by time and by traits.

Thus, the main objective of this study was to optimize the sweet corn breeding program based on simulated data. Aiming to be an applied and realistic study, it was considered the genetic parameters: genetic gain, genetic variance, accuracies, and costs of each distinct scenario (by considering the number of plots and applied technologies, molecular markers and doubled haploid technology implementation). Finally, we attempt to include a business perspective in the results, by considering not solely the genetic performance of each scenario in the long term, but also the implementation expenses of each one, whereas including product releasing time, which, ultimately, affects the market share of the company and its brand valuation.

2 Theoretical references

2.1 Simulation studies

Over the last century, quantitative genetics has developed and provided much of the knowledge and information regarding the design and analysis of selection methods applied into different breeding programs (Allard, 1971; Falconer and Mackay, 1996; Wang et al., 2003; Bernardo, 2020). The first computer simulation studies of genetic process were developed to investigate the likelihood of outcomes breeding strategies as well as evolutionary processes for known or hypothesized genetic models (Moore and Tonsor, 1994; Cox et al., 1995; Podlich and Cooper, 1998; van Berloo and Stam, 1998; Podlich et al., 1999). Cooper et al. (1999) highlighted the situation of the slow progress of theoretical framework for quantitative genetics comparing to the increasing amount of experimental information on the genetic architecture of traits at the time. Their study presented the computer simulation as an efficient way to deal with some of these issues associated with the conventional approach and be able to develop theoretical prediction equations in a shorter time.

The number of simulation studies increased after the 2000`s with the development of most powerful computers, regarding their memories and processing capacities. There are different tools which enable the breeder to investigate the target crop, as maize (Cowling et al., 2020); sorghum (Muleta et al., 2019); sugar kelp (Huang et al., 2021); or compare different crops (Marulanda et al., 2016; Ali et al., 2020) and even intercrop breeding programs (Bančič et al., 2021). All these comparisons and simulations can be developed due to different tools available to the breeders.

In 2016, Faux et al. (2016) developed a software called AlphaSim, which became a R package years later, named AlphaSimR (Gaynor et al., 2020). It is based on stochastic simulation and allow to implement a whole breeding program dynamic. The four main steps for a program simulation within the software are: (i) simulation of the founder population, with all the genomic information, ploidy, and number of chromosomes; (ii) the target traits, setting their variances, effects, and other important parameters; (iii) the breeding method and its practices, as random crosses, self, doubled haploid, and, (iv) measurement and comparison of the strategies' performances of the simulation. AlphaSimR also enables the user to apply selection based on the phenotypic data or the genomic data, using some genomic selection analyses too. There are many studies using this software/package to develop the populations and imputation analyses (Gonen et al., 2018), inbred development (Gaynor et al., 2017), and many crop breeding program designs (Muleta et al., 2019; Ali et al., 2020; Cowling et al., 2020; Huang et al., 2021).

2.2 Genomic Selection

The application of genomic tools to improve breeding programs has been increasing over the last decades (Bassi et al., 2016). The marker-assisted selection

(MAS), proposed by Tanksley et al. (1989), gained popularity among the breeders aiming to improve the breeding efficiency of their breeding programs.

With the cost decreasing of molecular markers (Bassi et al., 2016; Wetterstrand, 2020), and, due to MAS limitations (Dekkers and Hospital, 2002; Bernardo, 2008), the genomic selection (GS) emerged as a relevant tool (Meuwissen et al., 2001). Ever since, it has been adopted by many breeders with the intention of employing genome-wide marker data to make earlier selection and more accurate prediction, increasing the efficiency, genetic gain per unit time and decreasing the costs (Heffner et al., 2010; Bassi et al., 2016) of a target breeding program.

Briefly, the GS can predict, through different statistic models, the breeding value of individuals. The dataset must contain genomic and phenotypic information. The training set (or training population) needs both information to calibrate the statistic model. In this step, there will be different strategies of cross-validations to increase the predict ability of the model. Following, the best fit model will be applied to predict the breeding or genotypic values of the candidates with no phenotypic data records ("non-phenotyped"). These candidates are only genotyped and known as breeding population (Würschum et al., 2013).

Many studies have been developed to investigate more efficient ways to improve the genomic selection prediction ability, accuracy, and efficiency, as: (i) Statistical methods comparison (Heslot et al., 2012; Wang et al., 2015); (ii) Filtering and imputation investigation (Lin et al., 2010; Verma et al., 2014); (iii) Ideal training and validation population sizes and folds (Diana and Tommasi, 2002; Popovici et al., 2010; Boddhireddy et al., 2014); and others.

However, as mentioned by Verges & van Sanford (2020), one of the high-priority questions which stands is: "What is the optimal stage in the breeding process

at which GS should be implemented?". Some authors have already raised some studies in face of investigating the best alternatives to apply GS in the right stage (Heslot et al., 2015; Bassi et al., 2016; Gaynor et al., 2017). Further investigation can still be developed to include more evidence in this scarce literature area.

2.3 Doubled Haploid Technology

Since the beginning of the 19th century, the main maize breeder task was stablished. As proposed by Shull (1908), the maize breeder obligatory mission is to seek the best hybrid combination of parental materials, to originate seeds that could have better or, at least, same yield. After this statement, the improvement of parental inbred lines with a high homozygosity level (>95%), were developed through recurrent self-pollinations and selection (Hallauer et al., 2010), up to eight generations.

The doubled haploids (DH) studies were focus of many papers in the end of 19th century, and some questions arose about their using, efficiency, and application in many studies (Baenziger, 1996). Bouchez & Gallais, in 2000, came up with the idea of using *haplodiploidization*, as mentioned by them, (or doubled haploids) as an efficient way to induce complete homozygosity in a short period. This would be very interesting to any breeding program, allowing a quick hybrid development, as they also predicted.

In fact, after many uncertainties, the DH's technology has established as a very potential method for inbred development due to the shortening of time. Nowadays, the process basically samples the segregates gametes from the target germplasm (biparental crosses or breeding population), and produces a completely homozygous line with one fewer steps, as cited by Chaikam et al. (2019): haploid induction: i) by using a "haploid inducer" plant, which generates segregation within an ear, of diploid (2n) and certain fraction of haploid (n) due to anomalous fertilization; ii) haploid identification: after the induction, the haploid kernel is identified by a dominant

anthocyanin color marker (*R1-Navajo, R1-nj*); iii) chromosome doubling: the using of colchicine hormone is very common as a doubling agent (or mitotic inhibitor) in DH production, however, because of its high carcinogenic risk, other herbicides have been investigated as mitotic inhibitors; iv) agronomic management of the DH seeds: from the seedling in the greenhouse, through all the cycle, and the development in field too, the agronomic managements have to been well defined and made by the researchers, as irrigation, fertilizers, mechanic operation, and management of seeds, diseases and insects; and, vi) DH seed multiplication: this point is very relevant for some programs which do not produce the DH by themselves, and have to multiply the number of seeds after the DH production, aiming to guarantee the seeds for next cycles and crosses.

Currently, the majority haploidy induction is made in the F_1 generation materials (Melchinger et al., 2005; Smith et al., 2008). Also, some studies have compared the application of DHs, and how it affects the breeding program development (e.g. Gaynor et al., 2017) . The literature is very scarce about the using DH in sweet corn breeding program.

3 References

Akdemir, D., Sanchez, J. I., and Jannink, J. (2015). Optimization of genomic selection training populations with a genetic algorithm. *Genet. Sel. Evol.* 47, 1–10. doi: 10.1186/s12711-015-0116-6.

Ali, M., Zhang, L., DeLacy, I., Arief, V., Dieters, M., Pfeiffer, W. H., et al. (2020). Modeling and simulation of recurrent phenotypic and genomic selections in plant breeding under the presence of epistasis. *Crop J.* 8, 866–877. doi: 10.1016/j.cj.2020.04.002.

Allard, R. W. (1971). *Principles of plant breeding*. São Paulo - SP: Edgard Blücher.

Andorf, C., Beavis, W. D., Hufford, M., Smith, S., Suza, W. P., Wang, K., et al. (2019). Technological advances in maize breeding: past, present and future. *Theor. Appl. Genet.* 132, 817–849. doi: 10.1007/s00122-019-03306-3.

Baenziger, P. S. (1996). "Reflections on doubled haploids in plant breeding," in *In Vitro Haploid Production in Higher Plants*, 35–48. doi: 10.1007/978-94-017-1860-8_3.

Bančič, J., Werner, C. R., Gaynor, R. C., Gorjanc, G., Odeny, D. A., Ojulong, H. F., et al. (2021). Modeling Illustrates That Genomic Selection Provides New Opportunities for Intercrop Breeding. *Front. Plant Sci.* 12. doi: 10.3389/fpls.2021.605172.

Barnabás, B., Obert, B., and Kovács, G. (1999). Colchicine, an efficient genome-doubling agent for maize (Zea mays L.) microspores cultured in anthero. *Plant Cell Rep.* 18, 858–862. doi: 10.1007/s002990050674.

Bassi, F. M., Bentley, A. R., Charmet, G., Ortiz, R., and Crossa, J. (2016). Breeding schemes for the implementation of genomic selection in wheat (Triticum spp .). *Plant Sci.* 242, 23–36. doi: 10.1016/j.plantsci.2015.08.021.

Batista, L. G., Gaynor, R. C., Margarido, G. R. A., Byrne, T., Amer, P., Gorjanc, G., et al. (2021). Long-term comparison between index selection and optimal independent culling in plant breeding programs with genomic prediction. *PLoS One* 16, 1–5. doi: 10.1371/journal.pone.0235554.

Bernardo, R. (2008). Molecular markers and selection for complex traits in plants: Learning from the last 20 years. *Crop Sci.* 48, 1649–1664. doi: 10.2135/cropsci2008.03.0131.

Bernardo, R. (2020). Reinventing quantitative genetics for plant breeding: something old, something new, something borrowed, something BLUE. *Heredity (Edinb).* 125, 375–385. doi: 10.1038/s41437-020-0312-1.

Boddhireddy, P., Kelly, M. J., Northcutt, S., Prayaga, K. C., Rumph, J., and DeNise, S. (2014). Genomic predictions in Angus cattle: Comparisons of sample size, response variables, and clustering methods for cross-validation1. *J. Anim. Sci.* 92, 485–497. doi: 10.2527/jas.2013-6757.

Boerman, N. A., Frei, U. K., and Lübberstedt, T. (2020). Impact of Spontaneous Haploid Genome Doubling in Maize Breeding. *Plants* 9, 369. doi: 10.3390/plants9030369.

Bouchez, A., and Gallais, A. (2000). Efficiency of the Use of Doubled-Haploids in Recurrent Selection for Combining Ability. *Crop Sci.* 40, 23–29.

Broman, K. W., and Weber, J. L. (2000). Characterization of Human Crossover Interference. *Am. J. Hum. Genet.* 66, 1911–1926. doi: 10.1086/302923.

Chaikam, V., Molenaar, W., Melchinger, A. E., and Boddupalli, P. M. (2019). Doubled haploid technology for line development in maize: technical advances and prospects. *Theor. Appl. Genet.* 132, 3227–3243. doi: 10.1007/s00122-019-03433-x.

Chase, S. S. (1949). MONOPLOID FREQUENCIES IN A COMMERCIAL DOUBLE CROSS HYBRID MAIZE, AND IN ITS COMPONENT SINGLE CROSS HYBRIDS AND INBRED LINES. *Genetics* 34, 328–332. doi: 10.1093/genetics/34.3.328.

Chase, S. S. (1952). Production of Homozygous Diploids of Maize from Monoploids 1. *Agron. J.* 44, 263–267. doi: 10.2134/agronj1952.00021962004400050010x.

Chen, G. K., Marjoram, P., and Wall, J. D. (2009). Fast and flexible simulation of DNA sequence data. *Genome Res.* 19, 136–142. doi: 10.1101/gr.083634.108.

Coe, E. H., and Sarkar, K. R. (1964). The detection of haploids in maize. *J. Hered.* 55, 231–233. doi: 10.1093/oxfordjournals.jhered.a107340.

Cooper, M., Podlich, D. W., Jensen, N. M., Chapman, S. C., and Hammer, G. L. (1999). Modeling Plant Breeding Programs. *Trends Agron.* 2, 33–64.

Cowling, W. A., Gaynor, R. C., Antolín, R., Gorjanc, G., Edwards, S. M., Powell, O., et al. (2020). In silico simulation of future hybrid performance to evaluate heterotic pool formation in a self-pollinating crop. *Sci. Rep.* 10, 1–8. doi: 10.1038/s41598-020-61031-0.

Cox, T. S., Sears, R. G., Bequette, R. K., and Martin, T. J. (1995). Germplasm enhancement in winter wheat x Triticum tauschii backcross populations. *Crop Sci.* 35, 913–919. doi: 10.2135/cropsci1995.0011183X003500030047x.

Crossa, J., Burgueño, J., Cornelius, P. L., McLaren, G., Trethowan, R., and Krishnamachari, A. (2006). Modeling Genotype × Environment Interaction Using Additive Genetic Covariances of Relatives for Predicting Breeding Values of Wheat Genotypes. *Crop Sci.* 46, 1722–1733. doi: 10.2135/cropsci2005.11-0427.

Crossa, J., Pérez-Rodríguez, P., Cuevas, J., Montesinos-López, O., Jarquín, D., de los Campos, G., et al. (2017). Genomic Selection in Plant Breeding: Methods, Models, and Perspectives. *Trends Plant Sci.* 22, 961–975. doi: 10.1016/j.tplants.2017.08.011.

Dekkers, J. C. M., and Hospital, F. (2002). The use of molecular genetics in the improvement of agricultural populations. *Nat. Rev. Genet.* 3, 22–32. doi: 10.1038/nrg701.

Diana, G., and Tommasi, C. (2002). Cross-validation methods in principal component analysis: A comparison. *Stat. Methods Appt.* 11, 71–82. doi: 10.1007/BF02511446.

Eder, J., and Chalyk, S. (2002). In vivo haploid induction in maize. *Theor. Appl. Genet.* 104, 703–708. doi: 10.1007/s00122-001-0773-4.

Endelman, J. B. (2011). Ridge regression and other kernels for genomic selection with R package rrBLUP. *Plant Genome* 4, 250–255.

Falconer, D. S., and Mackay, T. F. C. (1996). Introduction to quantitative genetics. *Harlow, Essex, UK Longmans Green* 3, 280.

Faux, A., Gorjanc, G., Gaynor, R. C., Battagin, M., Edwards, S. M., Wilson, D. L., et al. (2016). AlphaSim: Software for Breeding Program Simulation. *Plant Genome* 9, plantgenome2016-02. doi: 10.3835/plantgenome2016.02.0013.

Gaynor, R. C., Gorjanc, G., Bentley, A. R., Ober, E. S., Howell, P., Jackson, R., et al. (2017). A two-part strategy for using genomic selection to develop inbred lines. *Crop Sci.* 57, 2372–2386. doi: 10.2135/cropsci2016.09.0742.

Gaynor, R. C., Gorjanc, G., and Hickey, J. (2020). AlphaSimR: An R-package for Breeding Program Simulations. *G3 Genes|Genomes|Genetics*, 1–21. doi: 10.1101/2020.08.10.245167.

Geiger, H. H., and Gordillo, G. A. (2009). Doubled haploids in hybrid maize breeding. *Maydica* 54, 485–499.

Gonen, S., Wimmer, V., Gaynor, R. C., Byrne, E., Gorjanc, G., and Hickey, J. M. (2018). A heuristic method for fast and accurate phasing and imputation of single-nucleotide polymorphism data in bi-parental plant populations. *Theor. Appl. Genet.* 131, 2345–2357. doi: 10.1007/s00122-018-3156-9.

Gorjanc, G., Jenko, J., Hearne, S. J., and Hickey, J. M. (2016). Initiating maize pre-breeding programs using genomic selection to harness polygenic variation from landrace populations. *BMC Genomics* 17, 1–15. doi: 10.1186/s12864-015-2345-z.

Hallauer, A. R., Carena, J. M., and Miranda Filho, J. B. de. (2010). *Quantitative genetics in maize breeding. New York: Springer, 2010. 500 p.* New York: Springer.

Hallauer, A. R., and Darrah, L. L. (1985). Compendium of recurrent selection methods and their application. *CRC. Crit. Rev. Plant Sci.* 3, 1–33. doi: 10.1080/07352688509382202.

Häntzschel, K. R., and Weber, G. (2010). Blockage of mitosis in maize root tips using colchicine-alternatives. *Protoplasma* 241, 99–104. doi: 10.1007/s00709-009-0103-2.

Heffner, E. L., Jannink, J.-L., and Sorrells, M. E. (2011). Genomic Selection Accuracy using Multifamily Prediction Models in a Wheat Breeding Program. *Plant Genome* 4, 65–75. doi: 10.3835/plantgenome2010.12.0029.

Heffner, E. L., Lorenz, A. J., Jannink, J. L., and Sorrells, M. E. (2010). Plant breeding with Genomic selection: Gain per unit time and cost. *Crop Sci. Madison* 50, 1681–1690. doi: 10.2135/cropsci2009.11.0662.

Heffner, E. L., Sorrells, M. E., and Jannink, J. L. (2009). Genomic selection for crop improvement. *Crop Sci.* 49, 1–12. doi: 10.2135/cropsci2008.08.0512.

Heslot, N., Jannink, J.-L., and Sorrells, M. E. (2015). Perspectives for Genomic Selection Applications and Research in Plants. *Crop Sci.* 55, 1–12. doi: 10.2135/cropsci2014.03.0249.

Heslot, N., Yang, H. P., Sorrells, M. E., and Jannink, J. L. (2012). Genomic selection in plant breeding: A comparison of models. *Crop Sci.* 52, 146–160. doi: 10.2135/cropsci2011.06.0297.

Hickey, J. M., Dreisigacker, S., Crossa, J., Hearne, S., Babu, R., Prasanna, B. M., et al. (2014). Evaluation of genomic selection training population designs and genotyping strategies in plant breeding programs using simulation. *Crop Sci.* 54, 1476–1488. doi: 10.2135/cropsci2013.03.0195.

Hu, Y., Colantonio, V., Müller, B. S. F., Leach, K. A., Nanni, A., Finegan, C., et al. (2021). Genome assembly and population genomic analysis provide insights into the evolution of modern sweet corn. *Nat. Commun.* 12. doi: 10.1038/s41467-021-21380-4.

Huang, M., Robbins, K. R., Li, Y., Umanzor, S., Marty-Rivera, M., Bailey, D., et al. (2021). Simulation of sugar kelp (Saccharina latissima) breeding guided by practices to prioritize accelerated research gains. *Sci. South Russ.*, 39–44. doi: 10.1101/2021.01.21.427651.

Isidro, J., Jannink, J.-L., Akdemir, D., Poland, J., Heslot, N., and Sorrells, M. E. (2015). Training set optimization under population structure in genomic selection. *Theor. Appl. Genet.* 128, 145–158. doi: 10.1007/s00122-014-2418-4.

Jannink, J.-L., Lorenz, A. J., and Iwata, H. (2010). Genomic selection in plant breeding: from theory to practice. *Brief. Funct. Genomics* 9, 166–177. doi: 10.1093/bfgp/elq001.

Jarquín, D., Pérez-Elizalde, S., Burgueño, J., and Crossa, J. (2016). A Hierarchical Bayesian Estimation Model for Multienvironment Plant Breeding Trials in Successive Years. *Crop Sci.* 56, 2260–2276. doi: 10.2135/cropsci2015.08.0475.

Kato, A. (2002). Chromosome doubling of haploid maize seedlings using nitrous oxide gas at the flower primordial stage. *Plant Breed.* 121, 370–377. doi: 10.1046/j.1439-0523.2002.743321.x.

Kelliher, T., Starr, D., Richbourg, L., Chintamanani, S., Delzer, B., Nuccio, M. L., et al. (2017). MATRILINEAL, a sperm-specific phospholipase, triggers maize haploid induction. *Nature* 542, 105–109. doi: 10.1038/nature20827.

Labroo, M. R., and Rutkoski, J. E. (2021). New cycle , same old mistakes ? Overlapping vs . discrete generations in long-term recurrent selection. doi: 10.1101/2021.10.12.464059.

Lertrat, K., and Pulam, T. (2007). Breeding for increased sweetness in sweet corn. *Int. J. Plant Breed.* 1, 27–30.

Lima, R. O. de, and Borém, A. (2018). *Melhoramento de Milho*. Viçosa - MG: Editora UFV.

Lin, P., Hartz, S. M., Zhang, Z., Saccone, S. F., Wang, J., Tischfield, J. A., et al. (2010). A New Statistic to Evaluate Imputation Reliability. *PLoS One* 5, e9697. doi: 10.1371/journal.pone.0009697.

Liu, C., Li, X., Meng, D., Zhong, Y., Chen, C., Dong, X., et al. (2017). A 4-bp Insertion at ZmPLA1 Encoding a Putative Phospholipase A Generates Haploid Induction in Maize. *Mol. Plant* 10, 520–522. doi: 10.1016/j.molp.2017.01.011.

Liu, Z., Ren, J., Trampe, B., Frei, U. K., and Lubberstedt, T. (2016). "Double haploid: from obscure phenomenon to key technology of current maize breeding programs," in *Plant Breeding Reviews, Vol 40*, 123–166.

Marulanda, J. J., Mi, X., Melchinger, A. E., Xu, J. L., Würschum, T., and Longin, C. F. H. (2016). Optimum breeding strategies using genomic selection for hybrid

breeding in wheat, maize, rye, barley, rice and triticale. *Theor. Appl. Genet.* 129, 1901–1913. doi: 10.1007/s00122-016-2748-5.

McPeek, M. S., and Speed, T. P. (1995). Modeling Inference in Genetic Recombination. *Genetics* 139, 1031–1044. doi: 10.1007/978-1-4614-1347-9_10.

Melchinger, A. E., Geiger, H. H., and Schnell, F. W. (1986). Epistasis in maize (Zea mays L.) - 2. Genetic effects in crosses among early flint and dent inbred lines determined by three methods. *Theor. Appl. Genet.* 72, 231–239. doi: 10.1007/BF00266997.

Melchinger, A. E., Longin, C., Utz, H., and Reif, J. (2005). Hybrid maize breeding with doubled haploid lines: quantitative genetic and selection theory for optimum allocation of resources. in *Forty first annual Illinois corn breeders' school*, 8–21.

Meuwissen, T. H. . E., Hayes, B. J., and Goddard, M. E. (2001). Prediction of total genetic value using genome-wide dense marker maps. *Genetics* 157, 1819–1829. doi: 11290733.

Meuwissen, T. H. E., and Sonesson, A. K. (1998). Maximizing the Response of Selection with a Predefined Rate of Inbreeding: Overlapping Generations. *J. Anim. Sci.* 76, 2575–2583. doi: 10.2527/1998.76102575x.

Moore, F. B. G., and Tonsor, S. J. (1994). A simulation of Wright's shifting-balance process: migration and the three phases. *Evolution (N. Y).* 48, 69–80.

Muleta, K. T., Pressoir, G., and Morris, G. P. (2019). Optimizing genomic selection for a sorghum breeding program in Haiti: A simulation study. *G3 Genes, Genomes, Genet.* 9, 391–401. doi: 10.1534/g3.118.200932.

Podlich, D. W., and Cooper, M. (1998). QU-GENE: a simulation platform for quantitative analysis of genetic models. *Bioinformatics* 14, 632–653. doi:

10.1093/bioinformatics/14.7.632.

Podlich, D. W., Cooper, M., Basford, K. E., and Geiger, H. H. (1999). Computer simulation of a selection strategy to accommodate genotype environment interactions in a wheat recurrent selection programme. *Plant Breed.* 118, 17–28. doi: 10.1046/j.1439-0523.1999.118001017.x.

Popovici, V., Chen, W., Gallas, B. D., Hatzis, C., Shi, W., Samuelson, F. W., et al. (2010). Effect of training-sample size and classification difficulty on the accuracy of genomic predictors. *Breast Cancer Res.* 12, R5. doi: 10.1186/bcr2468.

R Development Core Team (2020). R: A language and environment for statistical computing. Available at: https://www.r-project.org/.

Ren, J., Wu, P., Tian, X., Lübberstedt, T., and Chen, S. (2017). QTL mapping for haploid male fertility by a segregation distortion method and fine mapping of a key QTL qhmf4 in maize. *Theor. Appl. Genet.* 130, 1349–1359. doi: 10.1007/s00122-017-2892-6.

Revilla, P., Anibas, C. M., and Tracy, W. F. (2021). Sweet corn research around the world 2015–2020. *Agronomy* 11, 1–49. doi: 10.3390/agronomy11030534.

Rotarenco, V., Dicu, G., and Fuia, S. (2010). New inducers of maternal haploids in maize. *Maize Genet. Coop. Newsl.*, 21–22.

Sabadin, F., Dovale, J. C., Platten, J., and Fritsche-Neto, R. (2021). Optimizing self-pollinated crop breeding employing genomic selection: from schemes to updating training sets.

Schrag, T. A., Schipprack, W., and Melchinger, A. E. (2019). Across-years prediction of hybrid performance in maize using genomics. *Theor. Appl. Genet.* 132, 933–946. doi: 10.1007/s00122-018-3249-5.

Shull, G. H. (1908). The composition of a field of maize. *J. Hered.* 4, 296–301.

Smith, J. S. C., Hussain, T., Jones, E. S., Graham, G., Podlich, D., Wall, S., et al. (2008). Use of doubled haploids in maize breeding: implications for intellectual property protection and genetic diversity in hybrid crops. *Mol. Breed.* 22, 51–59. doi: 10.1007/s11032-007-9155-1.

Tanksley, S. D., Young, N. D., Paterson, A. H., and Bonierbale, M. W. (1989). RFLP Mapping in Plant Breeding: New Tools for an Old Science. *Nat. Biotechnol.* 7, 257–264. doi: 10.1038/nbt0389-257.

Tessema, B. B., Liu, H., Sørensen, A. C., Andersen, J. R., and Jensen, J. (2020). Strategies Using Genomic Selection to Increase Genetic Gain in Breeding Programs for Wheat. *Front. Genet.* 11, 1–12. doi: 10.3389/fgene.2020.578123.

Tracy, W. F. (1993). "Sweet corn," in *Genetic Improvement of Vegetable Crops* (Boston, MA: Elsevier), 777–807. doi: 10.1016/B978-0-08-040826-2.50058-8.

Tracy, W. F. (2001). "Sweet corn," in *Specialty Corns*, ed. A. R. Hallauer (Boca Raton, FL: Elsevier), 777–807.

Troyer, A. F., and Wellin, E. J. (2009). Heterosis Decreasing in Hybrids: Yield Test Inbreds. *Crop Sci.* 49, 1969–1976. doi: 10.2135/cropsci2009.04.0170.

van Berloo, R., and Stam, P. (1998). Marker-assisted selection in autogamous RIL populations: a simulation study. *Theor. Appl. Genet.* 96, 147–154.

Verges, V. L., and van Sanford, D. A. (2020). Genomic selection at preliminary yield trial stage: Training population design to predict untested lines. *Agronomy* 10, 1–16. doi: 10.3390/agronomy10010060.

Verma, S. S., de Andrade, M., Tromp, G., Kuivaniemi, H., Pugh, E., Namjou-Khales,

B., et al. (2014). Imputation and quality control steps for combining multiple genome-wide datasets. *Front. Genet.* 5. doi: 10.3389/fgene.2014.00370.

Wang, J., Ginkel, M., Podlich, D., Ye, G., Trethowan, R., Pfeiffer, W., et al. (2003). Comparison of Two Breeding Strategies by Computer Simulation. *Crop Sci.* 43, 1764–1773. doi: 10.2135/cropsci2003.1764.

Wang, X., Yang, Z., and Xu, C. (2015). A comparison of genomic selection methods for breeding value prediction. *Sci. Bull.* 60, 925–935. doi: 10.1007/s11434-015-0791-2.

Wetterstrand, K. A. (2020). The Cost of Sequencing a Human Genome. *Cost Seq. a Hum. Genome*.

Windhausen, V. S., Atlin, G. N., Hickey, J. M., Crossa, J., Jannink, J. L., Sorrells, M. E., et al. (2012). Effectiveness of Genomic Prediction of Maize Hybrid Performance in Different Breeding Populations and Environments. *G3 Genes|Genomes|Genetics* 2, 1427–1436. doi: 10.1534/g3.112.003699.

Würschum, T., Reif, J. C., Kraft, T., Janssen, G., and Zhao, Y. (2013). Genomic selection in sugar beet breeding populations. *BMC Genet.* 14, 85. doi: 10.1186/1471-2156-14-85.

Zhao, Y., Gowda, M., Liu, W., Würschum, T., Maurer, H. P., Longin, F. H., et al. (2012). Accuracy of genomic selection in European maize elite breeding populations. *Theor. Appl. Genet.* 124, 769–776. doi: 10.1007/s00122-011-1745-y.

Zystro, J., Peters, T., Miller, K., and Tracy, W. F. (2021). Classical and genomic prediction of hybrid sweet corn performance in organic environments. *Crop Sci.* 61, 1698–1708. doi: 10.1002/csc2.20400.

4 Paper

Optimizing a Sweet Corn Breeding Program: implementing genomic selection and doubled haploid technology

Coelho, I. F.; Peixoto, M. A.; Leach, K.; Lubberstedt, T.; Bhering, L. L.; Resende, M. F. R.

4.1 Abstract

Sweet corn is a high valuable crop, however, it has a restricted planted area because of its high implementation costs due to the fertility, irrigation, intensive pest management requirements. In this way, *in silico* models simulate the implementation of new strategies to breeding programs prior to the field trial. In this study, we simulated the pipeline of the Sweet corn breeding program of University of Florida to investigate new strategies to increase the efficiency of the program, in terms of genetic gains, time spending and costs. Currently, the conventional breeding program (*CONV*) which is running have successive self-pollinations and three testcrosses stages. Beyond the *CONV*, five alternative scenarios to the current were evaluated over 30 years of breeding program, being: i) *CONV* implementing genomic selection (GS) (*GS_CONV*); ii) the using of double haploids (DH) lines instead of the successive self-pollinations(*DH_CONV*); iii) adopting DH and GS in a breeding program (*DH_GS*); iv) adopting DH lines and reducing one testcross stage (TC1) (*FAST_DH*); and, v) using GS into the *FAST_DH* strategy (*FAST_DH.GS*). Two different genotype-by-environment interactions (GxE) were compared to understand the dynamics of each breeding strategy: a scenario with absence of GxE and other with presence of GxE. When the GxE was null, the scenarios had similar performances of hybrid gains, only increasing the budget, the only significant difference. Under presence of GxE, the *FAST_* scenarios were the most potential strategies in logistic and breeding terms,

generating better hybrids, releasing a higher number of lines in a shorter time, and increasing a reasonable amount in the budget of the program. Compared to the *CONV* strategies, the increase in price was around 60%, and the gains climbed around 20% in the *FAST* scenarios. The number of lines released was 40% higher after 15 years of breeding program, demonstrating the potential of the strategy, to produce top performances materials being competitive in the market.

4.2 Introduction

The sweet corn has many aspects of similarity with field corn. However, it does not present well-defined heterotic groups, as field corn does. The probable most important reason for that is the number of important key traits to make a successful hybrid, other than yield. As a specialty vegetable, the table quality is also very important in sweet corn evaluation. In many cases, some qualitative traits (e.g. flavor) have recessive or additive inheritance, meaning that would be interesting to have both parents bringing the same alleles to the hybrid. In consequence, it would probably have to develop parental lines from the same high quality inbred (Tracy, 2001). The only way to select top materials to generate good hybrids is by using testers, which are inbreds of proven commercial values. Also, the selection becomes an efficient process when using good testers with recessive regions to important traits already set. Where the good hybrids and tested lines would express the undesirable trait if they would not have the recessive alleles as the testers (Tracy, 2001).

Even being a high input, and very valuable vegetable crop, the sweet corn planted area is restricted to few areas because of the high requirements of the plant, as fertility, irrigation, and intensive pest management techniques (Revilla et al., 2021). This also brings a high budget into the breeding program. In this way, simulations

models can be useful for a breeding program in terms of optimizing the breeding pipeline, by comparing the inclusion of tools and strategies (Gorjanc et al., 2016). The *AlphaSim* was developed by Faux et al. (2016) and implemented in R in 2020 (Gaynor et al., 2020). It has been widely adopted in different crops (Cowling et al., 2020; Tessema et al., 2020; Batista et al., 2021; Sabadin et al., 2021). According to Faux et al. (2016), simulation has the potential of evaluation and develop new strategies for plant breeding. In this way, the prediction would not use the information of field experimentation, which reduced the costs of a breeding program as also speed up the time of making decision of the best strategy to adopt, or the best pipeline to follow.

The using of doubled haploid (DH) as potential lines was suggested by Chase (1949, 1952), and since then many improvements have been made on the process of DH production, by: the increasing in the haploid induction rate (Rotarenco et al., 2010; Kelliher et al., 2017; Liu et al., 2017); more efficient ways of identification of haploid kernels, as the R1-navajo gene (Coe and Sarkar, 1964) with colored aleurone crowns and scutella for diploid and colorless scutella for haploid kernels (Melchinger et al., 1986; Liu et al., 2016); and, higher rates of genome doubling (Barnabás et al., 1999; Eder and Chalyk, 2002; Kato, 2002; Häntzschel and Weber, 2010; Ren et al., 2017). The implementation of DH is already well set in maize breeding programs, however, during the identification process of haploid kernels, the R1-navajo gene expression can be masked by genetics and environment factors, which is frequent in the sweet corn plants (Revilla et al., 2021).

Many maize breeding programs combine the DH technology with genomic selection (GS) to increase the gain and shorten the time per breeding cycle (Marulanda et al., 2016; Andorf et al., 2019). The GS builds a predictive model and can bring an increasing in the accuracy by adopting the estimated breeding value to select the best

genotypes and circumvent the genotype-by-environment interaction (Jannink et al., 2010; Heffner et al., 2011; Muleta et al., 2019). Combining with GS and DH, the recurrent selection is a breeding strategy where the best genotypes of an specific stage/population are used as the parents of the next breeding cycle (Hallauer and Darrah, 1985). Labroo and Rutkoski (2021) demonstrated that the using of Reciprocal Recurrent Selection (RSS) strategy by replacing the whole parental population (Discrete) outperformed the partial updating of the parental lines (Partial).

Under the current scenario of sweet corn breeding programs, and with the availability of different tools to be implemented to bring potential gains to the pipeline, this study investigated the implementation of the DH, GS, and RSS strategies into a sweet corn breeding program (*SweetBP)*. Based on a simulated data, we compared genetic parameters, as genetic means and genetic variances of crossing block lines, and the performance of the released hybrids; logistic parameters, such as number of plots, costs per year and breeding cycle length; and the combination of them (genetic gain, budget, and time). Our main goal was to optimize the *SweetBP* pipeline based on the genetic gains and feasibility of bringing new tools to the program, including also the business perspective of a breeding program.

4.3 Material and Methods

Stochastic simulations were adopted to mimic a real breeding program, based on the current pipeline of the sweet corn breeding program of University of Florida. All simulations sets were run at HiPerGator, the University of Florida's supercomputer. Based on these simulations it was evaluated the potential returns of varying the breeding strategies and implementing biotechnologies into the pipeline. The investigated methods were:

- the most efficient breeding strategy to update the parental population;
- the implementation of genomic selection (GS) in the currently conventional breeding program;
- the using of doubled haploid (DH) technology in the breeding program;

All tested scenarios were generated over a 30-years' breeding program for comparisons purposes. Also, a 20 years *burn-in* phase were implemented, being the same for all scenarios. The *burn-in* phase represents the past years of the breeding program, which all scenarios passed through. The following 30 years will be called "Future years", they represent the upcoming years, where the strategies will be evaluated and selected according to their performance. Each scenario was replicated 50 times.

After introducing the methods of this topic, the materials are subdivided as following:

- the simulation of the founder population and the trait architecture;
- the burn-in phase and its populations' dynamics;
- the tested scenarios;
- the parameters' evaluated;
- the costs' study;

4.3.1 Simulation process

Genome sequence data

A genome containing 10 chromosome pairs was generated corresponding to the sweet corn genome. The Markovian Coalescent Simulator (MaCs, Chen et al., 2009) was used to generate each chromosome with a genetic length of 2.0 Morgans and a physical length of $2x10^8$ base pairs. The *MaCS* was used through the *AlphaSimR* package (Gaynor et al., 2020; R Development Core Team, 2020). The recombination

and mutation rates also were included within the package, being $1.25x10^{-8}$ and $1x10^{-8}$, respectively. For diploid species, *AlphaSimR* follows the gamma model to simulate the meiosis and genetic recombination, for diploid species, accommodating crossover interference (McPeek and Speed, 1995). This model fits real data (Broman and Weber, 2000), and the magnitude of crossover interference is set by the user, through a single parameter (Gaynor et al., 2020). Also, a split of two different populations was simulated 30 generations ago. This split mimics the first reported sweet corns' genotypes (breeding pools, populations, synthetics and composites) in United States, in 1980's (Tracy, 1993). A brief explanation of the breeding scheme is given by Table 1.

Table 1: Breeding scheme split by phases, simulation stages, details of parameters, and features.

Simulation Phases	Breeding Stages	Parameters and Features Information
Genome sequence	Genome Sequence	Maize historical effective population size 10 chromosome pairs (2.0 Morgan each) Physical length $2x10^8$ base pairs Mutation rate of 1×10^{-8} Recombination rate of 1.25×10^{-8}
Burn-In Years Breeding	Founder Population	200 inbred lines 3000 SNP/chromosome 300 QTN/chromosome QTN effects under normal distribution
	Past Years Breeding	ADG trait 20 years of baseline breeding program
Future Years Breeding	Future Years Breeding	ADG trait Test four G×E effects and two H^2 magnitudes Apply different technologies and techniques to make different breeding strategies Compare the strategies Calculate hybrid gains Compare strategies' costs and effectiveness

*The ADG trait means a trait controlled by additive, dominance, and genotype-by-environment (G×E) interaction effects. SNP: single nucleotide polymorphism. QTN: quantitative trait nucleotide.

4.3.2 Burn-In Years Breeding Phase

Founder Population

Based on the simulated genome sequences, 200 founder genotypes were generated in Hardy-Weinberg equilibrium, being two different groups as set by the split option. These genotypes originated from random samples of the 10 chromosomes pairs per individual. It was simulated a set of 300 quantitative trait nucleotides (QTNs) and 3,000 single nucleotide polymorphisms (SNPs) were randomly selected across each chromosome.

Trait Architecture

Aiming to mimic a trait very similar to yield and a target of the *SweetBP*, a quantitative trait was simulated based on previous works and on the historical dataset of the target program. The trait was controlled by 3,000 QTNs (10 chromosomes x 300 QNTs per chromosome) with additive, dominance, and genotype-by-environment interaction (G×E) effects, referred as ADG trait in *AlphaSimR*. Each QTN was assigned an additive genetic value, its interaction with the environment, referred as G×E effect of each quantitative trait nucleotide (QTN), and the dominance effect due to the interaction between alleles in the same locus.

The mean (meanA) and variance (varA) additive effect were meanA = 150 (bushels/acre) and varA = 30. The additive effect is a result of a summation over all QTNs for the product of the additive effect, and a dosage of the scaled additive effect of the given QTN. The scaled additive considers the set varA and involves the dominance effect in the equation. The calculation of the additive effects as follows:

$$\boldsymbol{A(x)} = \sum_{i=1}^{i} \boldsymbol{a x_A} \; ; \qquad [1]$$

where $\boldsymbol{A(x)}$ is the additive effect of each individual; $\boldsymbol{i}$ refers to each QTN, $\boldsymbol{a}$: additive effect and $\boldsymbol{x_A}$ is the scaled additive dosage, in each QTN.

The dominance effect of each individual is given by the summation over all QTNs for the product of the dominance effect, scaled by the dominance dosage. As the *AlphaSimR* uses the concept of dominance degrees to calculate the dominance effect, it will be affected by the additive effect, and all the previous explanation will be applied in here too. The dominance effect was built according to the dominance degree of the trait, with variance of 0.3, and the average value of 0.93. The average dominance degree represents the presence of a dominance deviation (δ) from the genetic mean of the homozygous. Due to an interaction between alleles, the dominance effect shift the mean, evidencing the presence heterosis and agreeing with historical data of commercial maize (Troyer and Wellin, 2009). It is worth remembering that these parameters are all set and will be replicated just in the founder population. As the breeding program starts with crosses and population advancements, the genetic control changes in each stage, according to the respective genic and allelic frequencies.

The functions of the dominance effects are equal to:

$$\boldsymbol{D(x)} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{i}} \boldsymbol{dx_D}; \text{ and,} \qquad [2]$$

$$\boldsymbol{d} = \boldsymbol{\delta|a|} \; ; \qquad [3]$$

where $\boldsymbol{D(x)}$ is the dominance effect of each individual; $\boldsymbol{i}$ refers to each QTN, $\boldsymbol{d}$: dominance effect and $\boldsymbol{x_D}$ is the scaled dominance dosage, in each QTN; $\boldsymbol{\delta}$ is the dominance degree at that locus/QTN, and $\boldsymbol{|a|}$ is the absolute value of that locus/QTN ($\boldsymbol{i}$) additive effect

The G×E effect calculation includes the meanA and varA, but also an environmental component (EC). The average value of EC is 0 and the variance is 1.

The meanA will be included in a specific term along with the varA, and it will be scaled as previously, to achieve a desired mean and variance, set by the user, in the founder population. The EC was set with two different magnitudes in this study. All interactions have a target environment, so, we adopted the first EC regarding to the genotype-by-year interaction effect (EC_{gy}), referring to the same locality of the target environment, planted in different years. The EC_{gy} would consider only the across years weather differences. The second EC considered the genotype-by-year-by-environment interaction effect (EC_{gye}), representing the different localities of the target environment planted in different years. The EC_{gye} would considerate not only the across years weather differences, but also the variation of soil, altitude, and other specific local weather parameters. The genotype-by-environment effects are calculated by the following functions:

$$\boldsymbol{G}(\boldsymbol{x}, \boldsymbol{\omega}) = \boldsymbol{\omega}\boldsymbol{b}(\boldsymbol{x}); \text{ and,} \quad [4]$$

$$\boldsymbol{b}(\boldsymbol{x}) = \mu_{\boldsymbol{G}} + \sum_{\boldsymbol{i}=1}^{\boldsymbol{i}} \boldsymbol{g}\boldsymbol{x}_{\boldsymbol{A}} ; \quad [5]$$

where $\boldsymbol{G}(\boldsymbol{x}, \boldsymbol{\omega})$ is the genotype-by-environment interaction effect of each individual; $\boldsymbol{\omega}$ refers to an environmental covariate, $\boldsymbol{b}(\boldsymbol{x})$ is a genotype specific slope, $\mu_{\boldsymbol{G}}$ is an intercept value, $\boldsymbol{i}$ is the QTN, $\boldsymbol{g}$ is the genotype-by-environment effect and $\boldsymbol{x}_{\boldsymbol{A}}$ is a scaled additive dosage.

Combining with the EC magnitudes, four different values of G×E effect variances (VarGE) were evaluated in this study. The values were related to the additive variance, being 0, 2, 4 and 8 times the VarA (0, 60, 120, and 240, respectively). Following, aiming to simplify the discussion the VarGE of 0 and 240, will be mentioned

as Absence and Presence of G×E, respectively, all the others will be exposed in the supplementary material as 2xVarA and 4xVarA.

Furthermore, as the trait yield is very complex and controlled by several genes in the whole genome, it was simulated a trait with 0.3 of broad-sense (H^2) heritability.

Past Years Breeding Program

The Figure 1 presents a detailed explanation of the conventional (*CONV*) breeding pipeline that is developed in the first 20 years of simulation. The *CONV* is set prior to any strategy simulation aiming to mimic the development of the sweet corn breeding program (*SweetBP*), of University of Florida. Thenceforth, all strategies will start from the same point, where the population is already under breeding process and have realistic values of alleles frequencies, genetic mean, and variance. Simultaneously to the *SweetBP,* a private program was also simulated. Our study tried to mimic the private industry by making a large size breeding program, it had the same five-years of successive self-pollinations and selections, and generating five elite lines every breeding cycle, which are used as testers of our program, and are updated yearly.

Testers genotypes

In paralallel with the *SweetBP*, a breeding program mimicking a private breeding program was set. The difference between the conventional *SweetBP* and the *Private* breeding program was the size. Aiming to simulate the magnitude of a breeding program in terms of number of tested genotypes and released lines and hybrids, the private breeding program was reproduced in a four times larger scale than the *SweetBP*. The main goal of this parallel breeding program was to generate and supply the testers of the top crosses stages of the *SweetBP*. The five best released lines from the private testers were used as the testers of the *SweetBP*, being updating yearly

trying to simulate the commercial lines used as testers by the *SweetBP*. The updating by year is due to the uncertainty of the genotypes into the *SweetBP*, which are decided out of the *SweetBP* staff.

4.3.3 Future Years Breeding Phase

The baseline breeding strategy currently running at the *SweetBP*, is based on the classical approaches. The main objective of the program is producing good hybrids in partnership with the private industry. So, generating lines with high combining abilities to industry's elite lines (represented by the testers) is the *SweetBP* goal (Figure 1).

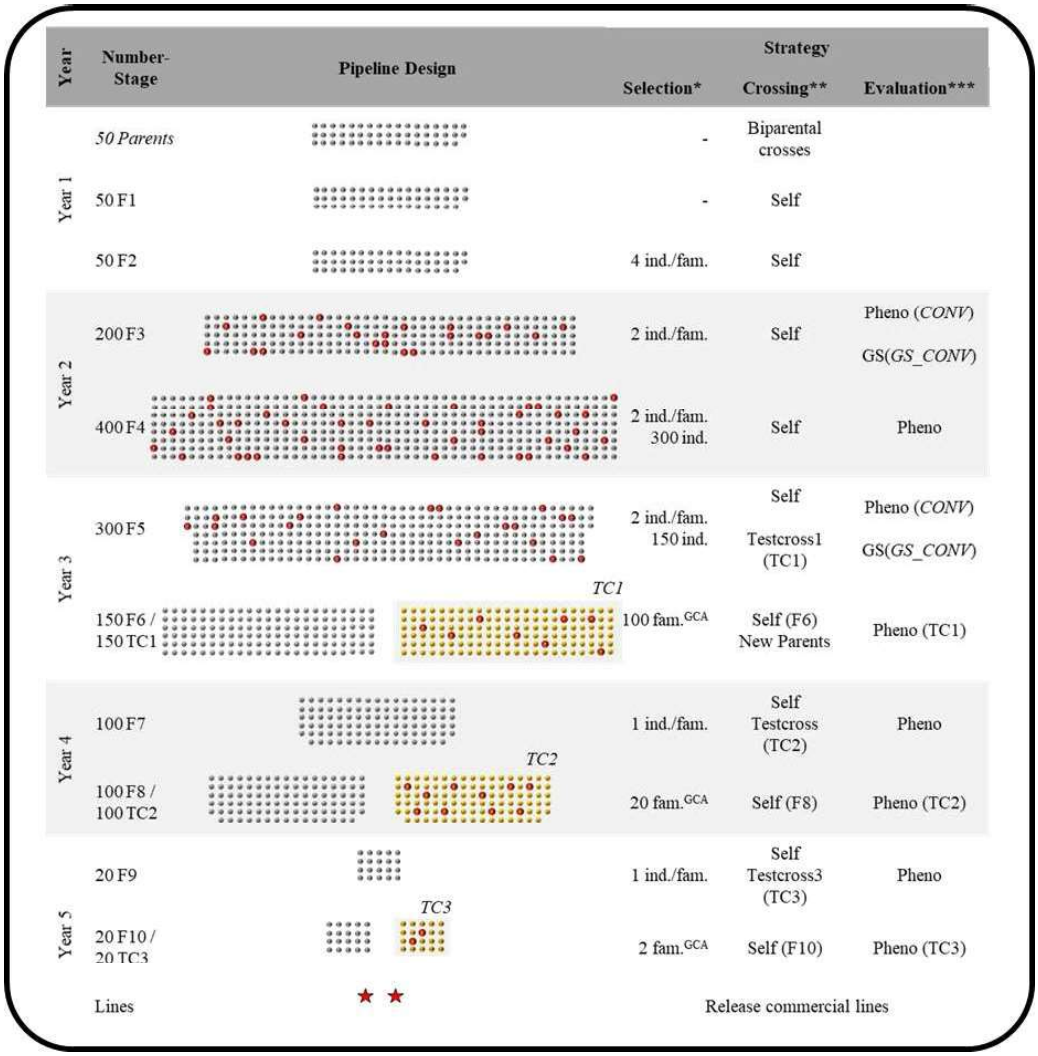

Figure 1. Sweet corn breeding program pipeline. Detailed explanation of each stage dynamics. The pipeline whole cycle completes in 5 years to generate two new elite lines. Every 4 years there is an update of the Parental lines. Testcross 1 (TC1) is developed in 2 environments, testcross 2 (TC2) in 5 environments and testcross 3 (TC3) in 20 different environments, using respectively 1, 2 and 5 testers. *Selection of individuals (ind), individuals per families (ind/fam), and families based on general combining ability (GCA); ** Crossing: self-pollination (self); and ***Evaluation based on Phenotypic values (Pheno) and Estimated breeding Values (GS). CONV: conventional, GS_CONV: Conventional using genomic selection.

Updating crossing block

The breeding cycle have two harvest seasons per year (Fall and Winter). Only in the year 1 of the breeding cycle the biparental crosses are developed during the spring, generating the F1 families. In this study, the crossing block, which contains the parental population, is composed by 50 lines and it is updated every third year of the breeding cycle. By selecting the highest general combining ability (GCA) of testcross1 (TC1) genotypes, the best F6 genotypes (potential parents), which were planted simultaneously, will be taken to the parental updating.

The first step of this study was to select the best way to update the crossing block aiming to have higher and faster gains of the crossing block genotypes, which is highly correlated with the hybrids' performances of the breeding program. Thus, it was compared two methods of recurrent selection: i) by overlapping the generations (crossing block individuals from the last cycle and potential new parents) and selecting the best 50 lines (Partial); and, ii) substituting the whole crossing block by the potential parents (Discrete).

Breeding strategies simulations

The current *SweetBP* was used as the reference pipeline in this study (*CONV*), with the real, or at least very close to real, number of plots, costs, and cycle.

i. *Conventional breeding strategy* (*CONV*)

As demonstrated in Figure 1, the *CONV* strategy is the current running pipeline of the *SweetBP*. By making successive self-pollinations and selections over seasons, and developing three testcrosses stages, starting with testcross 1 (TC1) with one tester in 2 environments, followed by testcross 2 (TC2), crossing with three testers and evaluating in 5 different environments, finishing in testcross 3 (TC3), evaluating a genotypes with five different testers in 20 different environments.

ii. *Conventional Genomic Selection breeding strategy* (*GS_CONV*)

The only difference between *CONV* and *GS_CONV* strategies is the way to make the selection in F3 and F5 stages. While *CONV* uses the phenotypic values, from the field evaluation, to rank and select the top genotypes, the *GS_CONV* will use the estimated breeding values (*EBV*) given by the genomic model and the markers of the individuals.

An important step of the genomic selection methodology is the model development, training, and validation. Aiming to compare all scenarios, the genomic selection methods must develop a training set (TS) to create and validate the genomic model. In this way, from the crossing block, it was generated 800 F1 families, 50 of them were part of the breeding program and advanced as the *CONV* strategy did. The other 750 F1 families were planted as a training set in the target environment, so, they were phenotypic evaluated and genotyped to develop a genomic model, by training

the model and validating using a cross validation method. The GS model used to generate the *EBV* in this study was a ridge-regression best linear unbiased prediction (*RRBLUP*) (Endelman, 2011), just considering additive genetic effects, according to the following equation:

$$\boldsymbol{y} = \mathbf{1}\,\mu + \boldsymbol{Z}_u\boldsymbol{u} + \boldsymbol{\varepsilon}; \qquad [6]$$

where $\boldsymbol{y}$ is the vector of individual phenotypic values from the training set (this will become the *EBV* when using the genomic selection to predict the breeding values); μ is the mean (intercept), as fixed effect; $\boldsymbol{u}$ is the vector of marker effects (in this study we had 3,000 SNPs per chromosome, which will be 30,000 SNPs per individual), where $\boldsymbol{u} \sim N\,(\mathbf{0}, \boldsymbol{I}\sigma_u^2)$; and $\boldsymbol{\varepsilon}$ is the vector of residuals, as random effects. $\mathbf{1}$ is the vector of ones, and $\boldsymbol{Z}_u$ is the incidence matrix of the training set genotypes for the 30,000 markers. In this study, $\boldsymbol{Z}_u$ is coded as 1, -1 and 0 for $\boldsymbol{A}_1\boldsymbol{A}_1$, $\boldsymbol{A}_2\boldsymbol{A}_2$ (homozygous) and $\boldsymbol{A}_1\boldsymbol{A}_2$ (heterozygous) alleles, respectively.

By using the *GS* model, the *GS_CONV* generates the EBVs for the F3 and F5 families, and it was selected the top families (and/or individuals, when applicable) based on them, instead phenotypic values.

iii. Conventional Doubled Haploid breeding strategy (DH_CONV)

The doubled haploid (DH) technology generates pure lines in only two generations, being able to shorten the breeding cycle considerably. The *DH_CONV* brings this technology on the *CONV* pipeline by generating the DH, pure lines, in the second year and saving the successive generations of self-pollinations (Figure 2). The breeding cycle time will still be the same due to the number of testcrosses, which will have to repeat over the third, fourth and fifth years. Even though the number of plots

is superior, it does not agree with the required labor which will reduce, as well as the logistic of the breeding program.

iv. Genomic selection DH breeding strategy (DH_GS)

The *DH_GS* tries to optimize the use of plots by excluding the lowest performing DH families, indicated by their EBV (Figure 2). The bottom ranked DH families will not be multiplied to be crossed and evaluated in the testcrosses. This would reduce the number of planted plots.

v. DH breeding strategy making 2 testcrosses (FAST_DH)

The *FAST_DH* generates only two testcrosses (Supplementary Material - Figure S1). After generating all the DH families and multiply the seeds, instead of excluding the low potential DH families, it will select the top families based on their phenotypic evaluation. Being both multiplied and testcrossed simultaneously, reducing one year of the program and the number of planted plots as well. The only difference with the previous strategies is that the testcross will be straight to the testcross 2. The update of the parents will be based on the individuals with better GCA performance from *TC2*, and best EBV of the DH families to complete the 50 lines of the crossing block.

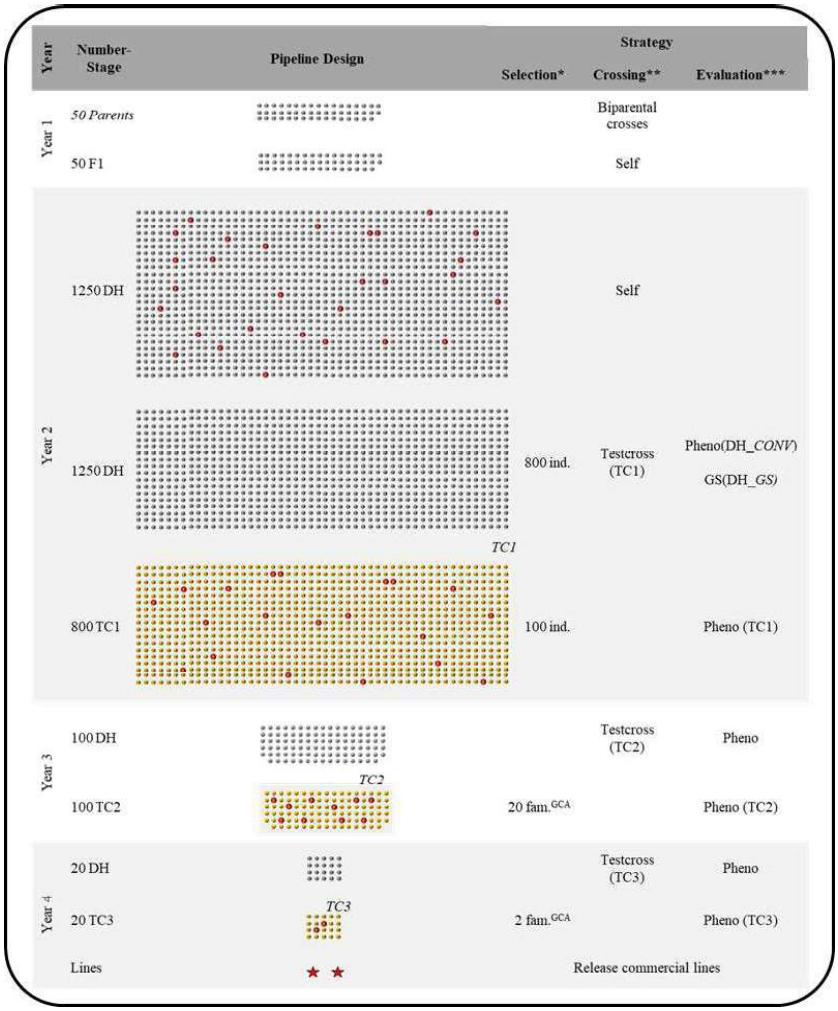

Figure 2. Sweet corn DH_CONV and DH_GS breeding program pipelines. Detailed explanation of each stage dynamics. The pipeline whole cycle completes in 4 years to generate two new elite lines. Every 2 years there is an update of the Parental lines. Testcross 1 (TC1) is developed in 2 environments, testcross 2 (TC2) in 5 environments and testcross 3 (TC3) in 20 different environments, using respectively 1, 2 and 5 testers. *Selection of individiduals (ind), individuals per families (ind/fam), and families based on general combining ability (GCA); ** Crossing: self-pollination (self); and ***Evaluation based on Phenotypic values (Pheno) and Esimated breeding Values

(GS). DH_CONV: conventional using doubled haploid technique, DH_GS: DH_CONV using genomic selection.

vi. DH breeding strategy making 2 testcrosses using genomic selection (FAST_DH.GS)

The only difference of *FAST_DH.GS* to *FAST_DH* will be on the selection of DH lines (Supplementary Material - Figure S1.1). After the training and validation of the genomic model, the best DH families will be selected based on their EBV. Being both multiplied and testcrossed simultaneously, reducing one year of the program and the number of planted plots as well.

Genetic Parameters and Costs

Aiming to compare the scenarios, the following genetic parameters were generated for each year/stage:

i. Genetic Parameters

By using the genetic values given by the allelic frequencies, additive, and dominance effect. The parameters were calculated for both hybrid's (testcross 3) and parental's (crossing block) populations, as follows:

- ***Genetic Mean (meanG)****:* calculated by the sum of the individuals' genetic values, where the genetic value is given by the sum of genetic additive and dominance values given by each QTN;
- ***Genetic Variance (varG)****:* calculated by the variance of the individuals' genetic values, and given by population variance, i.e. divides by all *n* individuals of the population;

- ***Genetic Gain (Gain)****:* calculated by the difference of the *meanG* of the year *i* and *0*, where *i* is the specific year of the Future Breeding Phase, from 1 to 30, and X_0 is the last year of the Burn-In Breeding Phase (year 0) (Table 1);
- ***Gain Efficiency****:* calculated by the $Gain_i$ over the standard deviation (SD_i) loss (given by $\sqrt{varG_i} - \sqrt{varG_0}$), where *i* is the specific year of the Future Breeding Phase, from 1 to 30, and *0* is the last year of the Burn-In Years Breeding Phase (year 0);
- ***Accuracy****:* calculated by the correlation of the true genetic values (*GV*) of each individual and the phenotypic (*Pheno*) or estimated breeding value (*EBV,* given by equation [6]), depending on the strategy. The equation of GV, phenotypic value and accuracy are:

$$GV(x)_j = A(x)_j + D(x)_j \; ; \qquad [7]$$

$$Pheno(x)_j = GV(x)_j + \varepsilon(x)_j \; ; \text{ and,} \qquad [8]$$

$$\rho_{(GV,w)} = (COV(GV,w))/(\sigma_{GV}.\sigma_w); \qquad [9]$$

where, x is the individual; j is the stage; ε is the random error associated to the individual; and, $\rho_{GV,w}$ is the accuracy, given by the correlation of the genetic values and the w are the values of the individuals in that population, where it can be associated to the *EBV* or $Pheno$, depending on the adopted strategy.

ii. *Costs*

Based on the current *SweetBP* logistic data, some parameters were estimated:

- ***Relative Gain****:* calculated by the hybrids' $Gain_j$ over the hybrids' $Gain_{j0}$, where *j* is the compared scenario, and *j0* is the *CONV* scenario already running in the *SweetBP;*

- ***Annual Costs ($)****:* calculated by the sum of costs per plot in the field (both for evaluation or crossing), costs per genotyped sample, and costs to generate doubled haploid plants;
- ***Relative Costs****:* calculated by the $Annual\ Costs_j$ over the $Annual\ Costs_{j0}$, where *j* is the compared scenario, and *j0* is the *CONV* scenario already running in the *SweetBP;*
- ***Cycle****:* number of years to complete the breeding cycle and release an elite line;
- ***Relative Cycle****:* calculated by the $Cycle_j$ over the $Cycle_{j0}$, where *j* is the compared scenario, and *j0* is the *CONV* scenario already running in the *SweetBP;*
- ***Gain Efficiency****:* calculated considering the *Relative Gain divided by the* multiplication of *Relative Costs* and *Relative Cycle*, as following:

$$Gain\ Efficiency_j = \frac{(Relative\ Gain_j)}{(Relative\ Cost_j)(Relative\ Cycle_j)}; \qquad [10]$$

where *j* is each scenario.

4.4 Results

The results will follow the material and methods topics' order, starting from the best strategy of recurrent selection strategy until the costs study. The presented Figures and tables bring the two extremes of G×E interaction effects, which will be referred as "Absence" and "Presence" of GE (0xVarA and 8xVarA, respectively). The plots and tables which are not presented here will be in the supplementary materials.

In summary, the Discrete Recurrent Selection strategy delivered gain over 7% under presence of G×E, both for hybrids and parental population, even after a

significant loss in genetic variance. Worth remembering that the variance loss was a reality in both strategies.

When compared all the scenarios with the current *SweetBP* (*CONV*), the *FAST* strategies overwhelmed the other scenarios, having hybrids gain over 17% over the 30 years, under presence of G×E, while the others had around 5% of relative gain. Even though the budget inflated around two times more comparing to the *CONV*, the breeding cycle time reduced in 2 years and the number of plots per year increased 10%, using the *FAST* strategies.

Updating crossing block

Overall, the results demonstrate a superiority of the discrete Recurrent Selection strategy, by replacing the full block crossing with the top F6 genotypes. For the genetic variance, both G×E scenarios had significant decay when compared the Discrete by the Partial Recurrent Selection strategies. Under presence of G×E, the variance shrank by up to 80%, while the Partial strategy reduced up to 70%, after the 30 years of future breeding program (Table 2, Supplementary Material - Figure S2.1 and S2.3).

Table 2. Parental parameters of the tested strategies which applied Discrete and Partial Recurrent Selection Strategies, and the comparison of Discrete over Partial strategy. The tested strategies applying genomic selection were conventional (CONV), conventional using genomic selection (GS_CONV), and the conventional double haploid strategy (DH_CONV). All strategies were analysed under presence and absence of genotype-by-environment interaction (G×E).

Parental Parameters	Strategy	Absence of G×E		Presence of G×E	
		Partial	Discrete	Partial	Discrete
Genetic Mean	CONV	210.38	217.43	163.23	194.26
	GS_CONV	208.61	214.45	165.39	196.19
	DH_CONV	198.66	203.32	169.50	182.89
Genetic Variance	CONV	17.92	5.56	27.96	5.77
	GS_CONV	18.23	5.35	22.94	6.37
	DH_CONV	10.65	7.07	17.36	9.39

The Table 3 demonstrates the superiority of strategies under presence of G×E. The discrete strategy performance of all three strategies overcomes from 7 to 19% in the parental genetic means.

Table 3. Parental parameters of the tested strategies which applied Discrete and Partial Recurrent Selection Strategies, and the comparison of Discrete over Partial strategy. The tested strategies applying genomic selection were conventional (CONV), conventional using genomic selection (GS_*CONV*), and the conventional doubled haploid strategy (*DH_CONV*). All strategies were compared under presence and absence of genotype-by-environment interaction (G×E).

Parental Parameters	Strategy	Performance of Discrete over Partial Recurrent Selection Strategies (%)	
		Absence of G×E	Presence of G×E
Genetic Mean	CONV	3.35	19.01
	GS_CONV	2.80	18.63
	DH_CONV	2.35	7.90
Genetic Variance	CONV	-68.96	-79.35
	GS_CONV	-70.63	-72.24
	DH_CONV	-33.61	-45.94

Breeding Strategies

The Figure 3 presents a summary of the strategies' cycles and plots numbers. The FAST_ strategies brought a reduction of two years of the CONV's cycles. The number of plots was also reduced when adopted the GS tool with the FAST strategy.

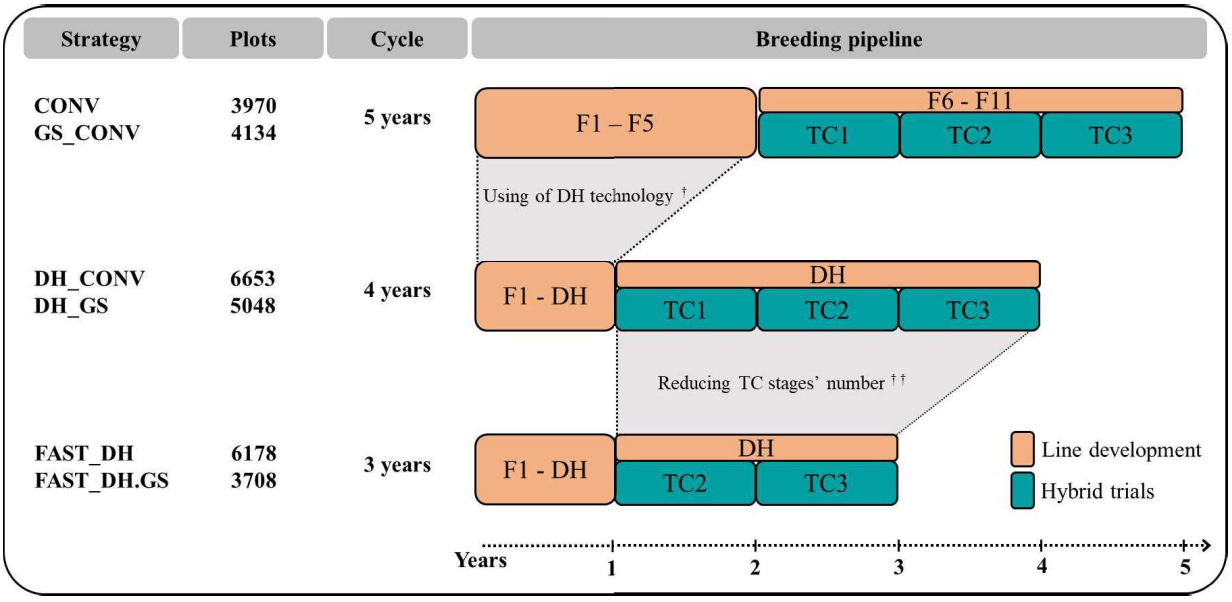

Figure 3. Breeding pipeline of all strategies, with plots and cycles information. The tested strategy were: the conventional strategy (CONV) with successive self-pollination's over years; the CONV including genomic selection (GS) (GS_CONV); the implementation of the double haploids technology (DH) (†) (DH_CONV), and the the

inclusion of GS with it (DH_GS); and, the withdraw of testcross 1 (TC1) (††) (FAST_DH), and applying GS (FAST_DH.GS). F1 to F11 stands for the stages of the breeding cycle; DH: doubled haploid; and, TC1 to TC3 are the testcrosses 1 to 3.

Breeding program performance over 30 years

The Figure 4 brings the performance of the hybrids over the 30 years, and the number of breeding cycles of each scenario, including the number of released lines after 15 and 30 years. The FAST_DH and FAST.DH.GS strategies demonstrated superior hybrid performances compared to the other strategies, releasing a greater number of lines with higher gains after 15 and 30 years (Figure 4). Even though the presence of G×E affected the performance of hybrids mean (Supplementary Material - Figure S3.1), the genetic gains remained relatively similar (Supplementary Material - Figure S3.2).

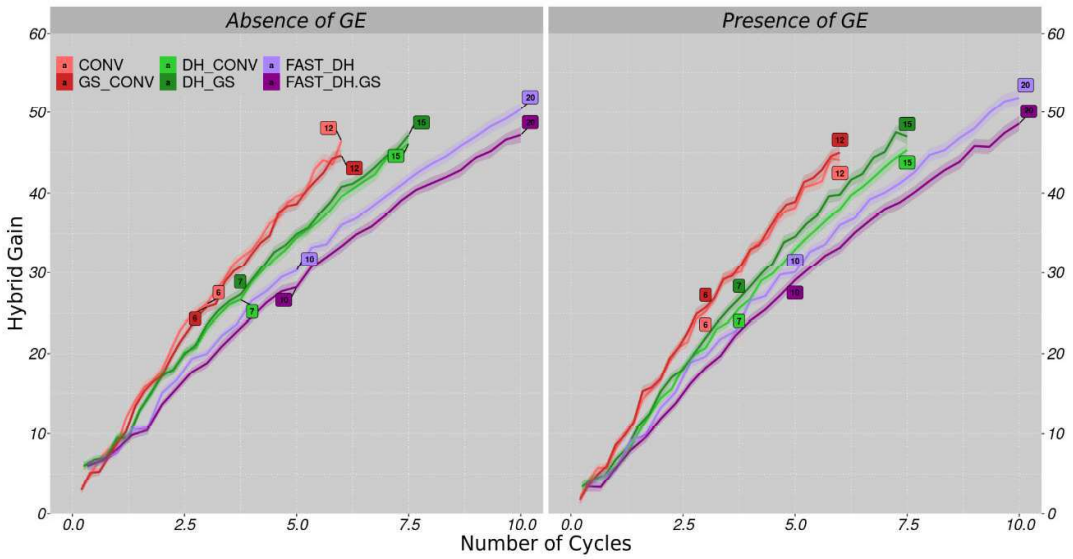

Figure 4. Hybrid genetic mean by the number of completed cycles of each strategy, under presence and absence of genotype-by-interaction effect (GE). The conventional (CONV) and conventional using genomic selection (GS)

(GS_CONV) strategies were compared to the double haploids (DH) conventional and GS (DH_CONV and DH_GS, respectively), and with the DH adopting only two testcross' stages (FAST_DH) and applying genomic (FAST_DH.GS). Labels in the middle of the line denote to the number of lines released after 15 years of breeding program, and the labels in the end of the lines were the number of lines released after the 30 years of breeding.

The gain of genetic mean and loss of genetic variance of the parental lines over the years were also evaluated (Supplementary Material - Figure S4.1 – S4.3). The DH_CONV and DH_GS demonstrated a higher efficiency in returning hybrid gain over the reducing of the parental genetic variance (presented as standard deviation). The CONV and GS_CONV return same gains of DH_CONV and DH_GS however they lost more variance, while the FAST_DH and FAST.DH.GS strategies lost more parental variance but also obtained higher hybrid gains.

Costs Study

The *CONV* budget mimics the current approximated costs of the *SweetBP.* The calculated budget included costs of evaluation and crossing plots, applied technologies, and staff labor. Overall, the implementation of the GS technology increases the costs considerably from *CONV* to *GS_CONV*. Both *DH* strategies (*DH_* and *FAST_*) presents higher number of plots, due to the number of generated *DH*'s to be evaluated in the second year of breeding. The Figure 5 demonstrates a comparison of gains by costs. The *DH_FAST* had the third lowest budget, around two times more expensive, and achieved the highest hybrid gain, over 20% higher under presence of

G×E, while *DH_* strategies, instance presented hybrid gains lower than 6% and a budget more than two times higher than the *CONV*.

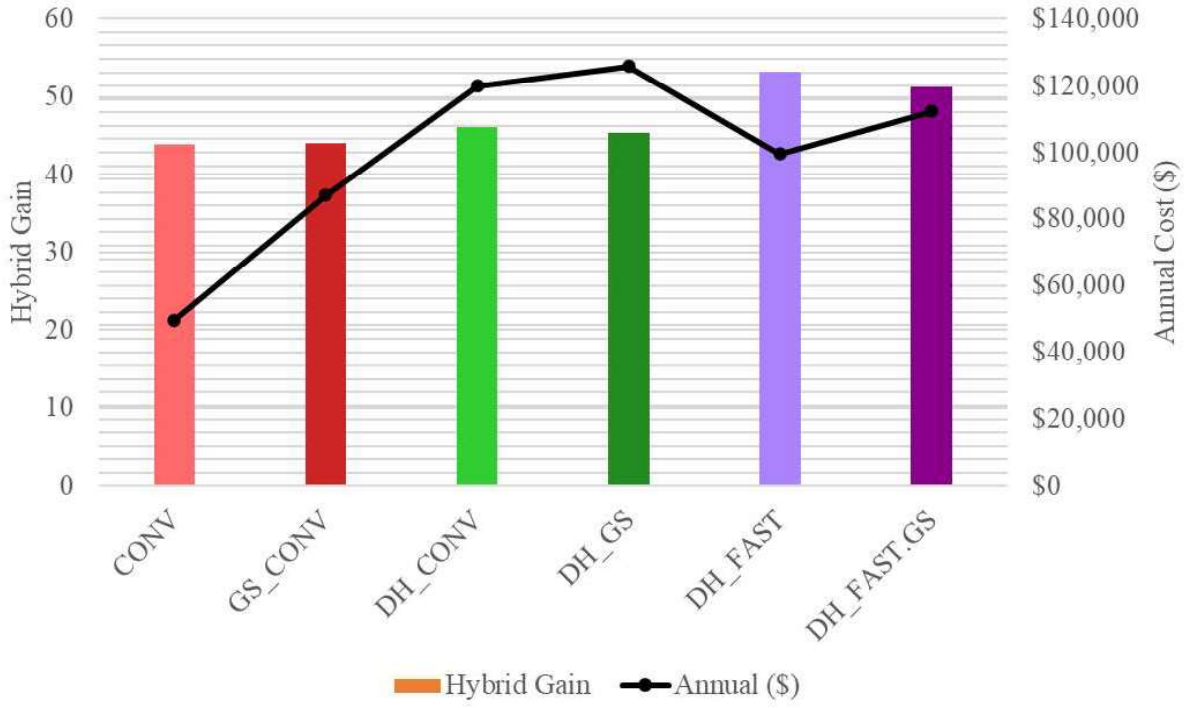

Figure 5. **Hybrid Gain after 30 years of breeding program (Primary Y axis) and Annual Costs (Secondary Y axis) in each scenario.** The conventional (CONV) and conventional using genomic selection (GS) (GS_CONV) strategies were compared to the double haploids (DH) conventional and GS (DH_CONV and DH_GS, respectively), and with the DH adopting only two testcross' stages (FAST_DH) and applying genomic analysis (FAST_DH.GS). The gains considered a scenario with presence of genotype-by-environment interaction.

The Table 3 brings the *Gain Efficiency* parameter in the cost's analysis, which includes the length of the breeding program in its equation. Under this perspective, the *FAST_DH* demonstrates a higher efficiency than the *CONV*, even though it is twice more expensive.

Table 4. Breeding strategies size, time, and performance information. The scenarios (Scen) were: conventional (CONV) and conventional using genomic selection (GS) (GS_CONV) strategies were compared to the doubled haploids (DH) conventional and GS (DH_CONV and DH_GS, respectively), and with the DH adopting only two testcross' stages (FAST_DH) and applying genomic (FAST_DH.GS). The years of the breeding cycle (Cycle) planted plots per year (Plots), annual costs, in dollars, relative gains related to the CONV strategy, under presence and absence of genotype-by-environment interaction (GE), and the Cost efficiency, considering relative gain, relative costs, and relative cycles.

Scen	Cycle	Plots	Cost	Relative Gain		Cost Efficiency	
	Years	Total	Annual ($)	Absence of GE	Presence of GE	Absence of GE	Presence of GE
CONV	5	3970	$49,640	1.00	1.00	1.00	1.00
GS_CONV	5	4134	$87,124	0.93	1.01	0.53	0.57
DH_CONV	4	5649	$120,000	1.00	1.05	0.52	0.54
DH_GS	4	6449	$125,852	0.99	1.04	0.49	0.51
DH_FAST	3	4309	$99,628	1.11	1.21	0.92	1.01
DH_FAST.GS	3	5109	$112,428	1.10	1.17	0.81	0.86

The GS implementation brings an increasing on the number of plots due to the training set population, which add 750 plots to the breeding program. And, as shown by the accuracy, the GS models presented an average accuracy below 0.5 for almost the whole breeding program length (Supplementary Material - Figure S6.1 and S6.2). As higher the cost efficiency better, in this way, it can be observed a higher cost efficiency for *DH_FAST*, under presence of G×E. In another hand, all other scenarios demonstrated a poor efficiency, when considering cycle, costs, and gains. Even though they presented higher relative gains, the costs and cycles are not decreasing enough to increase the *Gain efficiency*.

4.5 Discussion

Based on the resultus, we can highlight three relevant outputs: i) the importance of investigate the best strategy to update the parental population; ii) the significance of understanding how genomic selection and double-haploid technologies affect the genetic gain and diversity of the released lines and hybrids of the sweet corn breeding program; iii) the impact of understanding market development of plant breeding, based on cost efficiency of the breeding program as well as the market share, time of releasing materials and having a recognized and competitive brand name.

Firstly, the superiority performance of discrete recurrent selection over the partial strategy is expected due to the updating of all individuals every generation by using only the most recent generation, which also brings the inbreeding rates down and increase the genetic gain over years. There is a genetic loss of using individuals from previous generations as parents mainly because inbreeding rate increasing, while the genetic values are not the issue (Meuwissen and Sonesson, 1998). Also, a simulation study demonstrated that the average age of parental individuals did not change substantially between partial and discrete strategy, suggested that the parents with highest EBVs generally come from the most recent generations. By avoiding the propagation of selection errors, as the parental individuals are selected every generation, the discrete strategy would prevent the breeder of making incorrect selections, or, as they call it: *"the same old mistakes"* (Labroo and Rutkoski, 2021).

To respond primary questions about the efficiency of a method or a strategy, the absence of the G×E interaction is an option (Cowling et al., 2020; Bančič et al., 2021). In another hand, there are several studies to investigate better analysis and methodologies including G×E interaction to increase the accuracy of GS models (Crossa et al., 2006, 2017; Heslot et al., 2012; Jarquín et al., 2016). We aimed to understand the behavior of the strategies in absence of G×E, and posteriorly, we

observed it under the presence a strong G×E (eight times the additive genetic effect magnitude), which is the reality for most of the quantitative traits under multi-trail experiments. As expected, the CONV strategy had a good performance, even compared to the other strategies with GS_CONV, DH_CONV and FAST_DH. However, under presence of G×E the performances changed, and the inclusion of these tools started to overcome the CONV strategy. This fact is based on the decreasing of plots on field under the effects of G×E bias, where the high G×E will influence on the selection, also, the increasing of accuracy based on the genomic analyses will guarantee the selection of the better materials based on their real genetic value instead of the phenotypic value. Gaynor et al. (2017) also encountered a decreased gain over years under presence of G×E (adopting 10 times the VarA), confirming the bias of this effect on the selection.

The adoption of biotechnological tools brings gains in different perspectives of a breeding program. In terms of time, using of genomic selection enable the program to cut off some stages, and shorten the pipeline (Heffner et al., 2009), what it is also a benefit of the doubled haploid technology, which can generate a homozygous line within two years (Geiger and Gordillo, 2009). In terms of costs, both tools are improving in their processes to decrease the costs, the genotyping costs are very low compared to when it was proposed in early 2000's.

The goals of the *SweetBP* are focused in optimize three main points: (i) generating of top performance lines; (ii) releasing these lines in a short time; and, (iii) having a sustainable loss of genetic variance of the crossing block (or parental lines).

The evaluated scenarios brought two different ways to optimize the breeding program performance (Figure 3). Even though the *CONV* and *GS_CONV* strategies had fewer plots planted per year, they had to develop self-pollination (*self*) and make

selection until test the genotypes with tester to evaluate the general combining ability (*GCA*) of them with elite lines, on the testcrosses (*TC*). In this way, the implementation of doubled haploid technology (*DH*) brought a reduction on the amount of *self's*, and on time, consequently, reducing laboring and shortening the breeding program length to four years. However, these strategies (*DH_CONV* and *DH_GS*) increased the number of planted plots in at least 40% comparing to the *CONV* strategy. The second way of optimization was combined to the *DH* technique. The *FAST_DH* and *FAST_DH.GS* strategies skipped the *TC1* stage, evaluating the *DH* lines' *GCA* already on *TC2* stage, which reduced another year in the *SweetBP*. Regarding to the plots, the *FAST_DH* increased 10% the number of plots of the *CONV* strategy.

Gaynor et al. (2017) reported the superiority of GS selection over a conventional pipeline, and we could not observe the same in this paper. The probable reasons for this low accuracy and poor performance of GS can be due to the training set population. Aiming to compare different strategies, we set a training set that would be performed in both CONV and DH strategies, where the size (Windhausen et al., 2012; Zhao et al., 2012; Hickey et al., 2014) and relatedness (Akdemir et al., 2015; Isidro et al., 2015) are key points on the accuracy and performance of the GS. Following that, the size of the breeding program could be further investigated, where the field experimentation starts to be unfeasible, the GS becomes a very good alternative to investigate all possible materials without needing to plant. The last reason, but not least important, can be due to the stage of using GS, aiming to predict next generation in the target environment. In many studies, the GS is used to predict hybrid performances (Schrag et al., 2019; Zystro et al., 2021), however this practice was not possible in our case due to privacy terms of the testers' genetic materials. Further investigation could

be done to make new agreements with the companies in terms of privacy of their materials, aiming to improve the using of GS in the pipeline.

Because of the cycle of 3 years, instead of 5 as the CONV, in 15 years of breeding the FAST_ strategies released 66% more lines than the CONV. The investment returns in time and market share, as discussed, being even more efficient under presence of GxE. And, if we consider the applicability of the spontaneous doubling in the next few years, the efficiency increases even more, and the adoption of FAST_DH.GS also overcomes 1, where the development and maintenance of the homozygous line of the program would be facilitate. The number of self-pollination, which is very laborious, would decrease dramatically after the first year (generation of DH and multiplication of the DH seeds). Also, in terms of market, the competitiveness of a breeding program which is often generating hybrids and lines, is higher, and the market share of the breeding program tends to increase.

Regarding to the DH costs, it depends in several factors. Our results demonstrated the potential of cost efficiency of DH, where anything over 1 means a superiority over the current conventional breeding strategy cost efficiency. There is a high potentiality of the FAST_DH strategies, decreasing the number of fields and seasons, releasing lines faster, being more present and competitive in the market, and presenting a high-cost efficiency. An alternative method for the current haploid doubling, is the using of the spontaneous haploid genome doubling (Boerman et al., 2020), which could decrease the DH production cost around 40% (Supplementary Material - Table S.2). Using this strategy could bring the FAST_DH and the FAST_DH.GS even higher in terms of cost efficiency, even increasing 8-28% of field plots and up to 75% of the costs, the cycle would be completed in three years instead of five. The genetic gain is an empirical process, instead of 12 released lines in 15

years, the program would release 20, increasing the market share and representativeness.

This study presented the potentiality of including new tools to a *SweetBP*. It was showed the increasing in genetic gain of future generated hybrids in partnership with private company, as well as of the parental lines' gains. Even though there is an increasing in costs and plots number, which are key points of public programs, in terms of physical space and funded projects, the investments brought real gains in genetic means and reducing of time. Overall, looking the big picture, it could change the *SweetBP* in terms of market share, to become a more recognized institution that generates good elite inbreeds and, consequently, better hybrids faster. Also, the new technologies bring new opportunities in terms of focus in other problems, by investigating deeper questions of the breeding program instead of having to make tons of self-pollinations and selection every season to advance the populations.

There are some outputs we could take from this project. There were some challenges that could be further investigated, but were not the focus of this work. The genomic model developed by using different models (including a investigation of mixed models and Bayes models for instance), using historical empirical dataset as training population instead of a large F1 population every cycle, and/or, applying the GS in a different stage, as in the testcrosses. As strength points, the possibility of generate genomic models and a double haploid population on the AlphaSimR is a very reliable and trustfull package, based on a robust simulation model. As future perspectives, there is a possibility to implement more traits and work on a multi-trait breeding program.

4.6 Conclusion

The adoption of discrete strategy on the parental population update was the best option to increase the parental genetic mean of the breeding program. The using of DH presented a high potential in terms of logistic to a breeding program, decreasing the number of plots when using GS, or improving the logistic of the process. And, the using of GS in any strategy helped to decrease the number of plots and shortening the cycle, increasing the gains per unit of time. The costs demonstrated that the genetic results are important, but a breeding program must have a broad view and consider all areas to make the pipeline more efficient and feasible. The best decision has to consider the sustainability in terms of genetic gain, logistic of the process, budget and market goal of the company.

4.7 References

Akdemir, D., Sanchez, J. I., and Jannink, J. (2015). Optimization of genomic selection training populations with a genetic algorithm. *Genet. Sel. Evol.* 47, 1–10. doi: 10.1186/s12711-015-0116-6.

Ali, M., Zhang, L., DeLacy, I., Arief, V., Dieters, M., Pfeiffer, W. H., et al. (2020). Modeling and simulation of recurrent phenotypic and genomic selections in plant breeding under the presence of epistasis. *Crop J.* 8, 866–877. doi: 10.1016/j.cj.2020.04.002.

Allard, R. W. (1971). *Principles of plant breeding*. São Paulo - SP: Edgard Blücher.

Andorf, C., Beavis, W. D., Hufford, M., Smith, S., Suza, W. P., Wang, K., et al. (2019). Technological advances in maize breeding: past, present and future. *Theor. Appl. Genet.* 132, 817–849. doi: 10.1007/s00122-019-03306-3.

Baenziger, P. S. (1996). “Reflections on doubled haploids in plant breeding,” in *In Vitro Haploid Production in Higher Plants*, 35–48. doi: 10.1007/978-94-017-

1860-8_3.

Bančič, J., Werner, C. R., Gaynor, R. C., Gorjanc, G., Odeny, D. A., Ojulong, H. F., et al. (2021). Modeling Illustrates That Genomic Selection Provides New Opportunities for Intercrop Breeding. *Front. Plant Sci.* 12. doi: 10.3389/fpls.2021.605172.

Barnabás, B., Obert, B., and Kovács, G. (1999). Colchicine, an efficient genome-doubling agent for maize (Zea mays L.) microspores cultured in anthero. *Plant Cell Rep.* 18, 858–862. doi: 10.1007/s002990050674.

Bassi, F. M., Bentley, A. R., Charmet, G., Ortiz, R., and Crossa, J. (2016). Breeding schemes for the implementation of genomic selection in wheat (Triticum spp .). *Plant Sci.* 242, 23–36. doi: 10.1016/j.plantsci.2015.08.021.

Batista, L. G., Gaynor, R. C., Margarido, G. R. A., Byrne, T., Amer, P., Gorjanc, G., et al. (2021). Long-term comparison between index selection and optimal independent culling in plant breeding programs with genomic prediction. *PLoS One* 16, 1–5. doi: 10.1371/journal.pone.0235554.

Bernardo, R. (2008). Molecular markers and selection for complex traits in plants: Learning from the last 20 years. *Crop Sci.* 48, 1649–1664. doi: 10.2135/cropsci2008.03.0131.

Bernardo, R. (2020). Reinventing quantitative genetics for plant breeding: something old, something new, something borrowed, something BLUE. *Heredity (Edinb).* 125, 375–385. doi: 10.1038/s41437-020-0312-1.

Boddhireddy, P., Kelly, M. J., Northcutt, S., Prayaga, K. C., Rumph, J., and DeNise, S. (2014). Genomic predictions in Angus cattle: Comparisons of sample size, response variables, and clustering methods for cross-validation1. *J. Anim. Sci.* 92, 485–497. doi: 10.2527/jas.2013-6757.

Boerman, N. A., Frei, U. K., and Lübberstedt, T. (2020). Impact of Spontaneous Haploid Genome Doubling in Maize Breeding. *Plants* 9, 369. doi: 10.3390/plants9030369.

Bouchez, A., and Gallais, A. (2000). Efficiency of the Use of Doubled-Haploids in Recurrent Selection for Combining Ability. *Crop Sci.* 40, 23–29.

Broman, K. W., and Weber, J. L. (2000). Characterization of Human Crossover Interference. *Am. J. Hum. Genet.* 66, 1911–1926. doi: 10.1086/302923.

Chaikam, V., Molenaar, W., Melchinger, A. E., and Boddupalli, P. M. (2019). Doubled haploid technology for line development in maize: technical advances and prospects. *Theor. Appl. Genet.* 132, 3227–3243. doi: 10.1007/s00122-019-03433-x.

Chase, S. S. (1949). MONOPLOID FREQUENCIES IN A COMMERCIAL DOUBLE CROSS HYBRID MAIZE, AND IN ITS COMPONENT SINGLE CROSS HYBRIDS AND INBRED LINES. *Genetics* 34, 328–332. doi: 10.1093/genetics/34.3.328.

Chase, S. S. (1952). Production of Homozygous Diploids of Maize from Monoploids 1. *Agron. J.* 44, 263–267. doi: 10.2134/agronj1952.00021962004400050010x.

Chen, G. K., Marjoram, P., and Wall, J. D. (2009). Fast and flexible simulation of DNA sequence data. *Genome Res.* 19, 136–142. doi: 10.1101/gr.083634.108.

Coe, E. H., and Sarkar, K. R. (1964). The detection of haploids in maize. *J. Hered.* 55, 231–233. doi: 10.1093/oxfordjournals.jhered.a107340.

Cooper, M., Podlich, D. W., Jensen, N. M., Chapman, S. C., and Hammer, G. L. (1999). Modeling Plant Breeding Programs. *Trends Agron.* 2, 33–64.

Cowling, W. A., Gaynor, R. C., Antolín, R., Gorjanc, G., Edwards, S. M., Powell, O., et al. (2020). In silico simulation of future hybrid performance to evaluate

heterotic pool formation in a self-pollinating crop. *Sci. Rep.* 10, 1–8. doi: 10.1038/s41598-020-61031-0.

Cox, T. S., Sears, R. G., Bequette, R. K., and Martin, T. J. (1995). Germplasm enhancement in winter wheat x Triticum tauschii backcross populations. *Crop Sci.* 35, 913–919. doi: 10.2135/cropsci1995.0011183X003500030047x.

Crossa, J., Burgueño, J., Cornelius, P. L., McLaren, G., Trethowan, R., and Krishnamachari, A. (2006). Modeling Genotype × Environment Interaction Using Additive Genetic Covariances of Relatives for Predicting Breeding Values of Wheat Genotypes. *Crop Sci.* 46, 1722–1733. doi: 10.2135/cropsci2005.11-0427.

Crossa, J., Pérez-Rodríguez, P., Cuevas, J., Montesinos-López, O., Jarquín, D., de los Campos, G., et al. (2017). Genomic Selection in Plant Breeding: Methods, Models, and Perspectives. *Trends Plant Sci.* 22, 961–975. doi: 10.1016/j.tplants.2017.08.011.

Dekkers, J. C. M., and Hospital, F. (2002). The use of molecular genetics in the improvement of agricultural populations. *Nat. Rev. Genet.* 3, 22–32. doi: 10.1038/nrg701.

Diana, G., and Tommasi, C. (2002). Cross-validation methods in principal component analysis: A comparison. *Stat. Methods Appt.* 11, 71–82. doi: 10.1007/BF02511446.

Eder, J., and Chalyk, S. (2002). In vivo haploid induction in maize. *Theor. Appl. Genet.* 104, 703–708. doi: 10.1007/s00122-001-0773-4.

Endelman, J. B. (2011). Ridge regression and other kernels for genomic selection with R package rrBLUP. *Plant Genome* 4, 250–255.

Falconer, D. S., and Mackay, T. F. C. (1996). Introduction to quantitative genetics. *Harlow, Essex, UK Longmans Green* 3, 280.

Faux, A., Gorjanc, G., Gaynor, R. C., Battagin, M., Edwards, S. M., Wilson, D. L., et al. (2016). AlphaSim: Software for Breeding Program Simulation. *Plant Genome* 9, plantgenome2016-02. doi: 10.3835/plantgenome2016.02.0013.

Gaynor, R. C., Gorjanc, G., Bentley, A. R., Ober, E. S., Howell, P., Jackson, R., et al. (2017). A two-part strategy for using genomic selection to develop inbred lines. *Crop Sci.* 57, 2372–2386. doi: 10.2135/cropsci2016.09.0742.

Gaynor, R. C., Gorjanc, G., and Hickey, J. (2020). AlphaSimR: An R-package for Breeding Program Simulations. *G3 Genes|Genomes|Genetics*, 1–21. doi: 10.1101/2020.08.10.245167.

Geiger, H. H., and Gordillo, G. A. (2009). Doubled haploids in hybrid maize breeding. *Maydica* 54, 485–499.

Gonen, S., Wimmer, V., Gaynor, R. C., Byrne, E., Gorjanc, G., and Hickey, J. M. (2018). A heuristic method for fast and accurate phasing and imputation of single-nucleotide polymorphism data in bi-parental plant populations. *Theor. Appl. Genet.* 131, 2345–2357. doi: 10.1007/s00122-018-3156-9.

Gorjanc, G., Jenko, J., Hearne, S. J., and Hickey, J. M. (2016). Initiating maize pre-breeding programs using genomic selection to harness polygenic variation from landrace populations. *BMC Genomics* 17, 1–15. doi: 10.1186/s12864-015-2345-z.

Hallauer, A. R., Carena, J. M., and Miranda Filho, J. B. de. (2010). *Quantitative genetics in maize breeding. New York: Springer, 2010. 500 p.* New York: Springer.

Hallauer, A. R., and Darrah, L. L. (1985). Compendium of recurrent selection methods and their application. *CRC. Crit. Rev. Plant Sci.* 3, 1–33. doi: 10.1080/07352688509382202.

Häntzschel, K. R., and Weber, G. (2010). Blockage of mitosis in maize root tips using colchicine-alternatives. *Protoplasma* 241, 99–104. doi: 10.1007/s00709-009-0103-2.

Heffner, E. L., Jannink, J.-L., and Sorrells, M. E. (2011). Genomic Selection Accuracy using Multifamily Prediction Models in a Wheat Breeding Program. *Plant Genome* 4, 65–75. doi: 10.3835/plantgenome2010.12.0029.

Heffner, E. L., Lorenz, A. J., Jannink, J. L., and Sorrells, M. E. (2010). Plant breeding with Genomic selection: Gain per unit time and cost. *Crop Sci. Madison* 50, 1681–1690. doi: 10.2135/cropsci2009.11.0662.

Heffner, E. L., Sorrells, M. E., and Jannink, J. L. (2009). Genomic selection for crop improvement. *Crop Sci.* 49, 1–12. doi: 10.2135/cropsci2008.08.0512.

Heslot, N., Jannink, J.-L., and Sorrells, M. E. (2015). Perspectives for Genomic Selection Applications and Research in Plants. *Crop Sci.* 55, 1–12. doi: 10.2135/cropsci2014.03.0249.

Heslot, N., Yang, H. P., Sorrells, M. E., and Jannink, J. L. (2012). Genomic selection in plant breeding: A comparison of models. *Crop Sci.* 52, 146–160. doi: 10.2135/cropsci2011.06.0297.

Hickey, J. M., Dreisigacker, S., Crossa, J., Hearne, S., Babu, R., Prasanna, B. M., et al. (2014). Evaluation of genomic selection training population designs and genotyping strategies in plant breeding programs using simulation. *Crop Sci.* 54, 1476–1488. doi: 10.2135/cropsci2013.03.0195.

Hu, Y., Colantonio, V., Müller, B. S. F., Leach, K. A., Nanni, A., Finegan, C., et al. (2021). Genome assembly and population genomic analysis provide insights into the evolution of modern sweet corn. *Nat. Commun.* 12. doi: 10.1038/s41467-021-21380-4.

Huang, M., Robbins, K. R., Li, Y., Umanzor, S., Marty-Rivera, M., Bailey, D., et al. (2021). Simulation of sugar kelp (Saccharina latissima) breeding guided by practices to prioritize accelerated research gains. *Sci. South Russ.*, 39–44. doi: 10.1101/2021.01.21.427651.

Isidro, J., Jannink, J.-L., Akdemir, D., Poland, J., Heslot, N., and Sorrells, M. E. (2015). Training set optimization under population structure in genomic selection. *Theor. Appl. Genet.* 128, 145–158. doi: 10.1007/s00122-014-2418-4.

Jannink, J.-L., Lorenz, A. J., and Iwata, H. (2010). Genomic selection in plant breeding: from theory to practice. *Brief. Funct. Genomics* 9, 166–177. doi: 10.1093/bfgp/elq001.

Jarquín, D., Pérez-Elizalde, S., Burgueño, J., and Crossa, J. (2016). A Hierarchical Bayesian Estimation Model for Multienvironment Plant Breeding Trials in Successive Years. *Crop Sci.* 56, 2260–2276. doi: 10.2135/cropsci2015.08.0475.

Kato, A. (2002). Chromosome doubling of haploid maize seedlings using nitrous oxide gas at the flower primordial stage. *Plant Breed.* 121, 370–377. doi: 10.1046/j.1439-0523.2002.743321.x.

Kelliher, T., Starr, D., Richbourg, L., Chintamanani, S., Delzer, B., Nuccio, M. L., et al. (2017). MATRILINEAL, a sperm-specific phospholipase, triggers maize haploid induction. *Nature* 542, 105–109. doi: 10.1038/nature20827.

Labroo, M. R., and Rutkoski, J. E. (2021). New cycle , same old mistakes ? Overlapping vs . discrete generations in long-term recurrent selection. doi: 10.1101/2021.10.12.464059.

Lertrat, K., and Pulam, T. (2007). Breeding for increased sweetness in sweet corn. *Int. J. Plant Breed.* 1, 27–30.

Lima, R. O. de, and Borém, A. (2018). *Melhoramento de Milho*. Viçosa - MG: Editora

UFV.

Lin, P., Hartz, S. M., Zhang, Z., Saccone, S. F., Wang, J., Tischfield, J. A., et al. (2010). A New Statistic to Evaluate Imputation Reliability. *PLoS One* 5, e9697. doi: 10.1371/journal.pone.0009697.

Liu, C., Li, X., Meng, D., Zhong, Y., Chen, C., Dong, X., et al. (2017). A 4-bp Insertion at ZmPLA1 Encoding a Putative Phospholipase A Generates Haploid Induction in Maize. *Mol. Plant* 10, 520–522. doi: 10.1016/j.molp.2017.01.011.

Liu, Z., Ren, J., Trampe, B., Frei, U. K., and Lubberstedt, T. (2016). "Double haploid: from obscure phenomenon to key technology of current maize breeding programs," in *Plant Breeding Reviews, Vol 40*, 123–166.

Marulanda, J. J., Mi, X., Melchinger, A. E., Xu, J. L., Würschum, T., and Longin, C. F. H. (2016). Optimum breeding strategies using genomic selection for hybrid breeding in wheat, maize, rye, barley, rice and triticale. *Theor. Appl. Genet.* 129, 1901–1913. doi: 10.1007/s00122-016-2748-5.

McPeek, M. S., and Speed, T. P. (1995). Modeling Inference in Genetic Recombination. *Genetics* 139, 1031–1044. doi: 10.1007/978-1-4614-1347-9_10.

Melchinger, A. E., Geiger, H. H., and Schnell, F. W. (1986). Epistasis in maize (Zea mays L.) - 2. Genetic effects in crosses among early flint and dent inbred lines determined by three methods. *Theor. Appl. Genet.* 72, 231–239. doi: 10.1007/BF00266997.

Melchinger, A. E., Longin, C., Utz, H., and Reif, J. (2005). Hybrid maize breeding with doubled haploid lines: quantitative genetic and selection theory for optimum allocation of resources. in *Forty first annual Illinois corn breeders' school*, 8–21.

Meuwissen, T. H. . E., Hayes, B. J., and Goddard, M. E. (2001). Prediction of total genetic value using genome-wide dense marker maps. *Genetics* 157, 1819–

1829. doi: 11290733.

Meuwissen, T. H. E., and Sonesson, A. K. (1998). Maximizing the Response of Selection with a Predefined Rate of Inbreeding: Overlapping Generations. *J. Anim. Sci.* 76, 2575–2583. doi: 10.2527/1998.76102575x.

Moore, F. B. G., and Tonsor, S. J. (1994). A simulation of Wright's shifting-balance process: migration and the three phases. *Evolution (N. Y).* 48, 69–80.

Muleta, K. T., Pressoir, G., and Morris, G. P. (2019). Optimizing genomic selection for a sorghum breeding program in Haiti: A simulation study. *G3 Genes, Genomes, Genet.* 9, 391–401. doi: 10.1534/g3.118.200932.

Podlich, D. W., and Cooper, M. (1998). QU-GENE: a simulation platform for quantitative analysis of genetic models. *Bioinformatics* 14, 632–653. doi: 10.1093/bioinformatics/14.7.632.

Podlich, D. W., Cooper, M., Basford, K. E., and Geiger, H. H. (1999). Computer simulation of a selection strategy to accommodate genotype environment interactions in a wheat recurrent selection programme. *Plant Breed.* 118, 17–28. doi: 10.1046/j.1439-0523.1999.118001017.x.

Popovici, V., Chen, W., Gallas, B. D., Hatzis, C., Shi, W., Samuelson, F. W., et al. (2010). Effect of training-sample size and classification difficulty on the accuracy of genomic predictors. *Breast Cancer Res.* 12, R5. doi: 10.1186/bcr2468.

R Development Core Team (2020). R: A language and environment for statistical computing. Available at: https://www.r-project.org/.

Ren, J., Wu, P., Tian, X., Lübberstedt, T., and Chen, S. (2017). QTL mapping for haploid male fertility by a segregation distortion method and fine mapping of a key QTL qhmf4 in maize. *Theor. Appl. Genet.* 130, 1349–1359. doi: 10.1007/s00122-017-2892-6.

Revilla, P., Anibas, C. M., and Tracy, W. F. (2021). Sweet corn research around the world 2015–2020. *Agronomy* 11, 1–49. doi: 10.3390/agronomy11030534.

Rotarenco, V., Dicu, G., and Fuia, S. (2010). New inducers of maternal haploids in maize. *Maize Genet. Coop. Newsl.*, 21–22.

Sabadin, F., Dovale, J. C., Platten, J., and Fritsche-Neto, R. (2021). Optimizing self-pollinated crop breeding employing genomic selection: from schemes to updating training sets.

Schrag, T. A., Schipprack, W., and Melchinger, A. E. (2019). Across-years prediction of hybrid performance in maize using genomics. *Theor. Appl. Genet.* 132, 933–946. doi: 10.1007/s00122-018-3249-5.

Shull, G. H. (1908). The composition of a field of maize. *J. Hered.* 4, 296–301.

Smith, J. S. C., Hussain, T., Jones, E. S., Graham, G., Podlich, D., Wall, S., et al. (2008). Use of doubled haploids in maize breeding: implications for intellectual property protection and genetic diversity in hybrid crops. *Mol. Breed.* 22, 51–59. doi: 10.1007/s11032-007-9155-1.

Tanksley, S. D., Young, N. D., Paterson, A. H., and Bonierbale, M. W. (1989). RFLP Mapping in Plant Breeding: New Tools for an Old Science. *Nat. Biotechnol.* 7, 257–264. doi: 10.1038/nbt0389-257.

Tessema, B. B., Liu, H., Sørensen, A. C., Andersen, J. R., and Jensen, J. (2020). Strategies Using Genomic Selection to Increase Genetic Gain in Breeding Programs for Wheat. *Front. Genet.* 11, 1–12. doi: 10.3389/fgene.2020.578123.

Tracy, W. F. (1993). “Sweet corn,” in *Genetic Improvement of Vegetable Crops* (Boston, MA: Elsevier), 777–807. doi: 10.1016/B978-0-08-040826-2.50058-8.

Tracy, W. F. (2001). “Sweet corn,” in *Specialty Corns*, ed. A. R. Hallauer (Boca Raton, FL: Elsevier), 777–807.

Troyer, A. F., and Wellin, E. J. (2009). Heterosis Decreasing in Hybrids: Yield Test Inbreds. *Crop Sci.* 49, 1969–1976. doi: 10.2135/cropsci2009.04.0170.

van Berloo, R., and Stam, P. (1998). Marker-assisted selection in autogamous RIL populations: a simulation study. *Theor. Appl. Genet.* 96, 147–154.

Verges, V. L., and van Sanford, D. A. (2020). Genomic selection at preliminary yield trial stage: Training population design to predict untested lines. *Agronomy* 10, 1–16. doi: 10.3390/agronomy10010060.

Verma, S. S., de Andrade, M., Tromp, G., Kuivaniemi, H., Pugh, E., Namjou-Khales, B., et al. (2014). Imputation and quality control steps for combining multiple genome-wide datasets. *Front. Genet.* 5. doi: 10.3389/fgene.2014.00370.

Wang, J., Ginkel, M., Podlich, D., Ye, G., Trethowan, R., Pfeiffer, W., et al. (2003). Comparison of Two Breeding Strategies by Computer Simulation. *Crop Sci.* 43, 1764–1773. doi: 10.2135/cropsci2003.1764.

Wang, X., Yang, Z., and Xu, C. (2015). A comparison of genomic selection methods for breeding value prediction. *Sci. Bull.* 60, 925–935. doi: 10.1007/s11434-015-0791-2.

Wetterstrand, K. A. (2020). The Cost of Sequencing a Human Genome. *Cost Seq. a Hum. Genome*.

Windhausen, V. S., Atlin, G. N., Hickey, J. M., Crossa, J., Jannink, J. L., Sorrells, M. E., et al. (2012). Effectiveness of Genomic Prediction of Maize Hybrid Performance in Different Breeding Populations and Environments. *G3 Genes|Genomes|Genetics* 2, 1427–1436. doi: 10.1534/g3.112.003699.

Würschum, T., Reif, J. C., Kraft, T., Janssen, G., and Zhao, Y. (2013). Genomic selection in sugar beet breeding populations. *BMC Genet.* 14, 85. doi: 10.1186/1471-2156-14-85.

Zhao, Y., Gowda, M., Liu, W., Würschum, T., Maurer, H. P., Longin, F. H., et al. (2012). Accuracy of genomic selection in European maize elite breeding populations. *Theor. Appl. Genet.* 124, 769–776. doi: 10.1007/s00122-011-1745-y.

Zystro, J., Peters, T., Miller, K., and Tracy, W. F. (2021). Classical and genomic prediction of hybrid sweet corn performance in organic environments. *Crop Sci.* 61, 1698–1708. doi: 10.1002/csc2.20400.

4.8 Supplementary Material

4.8.1 Tables and Figures

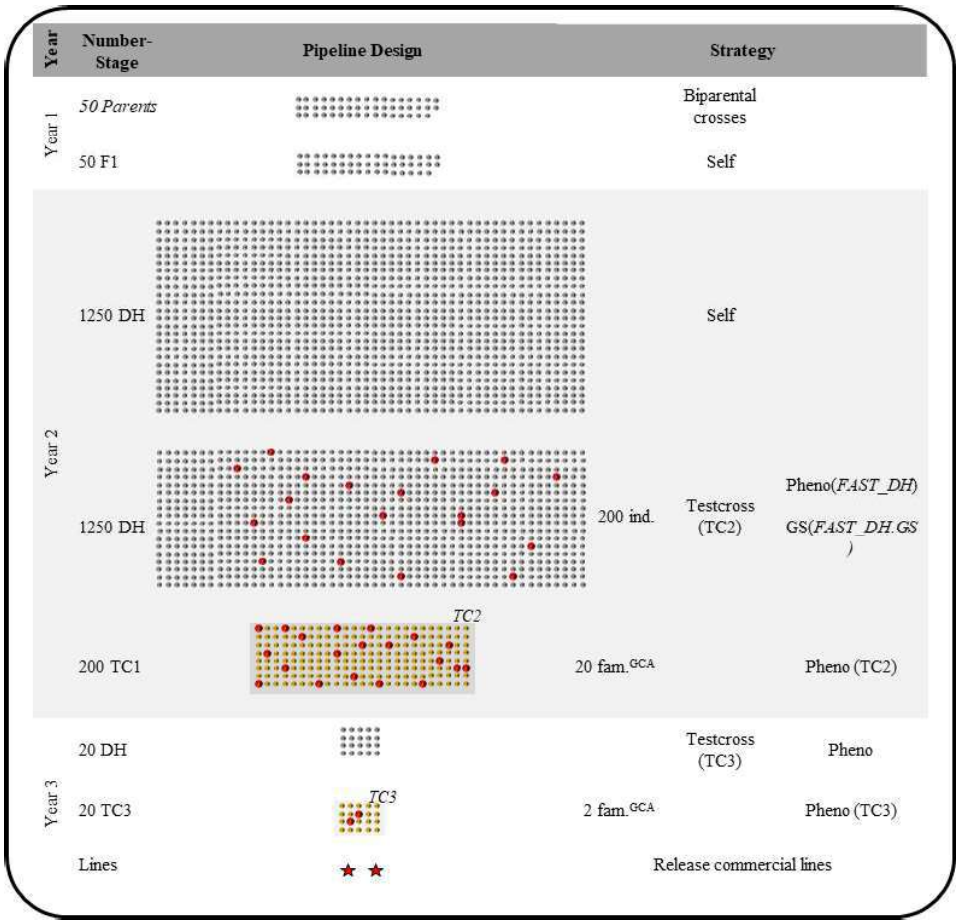

Figure S1 - Sweet corn FAST_DH and FAST_DH.GS breeding program pipelines. Detailed explanation of each stage dynamics. The pipeline whole cycle completes in 4 years to generate two new elite lines. Every 2 years there is an update of the Parental lines. The testcross 2 (TC2) stage is developed in in 5 environments and testcross 3 (TC3) stage in 20 different environments, using respectively 2 and 5 testers. *Selection of individiduals (ind), and families based on general combining ability (GCA); ** Crossing: self-pollination (self); and ***Evaluation based on Phenotypic values (Pheno) and Esimated breeding Values (GS). FAST_DH: Doubled haploid strategy skipping the

testcross 1 stage; conventional, FAST_DH.GS: FAST_DH strategy using genomic selection.

Table S1: Parental parameters of the tested strategies which applied Discrete and Partial Recurrent Selection Strategies, and the comparison of Discrete over Partial strategy. The tested strategies applying genomic selection were conventional (CONV), conventional using genomic selection (GS_*CONV*), and the conventional double haploid strategy (*DH_CONV*). All strategies were analysed under presence and absence of genotype-by-environment interaction (G×E).

Parental Parameters	Strategy	Absence of G×E		Presence of G×E	
		Partial	Discrete	Partial	Discrete
Genetic Mean	CONV	210.38	217.43	163.23	194.26
	GS_CONV	208.61	214.45	165.39	196.19
	DH_CONV	198.66	203.32	169.50	182.89
Genetic Variance	CONV	17.92	5.56	27.96	5.77
	GS_CONV	18.23	5.35	22.94	6.37
	DH_CONV	10.65	7.07	17.36	9.39

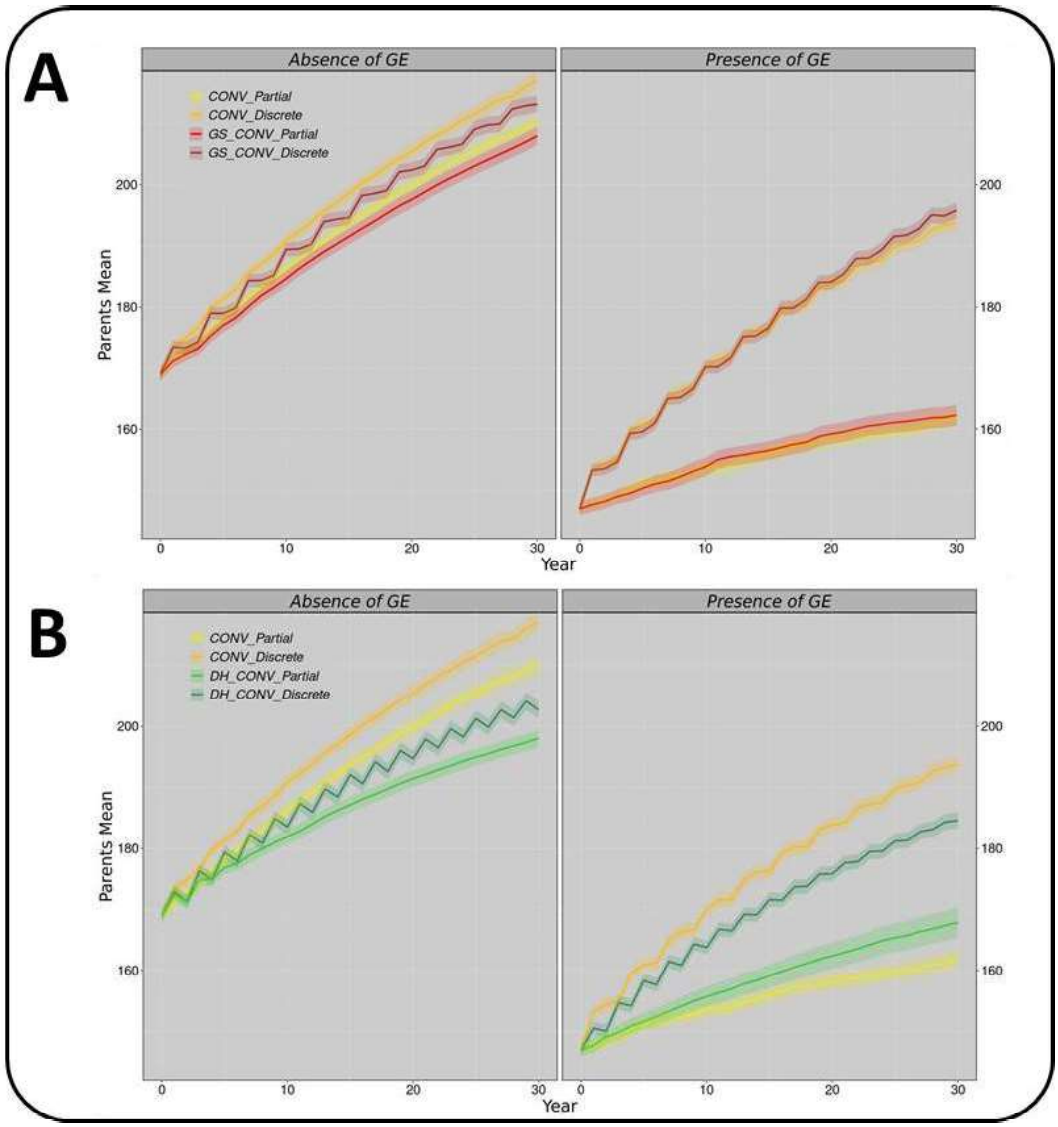

Figure S2.1 – Parental genetic mean over 30 years of breeding program, over presence and absence of genotype-by-interaction effect (GE). The conventional adopting Partial Recurrent Selection (CONV_Partial) and Discrete Recurrent Selection (CONV_Discrete) strategies, compared with the genomic selection strategy adopting Partial Recurrent Selection (GS_Partial) and Discrete Recurrent Selection (GS_Discrete) strategies, and double haploids breeding program adopting Partial Recurrent Selection (DH_Partial) and Discrete Recurrent Selection (DH_Discrete) strategies

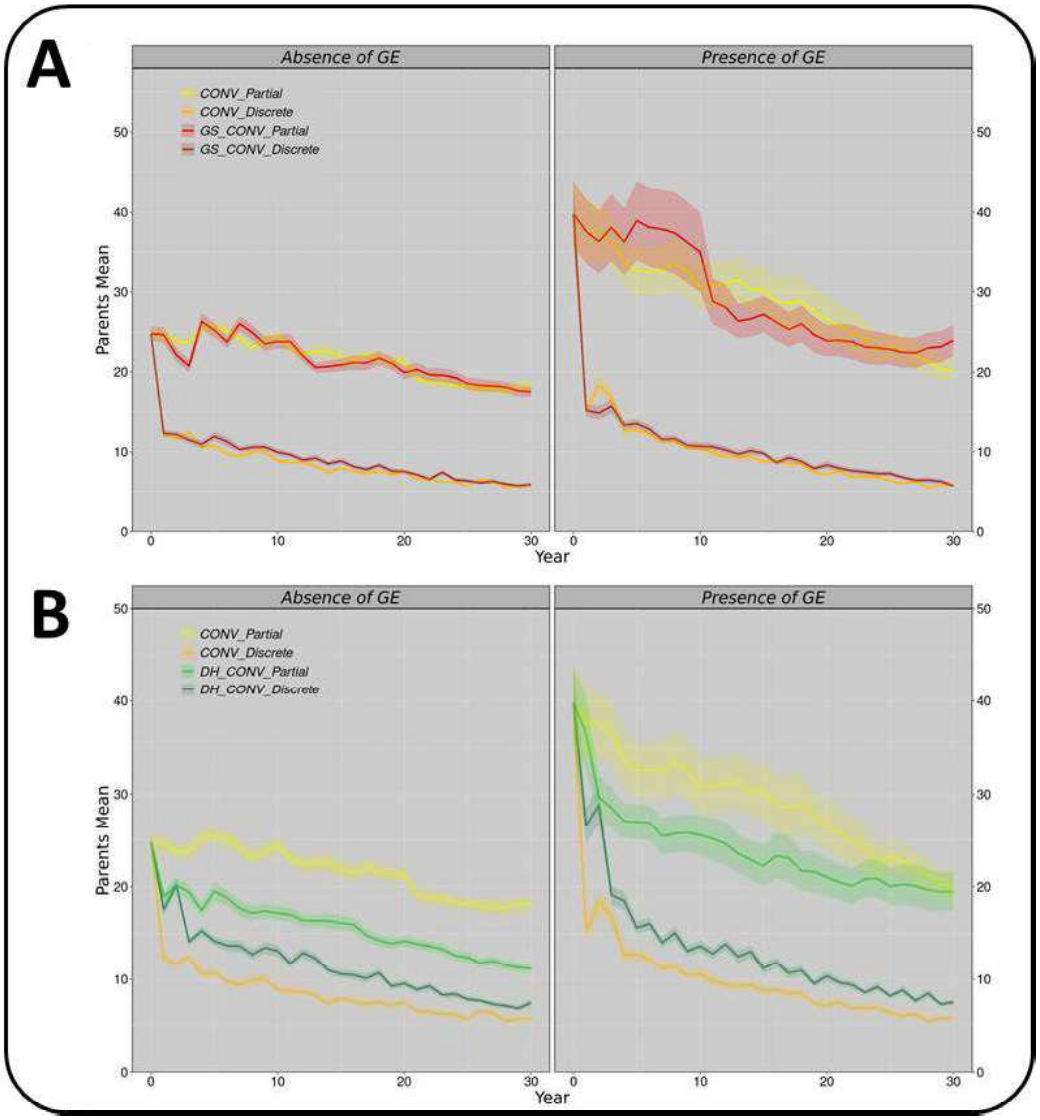

Figure S2.2 – Parental genetic variance over 30 years of breeding program, over presence and absence of genotype-by-interaction effect (GE). The conventional adopting Partial Recurrent Selection (CONV_Partial) and Discrete Recurrent Selection (CONV_Discrete) strategies, compared with the genomic selection strategy adopting Partial Recurrent Selection (GS_Partial) and Discrete Recurrent Selection (GS_Discrete) strategies, and double haploids breeding program adopting Partial Recurrent Selection (DH_Partial) and Discrete Recurrent Selection (DH_Discrete) strategies

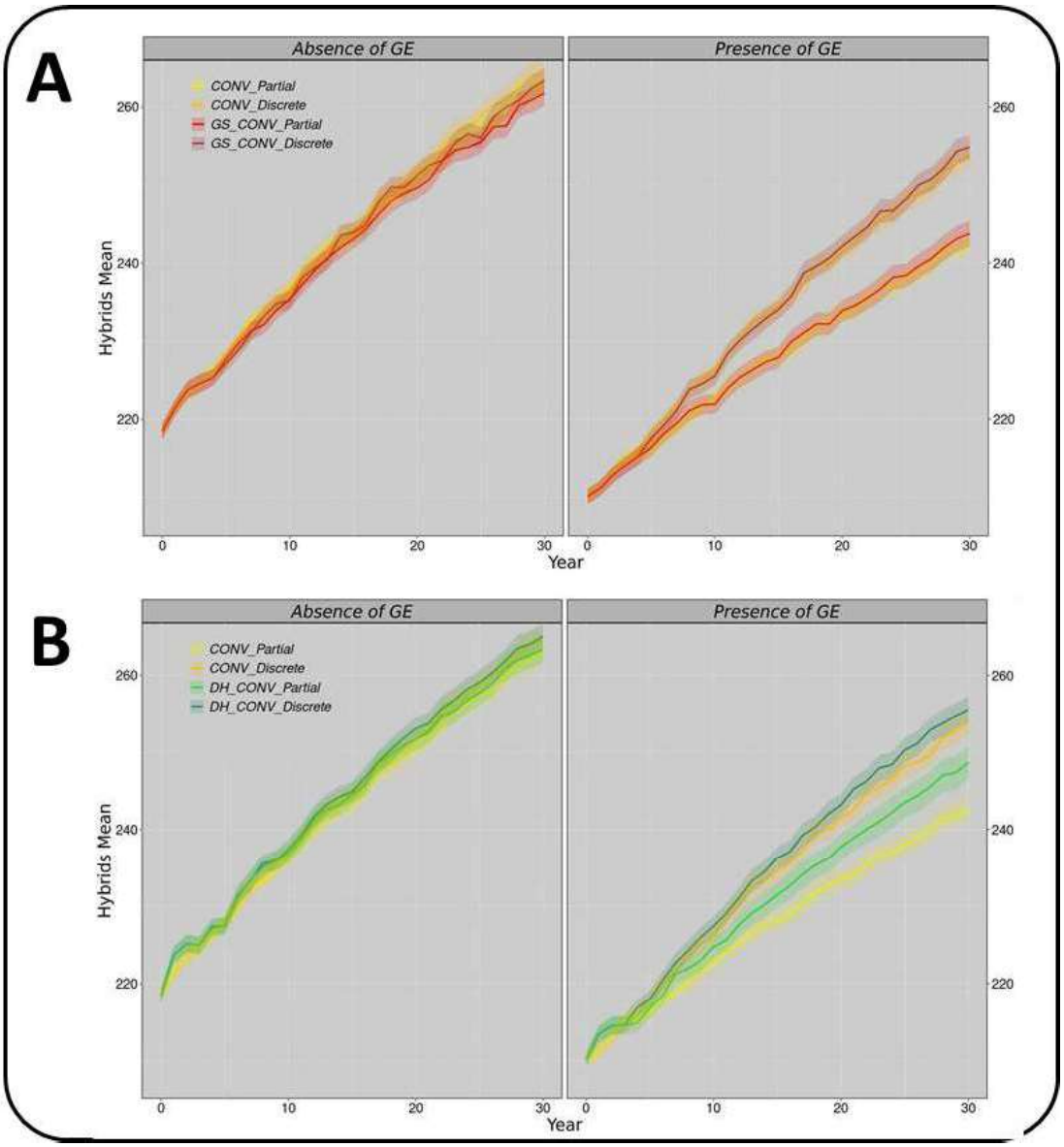

Figure S2.3 – Hybrid genetic mean over 30 years of breeding program, over presence and absence of genotype-by-interaction effect (GE). The conventional adopting Partial Recurrent Selection (CONV_Partial) and Discrete Recurrent Selection (CONV_Discrete) strategies, compared with the genomic selection strategy adopting Partial Recurrent Selection (GS_Partial) and Discrete Recurrent Selection (GS_Discrete) strategies, and double haploids breeding program adopting Partial Recurrent Selection (DH_Partial) and Discrete Recurrent Selection (DH_Discrete) strategies

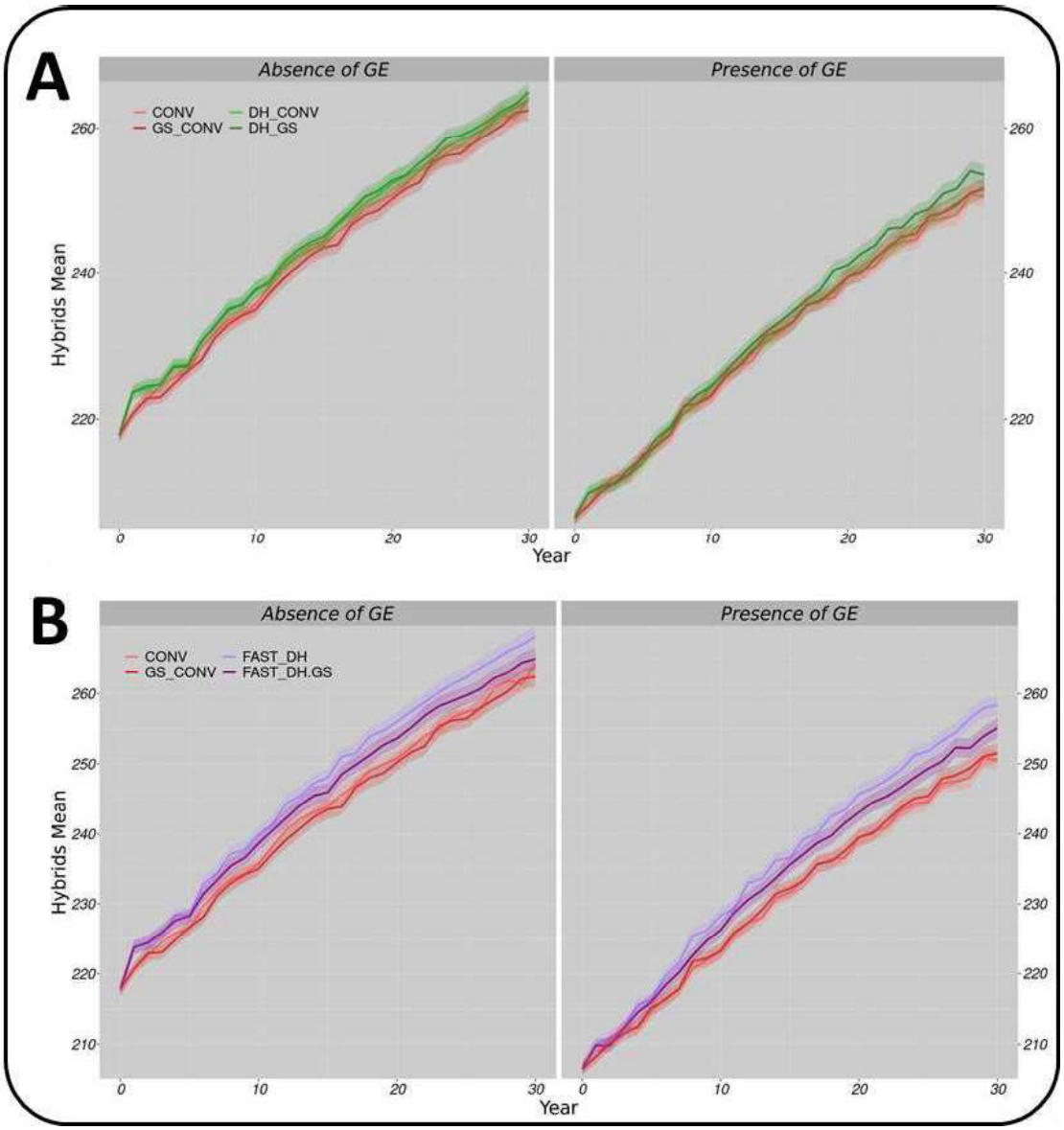

Figure S3.1 – Hybrid genetic mean over 30 years of breeding program, over presence and absence of genotype-by-interaction effect (GE). The conventional (CONV) and conventional using genomic selection (GS) (GS_CONV) strategies were compared to the double haploids (DH) conventional and GS (DH_CONV and DH_GS, respectively) (A) and with the DH adopting only two testcross' stages (FAST_DH) and applying genomic (FAST_DH.GS) (B).

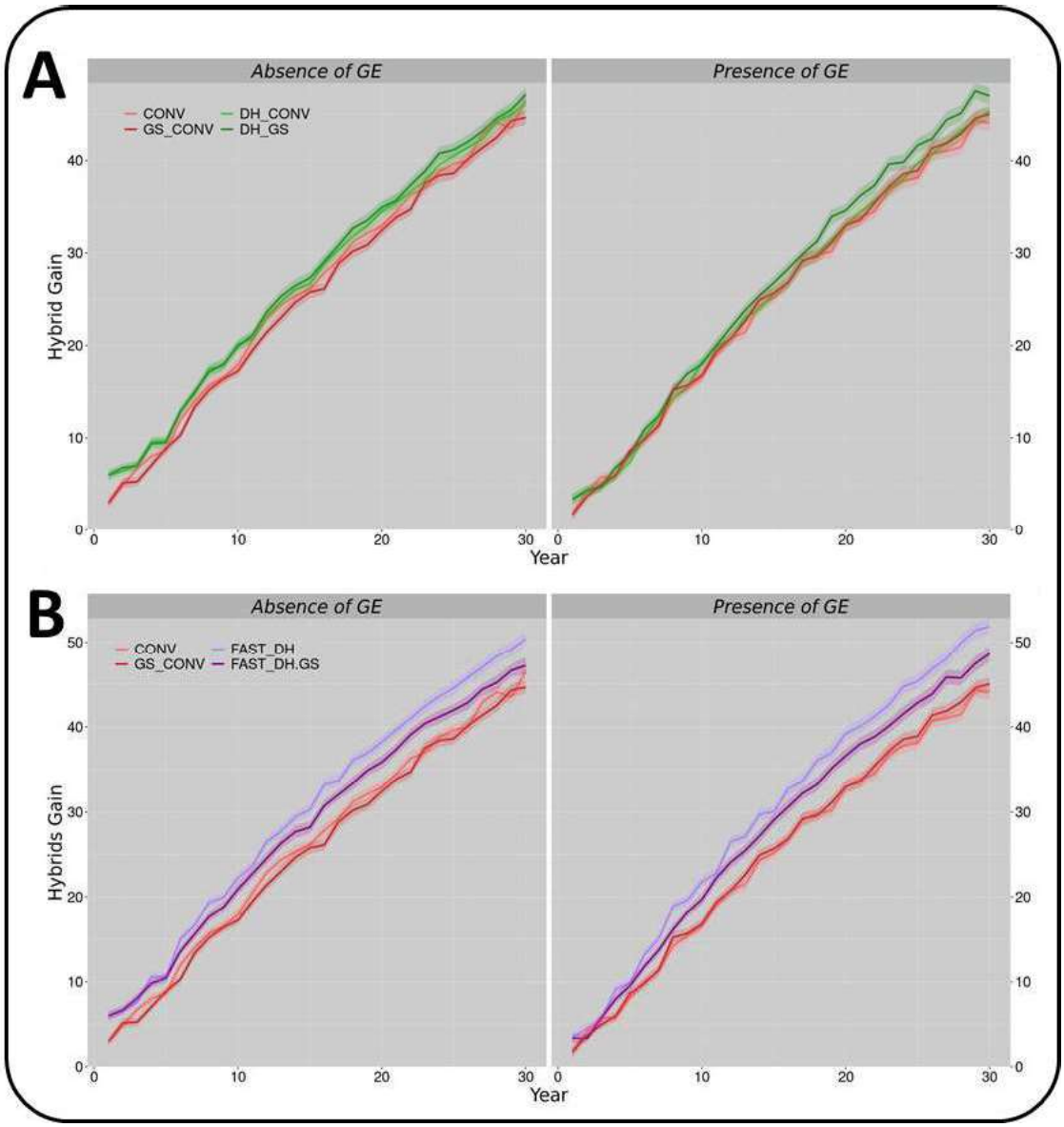

Figure S3.2– Hybrid genetic gain over 30 years of breeding program, over presence and absence of genotype-by-interaction effect (GE). The conventional (CONV) and conventional using genomic selection (GS) (GS_CONV) strategies were compared to the double haploids (DH) conventional and GS (DH_CONV and DH_GS, respectively) (A) and with the DH adopting only two testcross' stages (FAST_DH) and applying genomic (FAST_DH.GS) (B).

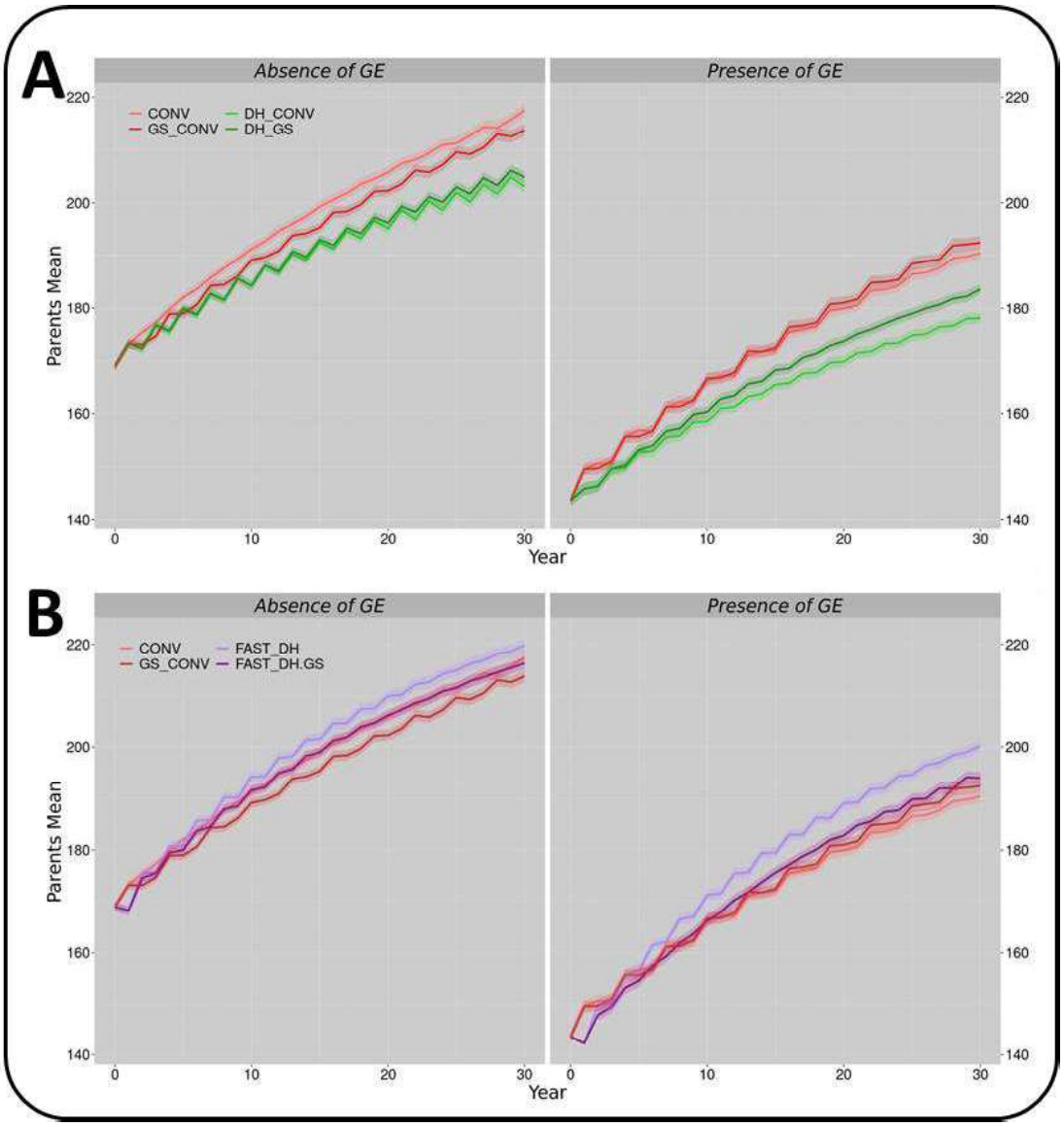

Figure S4.1– Parental (crossing block) genetic mean over 30 years of breeding program, over presence and absence of genotype-by-interaction effect (GE). The conventional (CONV) and conventional using genomic selection (GS) (GS_CONV) strategies were compared to the double haploids (DH) conventional and GS (DH_CONV and DH_GS, respectively) (A) and with the DH adopting only two testcross' stages (FAST_DH) and applying genomic (FAST_DH.GS) (B).

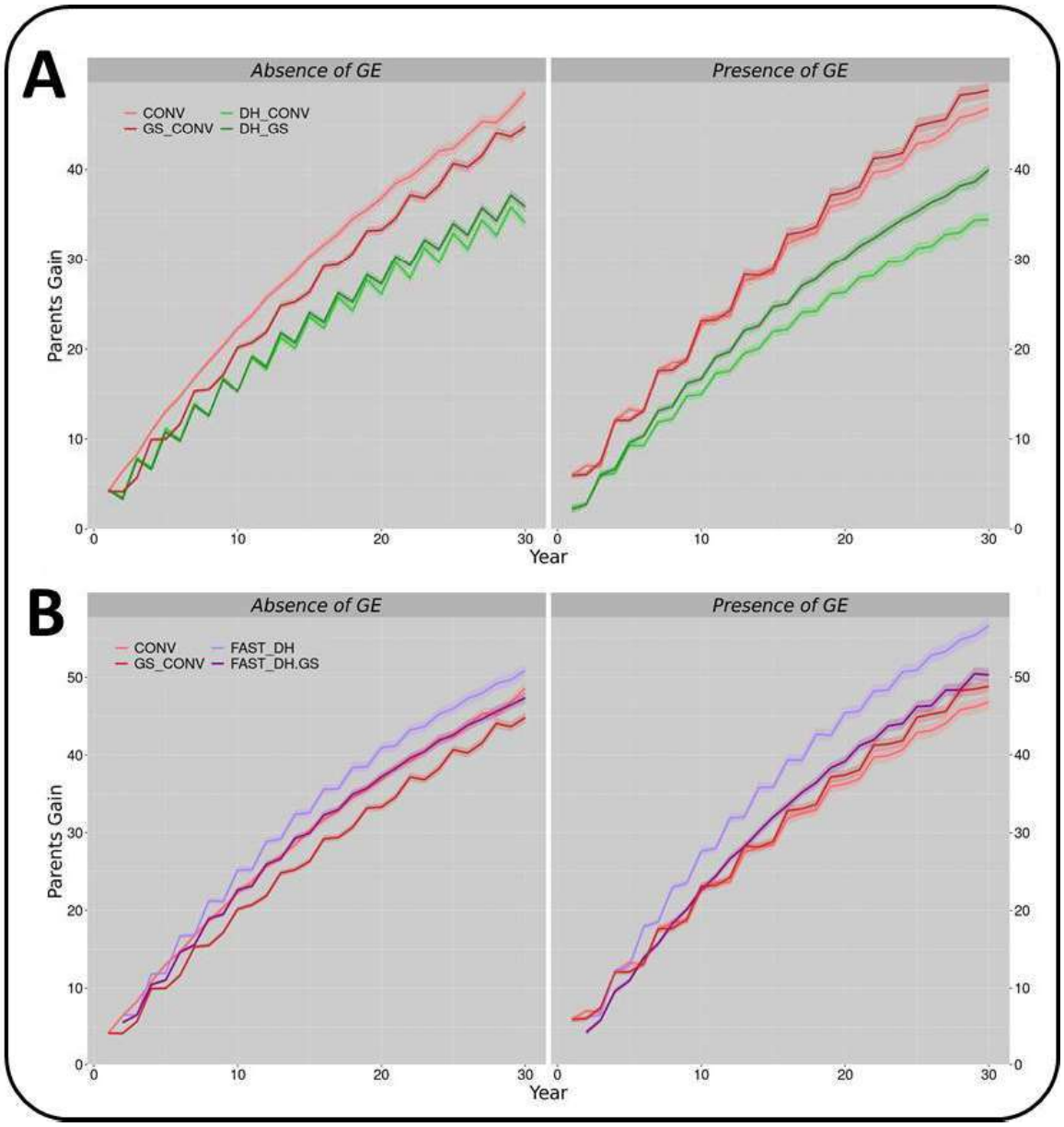

Figure S4.2 – Parental (crossing block) genetic gain over 30 years of breeding program, over presence and absence of genotype-by-interaction effect (GE). The conventional (CONV) and conventional using genomic selection (GS) (GS_CONV) strategies were compared to the double haploids (DH) conventional and GS (DH_CONV and DH_GS, respectively) (A) and with the DH adopting only two testcross' stages (FAST_DH) and applying genomic (FAST_DH.GS) (B).

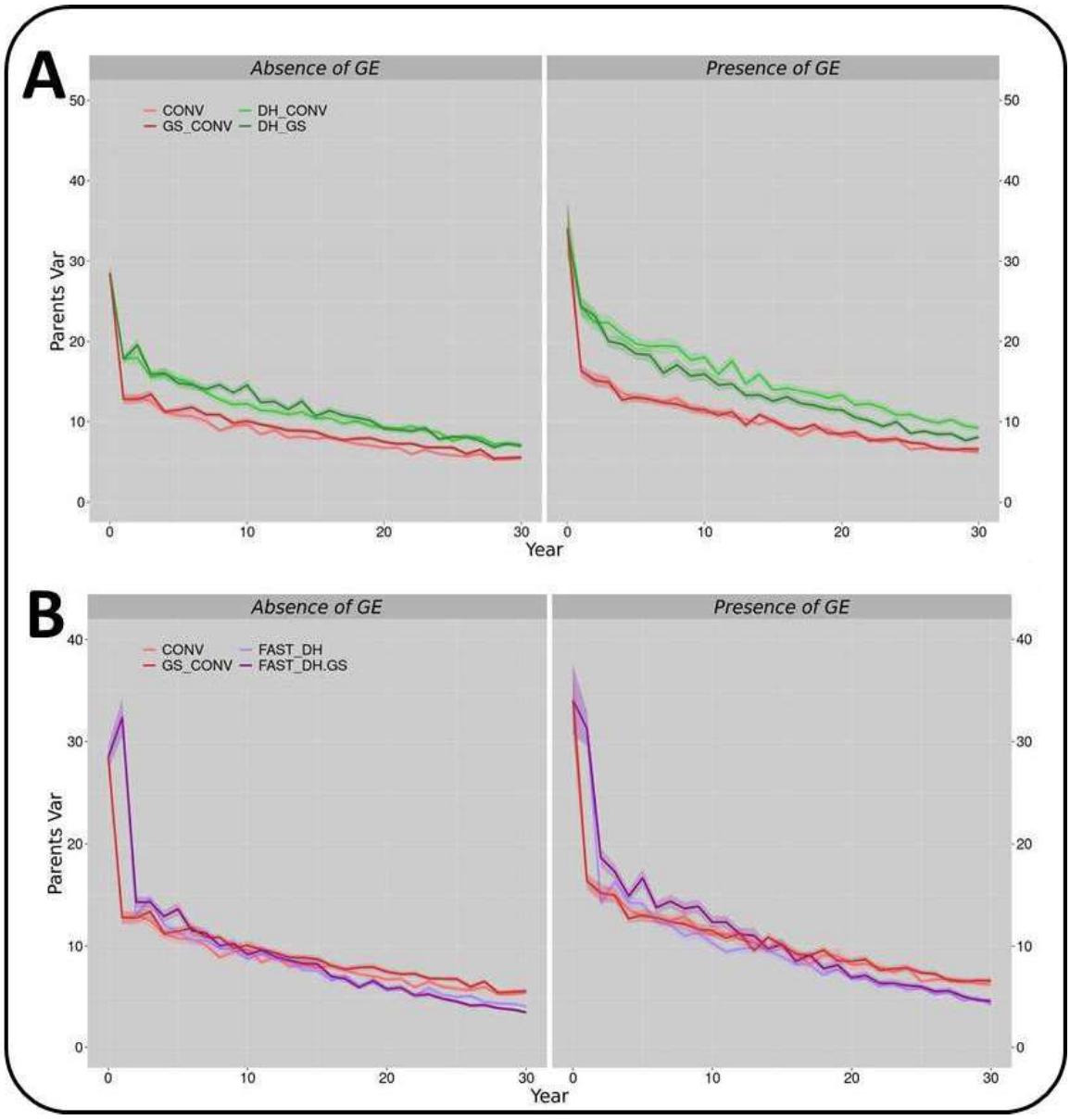

Figure S4.3 – Parental (crossing block) genetic variance over 30 years of breeding program, over presence and absence of genotype-by-interaction effect (GE). The conventional (CONV) and conventional using genomic selection (GS) (GS_CONV) strategies were compared to the double haploids (DH) conventional and GS (DH_CONV and DH_GS, respectively) (A) and with the DH adopting only two testcross' stages (FAST_DH) and applying genomic (FAST_DH.GS) (B).

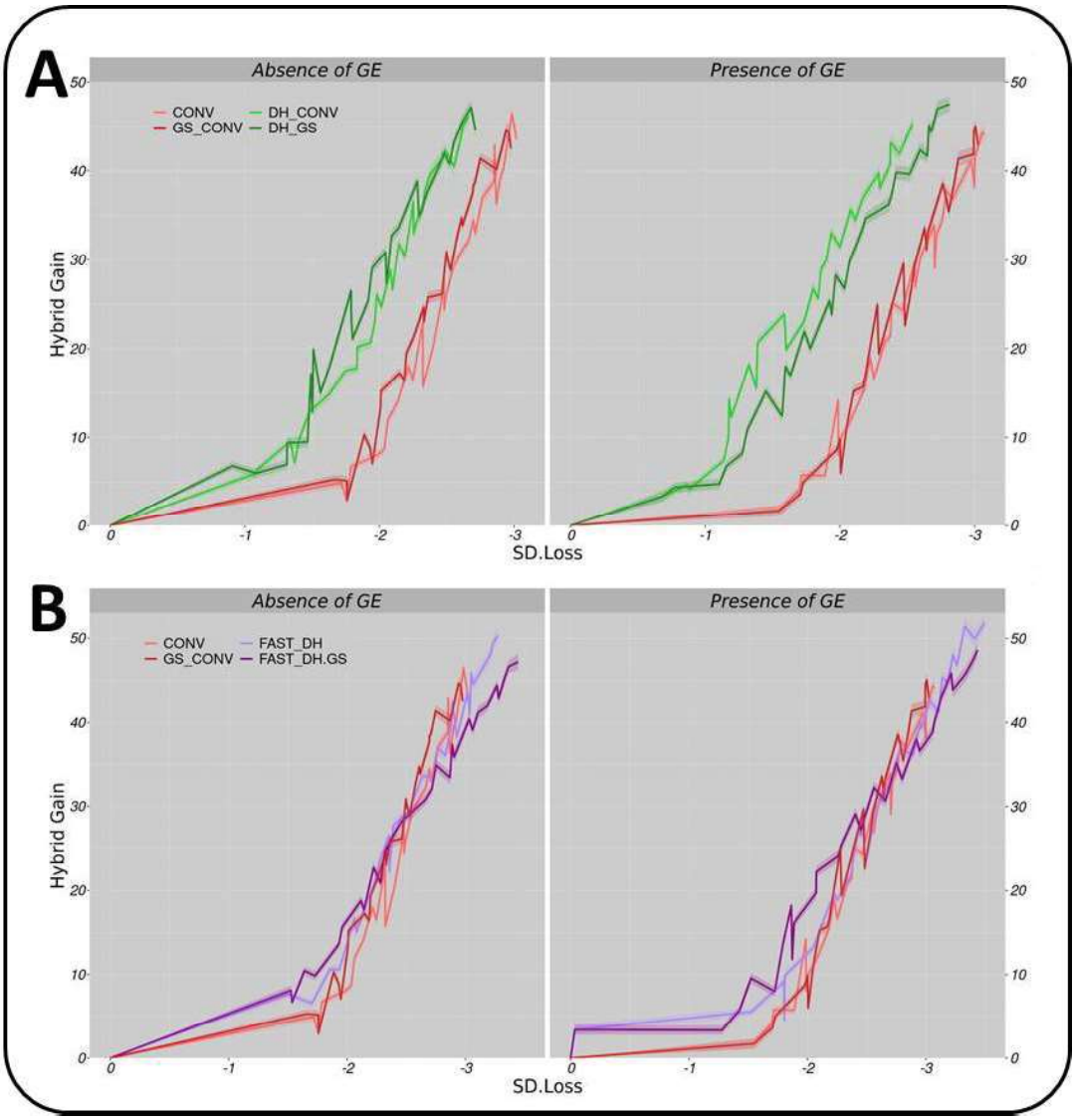

Figure S.5 – Hybrid genetic gain by the loss of genetic variance of the crossing block, under presence and absence of genotype-by-interaction effect (GE). The conventional (CONV) and conventional using genomic selection (GS) (GS_CONV) strategies were compared to the double haploids (DH) conventional and GS (DH_CONV and DH_GS, respectively) (A) and with the DH adopting only two testcross' stages (FAST_DH) and applying genomic (FAST_DH.GS) (B).

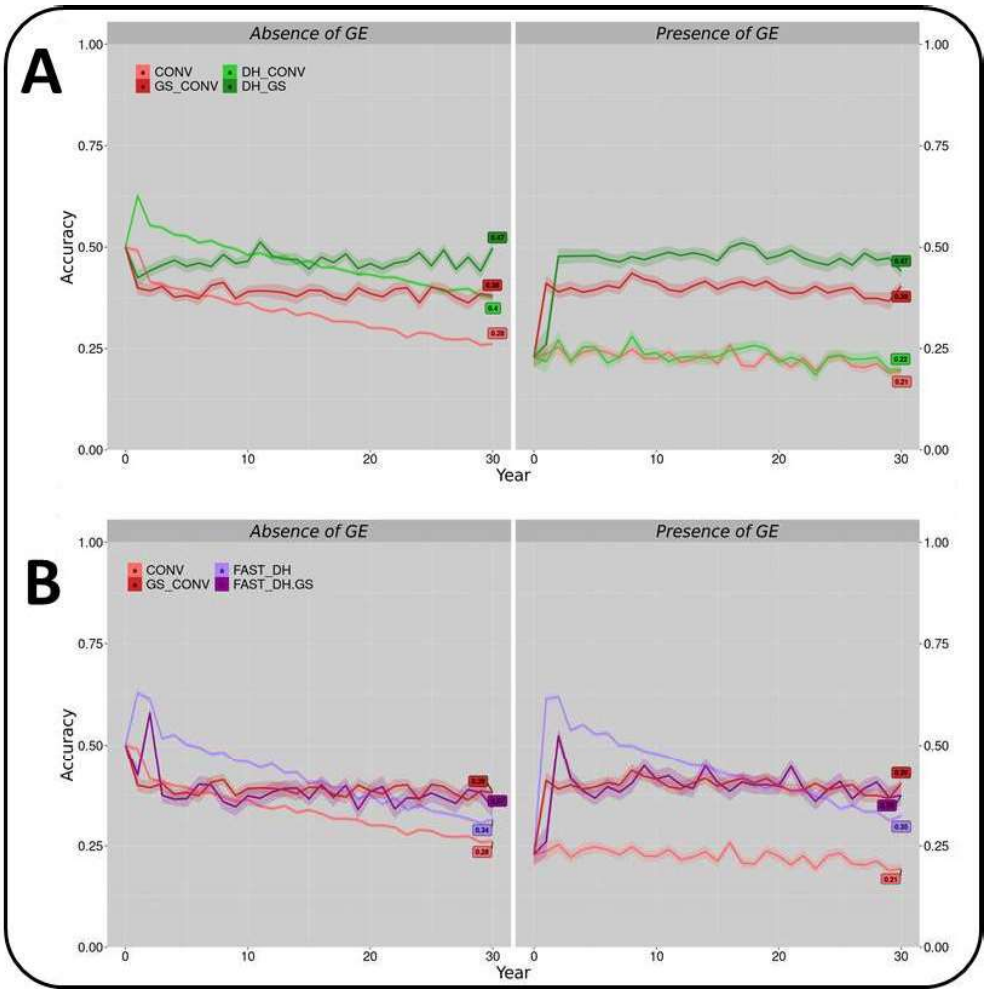

Figure S6.1 – Accuracy of genomic selection model, over presence and absence of genotype-by-interaction effect (GE). The conventional strategies which do not used the genomic selection model brings the correlation between phenotypic values and true genetic values *(TBV),* while the GS strategies, bring the correlation of estimated breeding value (*EBV*) and *TBV*. The conventional (CONV) and conventional using genomic selection (GS) (GS_CONV) strategies were compared to the double haploids (DH) conventional and GS (DH_CONV and DH_GS, respectively) (A) and with the DH adopting only two testcross' stages (FAST_DH) and applying genomic (FAST_DH.GS) (B). The *CONV* and *GS_CONV* brings the correlation of F3 stage. Labels denote the average accuracy of the last 10 years of the breeding program.

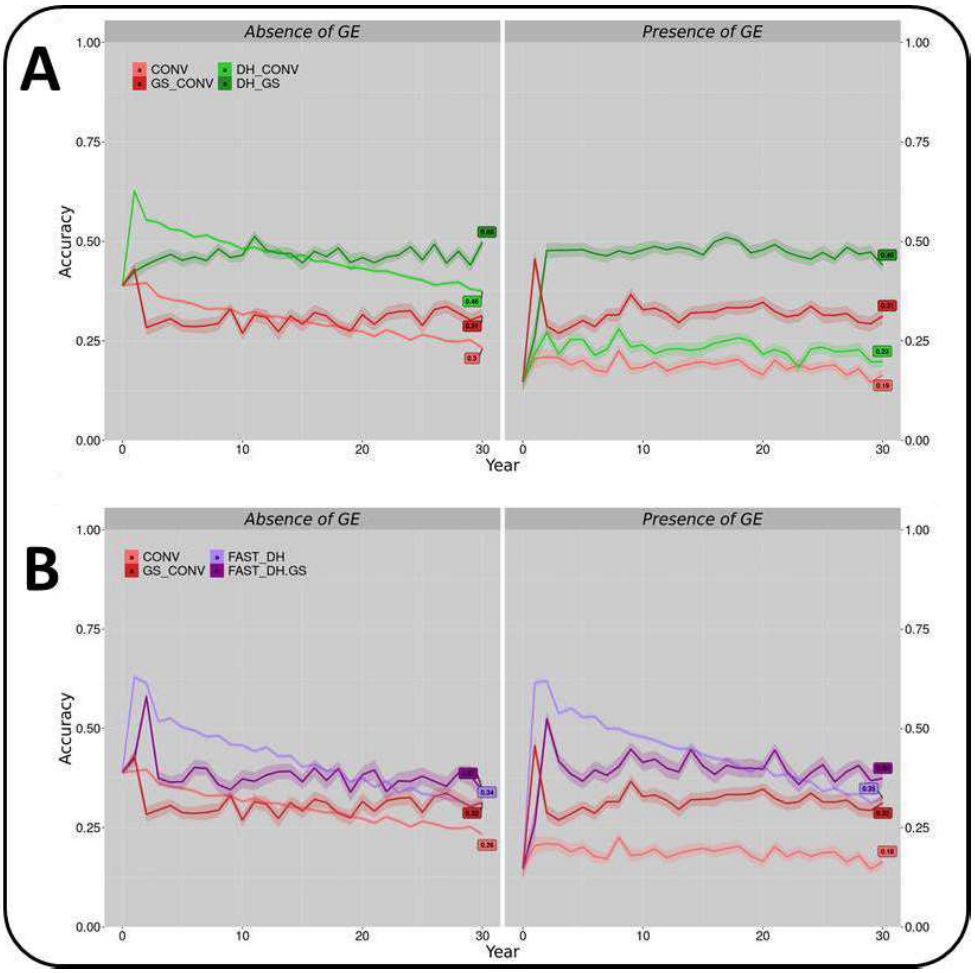

Figure S6.2 – Accuracy of genomic selection model, over presence and absence of genotype-by-interaction effect (GE). The conventional strategies which do not used the genomic selection model brings the correlation between phenotypic values and true genetic values *(TBV),* while the GS strategies, bring the correlation of estimated breeding value (*EBV*) and *TBV*. The conventional (CONV) and conventional using genomic selection (GS) (GS_CONV) strategies were compared to the double haploids (DH) conventional and GS (DH_CONV and DH_GS, respectively) (A) and with the DH adopting only two testcross' stages (FAST_DH) and applying genomic (FAST_DH.GS) (B). The *CONV* and *GS_CONV* brings the correlation of F5 stage. Labels denote the average accuracy of the last 10 years of the breeding program.

Table S2. Breeding strategies size, time, and performance information. The scenarios (Scen) were: conventional (CONV) and conventional using genomic selection (GS) (GS_CONV) strategies were compared to the double haploids (DH) conventional and GS (DH_CONV and DH_GS, respectively), and with the DH adopting only two testcross' stages (FAST_DH) and applying genomic (FAST_DH.GS). The years of the breeding cycle (Cycle) planted plots per year (Plots), annual costs, in dollars, relative gains related to the CONV strategy, under presence and absence of genotype-by-environment interaction (G×E), and the Cost efficiency, considering relative gain, relative costs, and relative cycles. The DH adopted the using of spontaneous haploid genome doubling, being 40% cheaper than the current cost.

Scen	Cycle	Plots	Cost	Relative Gain		Cost Efficiency	
	Years	Total	Annual ($)	Absence of GE	Presence of GE	Absence of GE	Presence of GE
CONV	5	3970	$49,640	1.00	1.00	1.00	1.00
GS_CONV	5	4134	$87,124	0.93	1.01	0.53	0.57
DH_CONV	4	5649	$95,000	1.00	1.05	0.65	0.69
DH_GS	4	6449	$100,852	0.99	1.04	0.61	0.64
DH_FAST	3	4309	$74,628	1.11	1.21	1.23	1.35
DH_FAST.GS	3	5109	$87,428	1.10	1.17	1.04	1.11

4.8.2 Scripts

```
#########################################################
######### AlphaSimR code for sweet corn breeding ###########
#########################################################
# This is the first script of the simulation
rm(list = ls())
require(AlphaSimR)
options(echo=TRUE)
args = commandArgs(trailingOnly=TRUE)
rep <- as.numeric(args[1])
VarGE <- as.numeric(args[2])
cat(rep)
cat(VarGE)

##>>>---------- setting the scenarios
source("Sc.0.0.GlobalParameters.R")

# Initialize variables for results
inbredMean_private = inbredVar_private =
  popMean_private = popVar_private =
  hybridMean = hybridVar =
  parentsMean = parentsVar =
  acc1 = acc2 =
  geneticGainHyb = geneticGainPar =
```

```
  GainHyb_perCycle = GainPar_perCycle =
  Efficiency =
  rep(NA_real_,burninYears+futureYears)

output0 = list(inbredMean_private=NULL,
               inbredVar_private=NULL,
               popMean_private=NULL,
               popVar_private=NULL)

output = list(hybridMean=NULL,
              hybridVar=NULL,
              parentsMean=NULL,
              parentsVar=NULL,
              acc1=NULL,
              acc2=NULL,
              geneticGainHyb=NULL,
              geneticGainPar=NULL,
              GainHyb_perCycle=NULL,
              GainPar_perCycle=NULL,
              Efficiency=NULL)

# Save results
saveRDS(output0,sprintf("Scenario_0_r%s_h2%s_GE%s.rds",rep,H2,VarGE))
saveRDS(output,sprintf("Scenario_BS_r%s_h2%s_GE%s.rds",rep,H2,VarGE))
saveRDS(output,sprintf("Scenario_GS_r%s_h2%s_GE%s.rds",rep,H2,VarGE))
```

```
saveRDS(output,sprintf("Scenario_DH_r%s_h2%s_GE%s.rds",rep,H2,VarGE))
saveRDS(output,sprintf("Scenario_DHGS_r%s_h2%s_GE%s.rds",rep,H2,VarGE))
saveRDS(output,sprintf("Scenario_DH1_r%s_h2%s_GE%s.rds",rep,H2,VarGE))
saveRDS(output,sprintf("Scenario_DHGS1_r%s_h2%s_GE%s.rds",rep,H2,VarGE))

### Loop through replications
# Create initial parents and set testers and hybrid parents
source("Sc.0.0.CreatParents.R")
# Fill breeding pipeline with unique individuals from initial parents
source("Sc.0.0.FillPipeline.R")
# p-values for GxY effects
Pgy = 0.5 + rnorm(burninYears+futureYears,0,0.03)
Pgye1 = 0.1 + rnorm(burninYears+futureYears,0,0.03)
Pgye2 = 0.9 + rnorm(burninYears+futureYears,0,0.03)
Pgye.all = c(Pgye1,Pgye2)
Pgye = sample(Pgye.all,burninYears+futureYears,replace=F)

##########################################################
############# Scenario 0 - Private breeding program ###########
##########################################################
cat("Working on Scenario 0\n")
for(year in 1:(burninYears)){
  cat("Working on Year:",year,"\n")
  pgy = Pgy[year]
  pgye = Pgye[year]
```

```
  source("Sc.0.1_UpdateParents_private.R") #Pick new parents and testers
  source("Sc.0.1_Advanced_cycle_private.R") #Advances private yield trials by a year
  source("Sc.0.1_UpdateResults_private.R") #Track summary data
  source("Sc.0.1_UpdateTesters.R") #Pick new testers
  source("Sc.0.2_UpdateParents_public.R") #Pick new parents
  source("Sc.0.2_Advanced_cycle_public.R") #Advances public yield trials by a year
  source("Sc.0.2_UpdateResults_public.R") #Track summary data
}

#Save burn-in to load later use
save.image(sprintf("tmp_r%s_h2%s_GE%s.rda",rep,H2,VarGE))

############################################################
############# Scenario 1 - BenchMark Program - Discrete
############################################################
load(sprintf("tmp_r%s_h2%s_GE%s.rda",rep,H2,VarGE))
cat("Working on Scenario 1\n")
for(year in (burninYears+1):(burninYears+futureYears)){
  cat("Working on Year:",year,"\n")
  pgy = Pgy[year]
  pgye = Pgye[year]
  source("Sc.0.1_UpdateParents_private.R") #Pick new parents and testers
  source("Sc.0.1_Advanced_cycle_private.R") #Advances private yield trials by a year
  source("Sc.0.1_UpdateResults_private.R") #Track summary data
  source("Sc.0.1_UpdateTesters.R") #Pick new testers
```

```
  source("Sc.1_UpdateParents.BS.R") #Pick new parents
  source("Sc.1_Advanced_cycle.BS.R") #Advances yield trials by a year
  source("Sc.1_UpdateResults.BS.R") #Track summary data
}

# Report results for scenario 1
output = readRDS(sprintf("Scenario_BS_r%s_h2%s_GE%s.rds",rep,H2,VarGE))
output = list(hybridMean=rbind(output$hybridMean,hybridMean),
              hybridVar=rbind(output$hybridVar,hybridVar),
              parentsMean=rbind(output$parentsMean,parentsMean),
              parentsVar=rbind(output$parentsVar,parentsVar),
              acc1=rbind(output$acc1,acc1),
              acc2=rbind(output$acc2,acc2),
              geneticGainHyb=rbind(output$geneticGainHyb,geneticGainHyb),
              geneticGainPar=rbind(output$geneticGainPar,geneticGainPar),
              GainHyb_perCycle=rbind(output$GainHyb_perCycle,GainHyb_perCycle),
              GainPar_perCycle=rbind(output$GainPar_perCycle,GainPar_perCycle),
              Efficiency=rbind(output$Efficiency,Efficiency))
saveRDS(output, sprintf("Scenario_BS_r%s_h2%s_GE%s.rds",rep,H2,VarGE))

#############################################################
############# Scenario 2 - Genomic selection breeding program - Discrete
#############################################################
load(sprintf("tmp_r%s_h2%s_GE%s.rda",rep,H2,VarGE))
cat("Working on Scenario 2\n")
```

```
for(year in (burninYears+1):(burninYears+futureYears)){
  cat("Working on Year:",year,"\n")
  pgy = Pgy[year]
  pgye = Pgye[year]
  source("Sc.0.1_UpdateParents_private.R") #Pick new parents and testers
  source("Sc.0.1_Advanced_cycle_private.R") #Advances private yield trials by a year
  source("Sc.0.1_UpdateResults_private.R") #Track summary data
  source("Sc.0.1_UpdateTesters.R") #Pick new testers
   if(year==(burninYears+1)){
    gsModel = RRBLUP(c(F1_public,F2_public), use = "gv", snpChip = 1)
  }
  source("Sc.2_UpdateParents.GS.R") #Pick new parents
  source("Sc.2_Advanced_cycle.GS.R") #Advances yield trials by a year
  source("Sc.2_UpdateResults.GS.R") #Track summary data
}

# Report results for scenario 2
output = readRDS(sprintf("Scenario_GS_r%s_h2%s_GE%s.rds",rep,H2,VarGE))
output = list(hybridMean=rbind(output$hybridMean,hybridMean),
              hybridVar=rbind(output$hybridVar,hybridVar),
              parentsMean=rbind(output$parentsMean,parentsMean),
              parentsVar=rbind(output$parentsVar,parentsVar),
              acc1=rbind(output$acc1,acc1),
              acc2=rbind(output$acc2,acc2),
              geneticGainHyb=rbind(output$geneticGainHyb,geneticGainHyb),
```

```
        geneticGainPar=rbind(output$geneticGainPar,geneticGainPar),
        GainHyb_perCycle=rbind(output$GainHyb_perCycle,GainHyb_perCycle),
        GainPar_perCycle=rbind(output$GainPar_perCycle,GainPar_perCycle),
        Efficiency=rbind(output$Efficiency,Efficiency))
saveRDS(output, sprintf("Scenario_GS_r%s_h2%s_GE%s.rds",rep,H2,VarGE))

#########################################################
############# Scenario 3 - Conventional DH breeding program
#########################################################
load(sprintf("tmp_r%s_h2%s_GE%s.rda",rep,H2,VarGE))
cat("Working on Scenario 3\n")
for(year in (burninYears+1):(burninYears+futureYears)){
  pgy = Pgy[year]
  pgye = Pgye[year]
  source("Sc.0.1_UpdateParents_private.R") #Pick new parents and testers
  source("Sc.0.1_Advanced_cycle_private.R") #Advances private yield trials by a year
  source("Sc.0.1_UpdateResults_private.R") #Track summary data
  source("Sc.0.1_UpdateTesters.R") #Pick new testers
  if(year==(burninYears+1)){
    DH_Pub1=makeDH(F1_public, nDH = pub_DH, keepParents = FALSE)
    TC3_DH=TC2_DH=TC1_DH=DH_Pub3=DH_Pub2=DH_Pub1
  }
  cat("Working on Cycle-futureYears:",year,"\n")
  source("Sc.3_UpdateParents.DH.R") #Pick new parents
  source("Sc.3_Advanced_cycle.DH.R") #Advances yield trials by a year
```

```
  source("Sc.3_UpdateResults.DH.R") #Track summary data
}

# Report results for scenario 3
output = readRDS(sprintf("Scenario_DH_r%s_h2%s_GE%s.rds",rep,H2,VarGE))
output = list(hybridMean=rbind(output$hybridMean,hybridMean),
              hybridVar=rbind(output$hybridVar,hybridVar),
              parentsMean=rbind(output$parentsMean,parentsMean),
              parentsVar=rbind(output$parentsVar,parentsVar),
              acc1=rbind(output$acc1,acc1),
              acc2=rbind(output$acc2,acc2),
              geneticGainHyb=rbind(output$geneticGainHyb,geneticGainHyb),
              geneticGainPar=rbind(output$geneticGainPar,geneticGainPar),
              GainHyb_perCycle=rbind(output$GainHyb_perCycle,GainHyb_perCycle),
              GainPar_perCycle=rbind(output$GainPar_perCycle,GainPar_perCycle),
              Efficiency=rbind(output$Efficiency,Efficiency))
saveRDS(output,sprintf("Scenario_DH_r%s_h2%s_GE%s.rds",rep,H2,VarGE))

##########################################################
############# Scenario 4 - Genomic Selection on DH breeding program
##########################################################
load(sprintf("tmp_r%s_h2%s_GE%s.rda",rep,H2,VarGE))
cat("Working on Scenario 4\n")
for(year in (burninYears+1):(burninYears+futureYears)){
  pgy = Pgy[year]
```

```
  pgye = Pgye[year]
  source("Sc.0.1_UpdateParents_private.R") #Pick new parents and testers
  source("Sc.0.1_Advanced_cycle_private.R") #Advances private yield trials by a year
  source("Sc.0.1_UpdateResults_private.R") #Track summary data
  source("Sc.0.1_UpdateTesters.R") #Pick new testers
  if(year==(burninYears+1)){
    DH_Pub1=makeDH(F1_public, nDH = pub_DH, keepParents = FALSE)
    TC3_DH=TC2_DH=TC1_DH=DH_Pub3=DH_Pub2=DH_Pub1
    gsModel = RRBLUP(F1_public, use = "gv", snpChip = 1)
    }
  cat("Working on Cycle-futureYears:",year,"\n")
  source("Sc.4_UpdateParents.DHGS.R") #Pick new parents
  source("Sc.4_Advanced_cycle.DHGS.R") #Advances yield trials by a year
  source("Sc.4_UpdateResults.DHGS.R") #Track summary data
}

# Report results for scenario 4
output = readRDS(sprintf("Scenario_DHGS_r%s_h2%s_GE%s.rds",rep,H2,VarGE))
output = list(hybridMean=rbind(output$hybridMean,hybridMean),
              hybridVar=rbind(output$hybridVar,hybridVar),
              parentsMean=rbind(output$parentsMean,parentsMean),
              parentsVar=rbind(output$parentsVar,parentsVar),
              acc1=rbind(output$acc1,acc1),
              acc2=rbind(output$acc2,acc2),
              geneticGainHyb=rbind(output$geneticGainHyb,geneticGainHyb),
```

```
        geneticGainPar=rbind(output$geneticGainPar,geneticGainPar),
        GainHyb_perCycle=rbind(output$GainHyb_perCycle,GainHyb_perCycle),
        GainPar_perCycle=rbind(output$GainPar_perCycle,GainPar_perCycle),
        Efficiency=rbind(output$Efficiency,Efficiency))
saveRDS(output,sprintf("Scenario_DHGS_r%s_h2%s_GE%s.rds",rep,H2,VarGE))

#########################################################
############# Scenario 5 - Proposed DH breeding program
#########################################################
load(sprintf("tmp_r%s_h2%s_GE%s.rda",rep,H2,VarGE))
cat("Working on Scenario 5\n")
for(year in (burninYears+1):(burninYears+futureYears)){
  pgy = Pgy[year]
  pgye = Pgye[year]
  source("Sc.0.1_UpdateParents_private.R") #Pick new parents and testers
  source("Sc.0.1_Advanced_cycle_private.R") #Advances private yield trials by a year
  source("Sc.0.1_UpdateResults_private.R") #Track summary data
  source("Sc.0.1_UpdateTesters.R") #Pick new testers
  if(year==(burninYears+1)){
    DH_Pub1=makeDH(F1_public, nDH = pub_DH, keepParents = FALSE)
    TC3_DH=TC2_DH=TC1_DH=DH_Pub3=DH_Pub2=DH_Pub1
  }
  cat("Working on Cycle-futureYears:",year,"\n")
  source("Sc.5_UpdateParents.DH.R") #Pick new parents
  source("Sc.5_Advanced_cycle.DH.R") #Advances yield trials by a year
```

```
  source("Sc.5_UpdateResults.DH.R") #Track summary data
 }

# Report results for scenario 5
output = readRDS(sprintf("Scenario_DH1_r%s_h2%s_GE%s.rds",rep,H2,VarGE))
output = list(hybridMean=rbind(output$hybridMean,hybridMean),
              hybridVar=rbind(output$hybridVar,hybridVar),
              parentsMean=rbind(output$parentsMean,parentsMean),
              parentsVar=rbind(output$parentsVar,parentsVar),
              acc1=rbind(output$acc1,acc1),
              acc2=rbind(output$acc2,acc2),
              geneticGainHyb=rbind(output$geneticGainHyb,geneticGainHyb),
              geneticGainPar=rbind(output$geneticGainPar,geneticGainPar),
              GainHyb_perCycle=rbind(output$GainHyb_perCycle,GainHyb_perCycle),
              GainPar_perCycle=rbind(output$GainPar_perCycle,GainPar_perCycle),
              Efficiency=rbind(output$Efficiency,Efficiency))
saveRDS(output,sprintf("Scenario_DH1_r%s_h2%s_GE%s.rds",rep,H2,VarGE))

##########################################################
############# Scenario 6 - Genomic Selection on Proposed DH breeding program
##########################################################
load(sprintf("tmp_r%s_h2%s_GE%s.rda",rep,H2,VarGE))
cat("Working on Scenario 6\n")
for(year in (burninYears+1):(burninYears+futureYears)){
  pgy = Pgy[year]
```

```
  pgye = Pgye[year]
  source("Sc.0.1_UpdateParents_private.R") #Pick new parents and testers
  source("Sc.0.1_Advanced_cycle_private.R") #Advances private yield trials by a year
  source("Sc.0.1_UpdateResults_private.R") #Track summary data
  source("Sc.0.1_UpdateTesters.R") #Pick new testers
  if(year==(burninYears+1)){
    DH_Pub1=makeDH(F1_public, nDH = pub_DH, keepParents = FALSE)
    TC3_DH=TC2_DH=TC1_DH=DH_Pub3=DH_Pub2=DH_Pub1
    gsModel = RRBLUP(F1_public, use = "gv", snpChip = 1)
  }
  cat("Working on Cycle-futureYears:",year,"\n")
  source("Sc.6_UpdateParents.DHGS.R") #Pick new parents
  source("Sc.6_Advanced_cycle.DHGS.R") #Advances yield trials by a year
  source("Sc.6_UpdateResults.DHGS.R") #Track summary data
}

# Report results for scenario 6
output = readRDS(sprintf("Scenario_DHGS1_r%s_h2%s_GE%s.rds",rep,H2,VarGE))
output = list(hybridMean=rbind(output$hybridMean,hybridMean),
              hybridVar=rbind(output$hybridVar,hybridVar),
              parentsMean=rbind(output$parentsMean,parentsMean),
              parentsVar=rbind(output$parentsVar,parentsVar),
              acc1=rbind(output$acc1,acc1),
              acc2=rbind(output$acc2,acc2),
              geneticGainHyb=rbind(output$geneticGainHyb,geneticGainHyb),
```

```
        geneticGainPar=rbind(output$geneticGainPar,geneticGainPar),
        GainHyb_perCycle=rbind(output$GainHyb_perCycle,GainHyb_perCycle),
        GainPar_perCycle=rbind(output$GainPar_perCycle,GainPar_perCycle),
        Efficiency=rbind(output$Efficiency,Efficiency))
saveRDS(output,sprintf("Scenario_DHGS1_r%s_h2%s_GE%s.rds",rep,H2,VarGE))

file.remove(sprintf("tmp_r%s_h2%s_GE%s.rda",rep,H2,VarGE))
```

- **Sc.0.0_CreatParents**

```
library(AlphaSimR)
##############################################
########### Initializing the Program : Parameters #####
##############################################
#Set.seed for each replication:
set.seed (011122)
n <- sample(0:100000,50,replace=F)
set.seed(n[rep])
#Creating a founder population:
founderPop <- runMacs(nInd = nParents, nChr = 10, segSites = nQtl+nSnp, inbred =
FALSE,
                      species = "MAIZE", split = 30, ploidy = 2L,
                      manualCommand = NULL, manualGenLen = NULL)
#Set the simulation parameters:
SP <- SimParam$new(founderPop)
SP$restrSegSites(nQtl,nSnp)
if(nSnp>0){
```

```
  SP$addSnpChip(nSnp)
}
SP$setSexes("no")
SP$addTraitADG(nQtl,mean=MeanG,var=VarG,
            varGxE=VarGE,meanDD=ddMean,varDD=ddVar)
SP$setVarE(H2=H2)

##############################################
###############  Breeding Cycles ###############
##############################################
Parents_public = newPop(founderPop[1:50])
Parents_public_update = Parents_public
Parents_private = newPop(founderPop[101:200])
Parents_private_update = Parents_private
#Set hybrid parents for later yield trials
nElite = 5
nTester1 = 1
nTester2 = 3
nTester3 = 5
Elite_private = selectInd(Parents_private,nElite,use="gv")
Elite_private = Elite_private[nElite:1]
Tester1_private = Elite_private[1:nTester1]
Tester2_private = Elite_private[1:nTester2]
Tester3_private = Elite_private[1:nTester3]
repTC1 = 2
```

```
repTC2 = 5

repTC3 = 20

nCrosses_TS= 800

nCrosses_public = 50

nCrosses_private = 200

pub_DH = 25
```

- **Sc.0.0.FillPipeline**

```
library(AlphaSimR)

# p-values for GxY effects

Pgy = 0.5 + rnorm(burninYears+futureYears,0,0.03)

Pgye = 0.1 + rnorm(burninYears+futureYears,0,0.03)

P = runif(7)

#p-values for GxY effect

for(year in 1:7){

  cat("FillPipeline year:",year,"of 7\n")

  pgy = Pgy[year]

  pgye = Pgye[year]

  p=P[year]


  #Year 1

  ####>>>>----Public program

  F1_public = randCross(Parents_public_update, nCrosses_public, ignoreSexes =
TRUE) # Spring

  F2_public = self(F1_public, nProgeny = 10, parents = NULL, keepParents = FALSE) #
Fall
```

```
  F3_public = self(F2_public, nProgeny = 6, parents = NULL, keepParents = FALSE) #
Winter

  F2_public = setPheno(F2_public,reps=1) # Winter

  F2_public.sel = selectWithinFam(F2_public, nInd = 4, selectTop = TRUE) # Winter

  F3_public.s0 = F3_public[F3_public@mother%in%F2_public.sel@id] # Winter

  ####>>>>----Private program

  F1_private = randCross(Parents_private, nCrosses_private, ignoreSexes = TRUE)

  F2_private = self(F1_private, nProgeny = 10, parents = NULL, keepParents = FALSE)

  F2_private = setPheno(F2_private,reps=1)

  F2_private.sel = selectWithinFam(F2_private, nInd = 6, selectTop = TRUE)


  #Year 2

  if(year<7){

    pgy = Pgy[year]

    pgye = Pgye[year]

    p=P[year]

    ####>>>>----Public program

    F4_public = self(F3_public.s0, nProgeny = 4, parents = NULL, keepParents = FALSE)
# Fall

    F3_public.s1 = setPheno(F3_public.s0,reps=1, p=pgye) # Fall

    F3_public.sel = selectWithinFam(F3_public.s1, nInd = 2, selectTop = TRUE) # Fall

    F4_public.s0 = F4_public[F4_public@mother%in%F3_public.sel@id] # Fall

    F5_public = self(F4_public.s0, nProgeny = 4, parents = NULL, keepParents = FALSE)
# Winter

    F4_public.s0 = setPheno(F4_public.s0,reps=2, p=pgy) # Winter

    F4_public.sel = selectWithinFam(F4_public.s0, nInd = 2, selectTop = TRUE) # Winter

    F4_public.sel = selectInd(F4_public.sel, nInd = 300, selectTop = TRUE) # Winter
```

```
  F5_public.s0 = F5_public[F5_public@mother%in%F4_public.sel@id]

  ####>>>>----Private program

  F3_private = self(F2_private.sel, nProgeny = 8, parents = NULL, keepParents =
FALSE)

  F3_private = setPheno(F3_private,reps=1, p=p)

  F3_private.sel = selectWithinFam(F3_private, nInd = 2, selectTop = TRUE)

 }


 #Year 3

 if(year<6){

  pgy = Pgy[year]

  pgye = Pgye[year]

  p=P[year]

  ####>>>>----Public program

  F6_public = self(F5_public.s0, nProgeny = 3, parents = NULL, keepParents = FALSE)
# Fall

  F5_public.s1 = setPheno(F5_public.s0,reps=1, p=pgye) # Fall

  F5_public.sel = selectWithinFam(F5_public.s1, nInd = 2, selectTop = TRUE) # Fall

  F5_public.sel = selectInd(F5_public.sel, nInd = 150, selectTop = TRUE) # Winter

  F6_public.s0 = F6_public[F6_public@mother%in%F5_public.sel@id] # Fall

  TC1 = setPhenoGCA(F5_public.sel,Tester1_private,reps=repTC1,p=pgy) #
Fall/Winter

  TC1 = selectInd(TC1,nInd=100, selectTop = TRUE) # Winter

  F7_public = self(F6_public.s0, nProgeny = 1, parents = NULL, keepParents = FALSE)
# Winter

  F6_public.sel = F6_public.s0[F6_public.s0@mother%in%TC1@id]

  F6_public.sel = selectWithinFam(F6_public.sel, nInd = 1, selectTop = TRUE) # Fall
```

```
    F7_public.sel = F7_public[F7_public@mother%in%F6_public.sel@id] # Fall

    ####>>>>----Private program

    F4_private = self(F3_private.sel, nProgeny = 6, parents = NULL, keepParents =
FALSE)

    F4_private = setPheno(F4_private,reps=2, p=p)

    F4_private.sel = selectWithinFam(F4_private, nInd = 2, selectTop = TRUE)

    F4_private.sel = selectInd(F4_private.sel, nInd = length(F4_private.sel@id) * 0.2,
selectTop = TRUE)

  }


  #Year 4

  if(year<5){

    pgy = Pgy[year]

    pgye = Pgye[year]

    p=P[year]

    ####>>>>----Public program

    F8_public = self(F7_public.sel, nProgeny = 1, parents = NULL, keepParents =
FALSE) # Fall

    TC2 = setPhenoGCA(F7_public.sel, Tester2_private,reps=repTC2,p=pgy) #
Fall/Winter

    F9_public = self(F8_public, nProgeny = 1, parents = NULL, keepParents = FALSE) #
Winter

    TC2 = selectInd(TC2,nInd=20, selectTop = TRUE) # Winter

    F8_public.sel = F8_public[F8_public@mother%in%TC2@id]

    F9_public.sel = F9_public[F9_public@mother%in%F8_public.sel@id] # Winter

    ####>>>>----Private program

    F5_private = self(F4_private.sel, nProgeny = 6, parents = NULL, keepParents =
FALSE)
```

```
    F5_private = setPheno(F5_private,reps=2, p=p)

    F5_private.sel = selectWithinFam(F5_private, nInd = 1, selectTop = TRUE)

    F5_private.sel = selectInd(F5_private.sel, nInd = length(F5_private.sel@id) * 0.15,
selectTop = TRUE)

  }


  #Year 5

  if(year<4){

    pgy = Pgy[year]

    pgye = Pgye[year]

    p=P[year]

    ####>>>>----Public program

    F10_public = self(F9_public.sel, nProgeny = 1, parents = NULL, keepParents =
FALSE) # Fall

    TC3 = setPhenoGCA(F9_public.sel, Tester3_private,reps=repTC3,p=pgy) # Winter

    F11_public = self(F10_public, nProgeny = 1, parents = NULL, keepParents = FALSE)
# Winter

    TC3 = selectInd(TC3,nInd=2, selectTop = TRUE) # Winter

    F10_public.sel = F10_public[F10_public@mother%in%TC3@id]

    F11_public.sel = F11_public[F11_public@mother%in%F10_public.sel@id] # Winter

    Hybrid = hybridCross(F9_public.sel, Tester3_private,crossPlan = "testcross") # Fall

    Hybrid = setPheno(Hybrid,reps=repTC3,p=pgy) # Winter

    Hybrid = selectInd(Hybrid,nInd=2, selectTop = TRUE) # Winter

   ####>>>>----Private program

    F6_private = self(F5_private.sel, nProgeny = 6, parents = NULL, keepParents =
FALSE)

    F6_private = setPheno(F6_private,reps=2, p=p)
```

```
    F6_private.sel = selectWithinFam(F6_private, nInd = 1, selectTop = TRUE)

    F6_private.sel = selectInd(F6_private.sel, nInd = length(F6_private.sel@id) * 0.5,
selectTop = TRUE)

  }


  #Year 6

  if(year<3){

    pgy = Pgy[year]

    pgye = Pgye[year]

    p=P[year]

    ####>>>>----Private program

    F7_private = self(F6_private.sel, nProgeny = 6, parents = NULL, keepParents =
FALSE)

    F7_private = setPheno(F7_private,reps=2, p=p)

    F7_private.sel = selectWithinFam(F7_private, nInd = 1, selectTop = TRUE)

    F7_private.sel = selectInd(F7_private.sel, nInd = 15, selectTop = TRUE)

  }


  #Year 7

  if(year<2){

    pgy = Pgy[year]

    pgye = Pgye[year]

    p=P[year]

    ####>>>>----Private program

    F8_private = self(F7_private.sel, nProgeny = 6, parents = NULL, keepParents =
FALSE)

    F8_private = setPheno(F8_private,reps=2, p=p)
```

```
    F8_private.sel = selectWithinFam(F8_private, nInd = 1, selectTop = TRUE)
    F8_private.sel = selectInd(F8_private.sel, nInd = 5, selectTop = TRUE)
  }
}
```

- **Sc.0.0.GlobalParameters**

```
############################################################
#### Parameters of the base population
############################################################
burninYears = 20
futureYears = 30
nParents=200
nParents_public = 50
nParents_private = 100
nQtl = 300
nSnp = 3000
MeanG = 150
VarG =  30
ddVar = 0.3
ddMean = 0.93
H2 = 0.3
```

- **Sc.0.0_Parameters_loop**

```
#Testing Parameters
rep <- rep(1:50,each=4)
```

```
VarGE <- rep(c(0,30,60,120),each=1,times=50)

INPUT.FILE = data.frame(rep,VarGE)

write.table(INPUT.FILE,"3.Scenarios.txt",sep="\t",col.names=FALSE,row.names=FALSE
)
```

- **Sc.0.1_Advanced_cycle_private**

```
######  Year 7  #######

F8_private = self(F7_private.sel, nProgeny = 6, parents = NULL, keepParents = FALSE)
# Fall

F8_private = setPheno(F8_private,reps=2, p=pgy) # Winter

F8_private.par = selectWithinFam(F8_private, nInd = 1, selectTop = TRUE) # Winter

F8_private.sel = selectInd(F8_private.par, nInd = 5, selectTop = TRUE) # Winter


######  Year 6  #######

F7_private = self(F6_private.sel, nProgeny = 6, parents = NULL, keepParents = FALSE)
# Fall

F7_private = setPheno(F7_private,reps=2, p=pgy) # Winter

F7_private.sel = selectWithinFam(F7_private, nInd = 1, selectTop = TRUE) # Winter

F7_private.sel = selectInd(F7_private.sel, nInd = 15, selectTop = TRUE) # Winter


######  Year 5  #######

F6_private = self(F5_private.sel, nProgeny = 6, parents = NULL, keepParents = FALSE)
# Fall

F6_private = setPheno(F6_private,reps=2, p=pgy) # Winter

F6_private.sel = selectWithinFam(F6_private, nInd = 1, selectTop = TRUE) # Winter

F6_private.sel = selectInd(F6_private.sel, nInd = 85, selectTop = TRUE) # Winter
```

```
######  Year 4  #######

F5_private = self(F4_private.sel, nProgeny = 6, parents = NULL, keepParents = FALSE)
# Fall

F5_private = setPheno(F5_private,reps=2, p=pgy) # Winter

F5_private.sel = selectWithinFam(F5_private, nInd = 1, selectTop = TRUE) # Winter

F5_private.sel = selectInd(F5_private.sel, nInd = length(F5_private.sel@id) * 0.15,
selectTop = TRUE) # Winter


######  Year 3  #######

F4_private = self(F3_private.sel, nProgeny = 6, parents = NULL, keepParents = FALSE)
# Fall

F4_private = setPheno(F4_private,reps=2, p=pgy) # Winter

F4_private.sel = selectWithinFam(F4_private, nInd = 2, selectTop = TRUE) # Winter

F4_private.sel = selectInd(F4_private.sel, nInd = length(F4_private.sel@id) * 0.2,
selectTop = TRUE) # Winter


######  Year 2  #######

F3_private = self(F2_private.sel, nProgeny = 6, parents = NULL, keepParents = FALSE)
# Fall

F3_private = setPheno(F3_private,reps=1, p=pgy) # Winter

F3_private.sel = selectWithinFam(F3_private, nInd = 2, selectTop = TRUE) # Winter


######  Year 1  #######

F1_private = randCross(Parents_private_update, nCrosses_private, ignoreSexes =
TRUE) # Spring

F2_private = self(F1_private, nProgeny = 10, parents = NULL, keepParents = FALSE) #
Fall

F2_private = setPheno(F2_private,reps=1, p=pgy) # Winter
```

F2_private.sel = selectWithinFam(F2_private, nInd = 5, selectTop = TRUE) # Winter

- **Sc.0.1_UpdateParents_private**

Parents_private_update = c(F7_private.sel,F6_private.sel)

- **Sc.0.1_UpdateResults_private**

popMean_private[year] = meanG(c(F8_private.sel, F7_private.sel, F6_private.sel, F5_private.sel, F4_private.sel, F3_private.sel, F2_private.sel))

popVar_private[year] = varG(c(F8_private.sel, F7_private.sel, F6_private.sel, F5_private.sel, F4_private.sel, F3_private.sel, F2_private.sel))

inbredMean_private[year] = meanG(c(F8_private.sel, F7_private.sel))

inbredVar_private[year] = varG(c(F8_private.sel, F7_private.sel))

- **Sc.0.1_UpdateTesters**

Elite_private = c(Elite_private,Parents_private_update)

Elite_private = selectInd(Elite_private, nInd = nElite, selectTop = TRUE,use="gv")

#Update testers

Tester1_private = Elite_private[1:nTester1]

Tester2_private = Elite_private[1:nTester2]

Tester3_private = Elite_private[1:nTester3]

- **Sc.0.2_Advanced_cycle_public**

Year 5

F10_public = self(F9_public.sel, nProgeny = 1, parents = NULL, keepParents = FALSE) # Fall

TC3 = setPhenoGCA(F9_public.sel, Tester3_private,reps=repTC3,p=pgy) # Winter

F11_public = self(F10_public, nProgeny = 1, parents = NULL, keepParents = FALSE) # Winter

```
TC3 = selectInd(TC3,nInd=2, selectTop = TRUE) # Winter

F10_public.sel = F10_public[F10_public@mother%in%TC3@id]

F11_public.sel = F11_public[F11_public@mother%in%F10_public.sel@id] # Winter

Hybrid = hybridCross(F9_public.sel, Tester3_private,crossPlan = "testcross") # Fall

Hybrid = setPheno(Hybrid,reps=repTC3,p=pgy) # Winter

Hybrid = selectInd(Hybrid,nInd=2, selectTop = TRUE) # Winter


######  Year 4  #######

F8_public = self(F7_public.sel, nProgeny = 1, parents = NULL, keepParents = FALSE) #
Fall

TC2 = setPhenoGCA(F7_public.sel, Tester2_private,reps=repTC2,p=pgy) # Fall/Winter

F9_public = self(F8_public, nProgeny = 1, parents = NULL, keepParents = FALSE) #
Winter

TC2 = selectInd(TC2,nInd=15, selectTop = TRUE) # Winter

F8_public.sel = F8_public[F8_public@mother%in%TC2@id]

F9_public.sel = F9_public[F9_public@mother%in%F8_public.sel@id] # Winter


######  Year 3  #######

F6_public = self(F5_public.s0, nProgeny = 3, parents = NULL, keepParents = FALSE) #
Fall

F5_public.s1 = setPheno(F5_public.s0,reps=1, p=pgye) # Fall

F5_public.sel = selectWithinFam(F5_public.s1, nInd = 2, selectTop = TRUE) # Fall

F5_public.sel = selectInd(F5_public.sel, nInd = 150, selectTop = TRUE) # Winter

F6_public.s0 = F6_public[F6_public@mother%in%F5_public.sel@id] # Fall

TC1 = setPhenoGCA(F5_public.sel,Tester1_private,reps=repTC1,p=pgy) # Fall/Winter

TC1 = selectInd(TC1,nInd=100, selectTop = TRUE) # Winter
```

```
F7_public = self(F6_public.s0, nProgeny = 1, parents = NULL, keepParents = FALSE) #
Winter

F6_public.sel = F6_public.s0[F6_public.s0@mother%in%TC1@id]

F6_public.sel = selectWithinFam(F6_public.sel, nInd = 1, selectTop = TRUE) # Fall

F7_public.sel = F7_public[F7_public@mother%in%F6_public.sel@id] # Fall


######  Year 2  #######

F4_public = self(F3_public.s0, nProgeny = 4, parents = NULL, keepParents = FALSE) #
Fall

F3_public.s1 = setPheno(F3_public.s0,reps=1, p=pgye) # Fall

F3_public.sel = selectWithinFam(F3_public.s1, nInd = 2, selectTop = TRUE) # Fall

F4_public.s0 = F4_public[F4_public@mother%in%F3_public.sel@id] # Fall

F5_public = self(F4_public.s0, nProgeny = 4, parents = NULL, keepParents = FALSE) #
Winter

F4_public.s0 = setPheno(F4_public.s0,reps=2, p=pgy) # Winter

F4_public.sel = selectWithinFam(F4_public.s0, nInd = 2, selectTop = TRUE) # Winter

F4_public.sel = selectInd(F4_public.sel, nInd = 300, selectTop = TRUE) # Winter

F5_public.s0 = F5_public[F5_public@mother%in%F4_public.sel@id]


######  Year 1  #######

F1_public = randCross(Parents_public_update, nCrosses_public, ignoreSexes = TRUE)
# Spring

F2_public = self(F1_public, nProgeny = 10, parents = NULL, keepParents = FALSE) #
Fall

F3_public = self(F2_public, nProgeny = 6, parents = NULL, keepParents = FALSE) #
Winter

F2_public = setPheno(F2_public,reps=2, p=pgy) # Winter

F2_public.sel = selectWithinFam(F2_public, nInd = 4, selectTop = TRUE) # Winter
```

```
F3_public.s0 = F3_public[F3_public@mother%in%F2_public.sel@id] # Winter
```

- **Sc.0.1_UpdateParents_public**

```
Parents_public_update = c(Parents_public_update,F7_public.sel)

Parents_public_update = selectInd(Parents_public_update, nInd= nParents_public,
selectTop = TRUE)
```

- **Sc.0.1_UpdateResults_public**

```
hybridMean[year] = meanG(Hybrid)

hybridVar[year] = varG(Hybrid)

parentsMean[year] = meanG(Parents_public_update)

parentsVar[year] = varG(Parents_public_update)

acc1[year] = cor(F3_public.s1@gv,F3_public.s1@pheno)

acc2[year] = cor(F5_public.s1@gv,F5_public.s1@pheno)

geneticGainHyb[year] = hybridMean[year]-hybridMean[1]

geneticGainPar[year] = parentsMean[year]-parentsMean[1]

GainHyb_perCycle[year] = geneticGainHyb[year]/5

GainPar_perCycle[year] = geneticGainPar[year]/5


Efficiency[year] = (parentsMean[year])/(parentsVar)[year])
```

- **Sc.1_Advanced_cycle.BS**

```
######  Year 5  #######

F10_public = self(F9_public.sel, nProgeny = 1, parents = NULL, keepParents = FALSE)
# Fall

TC3 = setPhenoGCA(F9_public.sel, Tester3_private,reps=repTC3,p=pgy) # Winter
```

```
F11_public = self(F10_public, nProgeny = 1, parents = NULL, keepParents = FALSE) # Winter

TC3 = selectInd(TC3,nInd=2, selectTop = TRUE) # Winter

F10_public.sel = F10_public[F10_public@mother%in%TC3@id]

F11_public.sel = F11_public[F11_public@mother%in%F10_public.sel@id] # Winter

Hybrid = hybridCross(F9_public.sel, Tester3_private,crossPlan = "testcross") # Fall

Hybrid = setPheno(Hybrid,reps=repTC3,p=pgy) # Winter

Hybrid = selectInd(Hybrid,nInd=2, selectTop = TRUE) # Winter


######  Year 4  #######

F8_public = self(F7_public.sel, nProgeny = 1, parents = NULL, keepParents = FALSE) # Fall

TC2 = setPhenoGCA(F7_public.sel, Tester2_private,reps=repTC2,p=pgy) # Fall/Winter

F9_public = self(F8_public, nProgeny = 1, parents = NULL, keepParents = FALSE) # Winter

TC2 = selectInd(TC2,nInd=15, selectTop = TRUE) # Winter

F8_public.sel = F8_public[F8_public@mother%in%TC2@id]

F9_public.sel = F9_public[F9_public@mother%in%F8_public.sel@id] # Winter


######  Year 3  #######

F6_public = self(F5_public.s0, nProgeny = 3, parents = NULL, keepParents = FALSE) # Fall

F5_public.s1 = setPheno(F5_public.s0,reps=1, p=pgye) # Fall

F5_public.sel = selectWithinFam(F5_public.s1, nInd = 2, selectTop = TRUE) # Fall

F5_public.sel = selectInd(F5_public.sel, nInd = 150, selectTop = TRUE) # Winter

F6_public.s0 = F6_public[F6_public@mother%in%F5_public.sel@id] # Fall

TC1 = setPhenoGCA(F5_public.sel,Tester1_private,reps=repTC1,p=pgy) # Fall/Winter
```

```
TC1 = selectInd(TC1,nInd=100, selectTop = TRUE) # Winter

F7_public = self(F6_public.s0, nProgeny = 1, parents = NULL, keepParents = FALSE) # Winter

F6_public.sel = F6_public.s0[F6_public.s0@mother%in%TC1@id]

F6_public.sel = selectWithinFam(F6_public.sel, nInd = 1, selectTop = TRUE) # Fall

F7_public.sel = F7_public[F7_public@mother%in%F6_public.sel@id] # Fall


######  Year 2  #######

F4_public = self(F3_public.s0, nProgeny = 4, parents = NULL, keepParents = FALSE) # Fall

F3_public.s1 = setPheno(F3_public.s0,reps=1, p=pgye) # Fall

F3_public.sel = selectWithinFam(F3_public.s1, nInd = 2, selectTop = TRUE) # Fall

F4_public.s0 = F4_public[F4_public@mother%in%F3_public.sel@id] # Fall

F5_public = self(F4_public.s0, nProgeny = 4, parents = NULL, keepParents = FALSE) # Winter

F4_public.s0 = setPheno(F4_public.s0,reps=2, p=pgy) # Winter

F4_public.sel = selectWithinFam(F4_public.s0, nInd = 2, selectTop = TRUE) # Winter

F4_public.sel = selectInd(F4_public.sel, nInd = 300, selectTop = TRUE) # Winter

F5_public.s0 = F5_public[F5_public@mother%in%F4_public.sel@id]


######  Year 1  #######

F1_public = randCross(Parents_public_update, nCrosses_public, ignoreSexes = TRUE) # Spring

F2_public = self(F1_public, nProgeny = 10, parents = NULL, keepParents = FALSE) # Fall

F3_public = self(F2_public, nProgeny = 6, parents = NULL, keepParents = FALSE) # Winter

F2_public = setPheno(F2_public,reps=2, p=pgy) # Winter
```

```
F2_public.sel = selectWithinFam(F2_public, nInd = 4, selectTop = TRUE) # Winter

F3_public.s0 = F3_public[F3_public@mother%in%F2_public.sel@id] # Winter
```

- **Sc.1_UpdateParents.BS**

```
Parents_public_update = F7_public.sel

nP = nParents_public - Parents_public_update@nInd

if(nP>0){

  Parents_public_update = c(Parents_public_update, selectInd(F6_public.sel, nP))

}

  Parents_public_update = selectInd(Parents_public_update, nInd= nParents_public,
selectTop = TRUE)
```

- **Sc.1_UpdateResults.BS**

```
hybridMean[year] = meanG(Hybrid)

hybridVar[year] = varG(Hybrid)

parentsMean[year] = meanG(Parents_public_update)

parentsVar[year] = varG(Parents_public_update)

acc1[year] = cor(F3_public.s1@gv,F3_public.s1@pheno)

acc2[year] = cor(F5_public.s1@gv,F5_public.s1@pheno)

geneticGainHyb[year] = hybridMean[year]-hybridMean[burninYears]

geneticGainPar[year] = parentsMean[year]-parentsMean[burninYears]

GainHyb_perCycle[year] = geneticGainHyb[year]/5

GainPar_perCycle[year] = geneticGainPar[year]/5

Efficiency[year] = (parentsMean[year])/(parentsVar[year])
```

- **Sc.2_Advanced_cycle.GS**

```
######  Year 5  #######

# Fall = Self F7 -> F8 // TestCrosses 3

# Winter = Evaluate TC3 +++ Select F8

F10_public = self(F9_public.sel, nProgeny = 1, parents = NULL, keepParents = FALSE)
# Fall

TC3 = setPhenoGCA(F9_public.sel, Tester3_private,reps=repTC3,p=pgy) # Winter

TC3 = selectInd(TC3,nInd=2, selectTop = TRUE) # Winter

F11_public = self(F10_public, nProgeny = 1, parents = NULL, keepParents = FALSE) #
Winter

F10_public.sel = F10_public[F10_public@mother%in%TC3@id]

F11_public.sel = F11_public[F11_public@mother%in%F10_public.sel@id] # Winter

Hybrid = hybridCross(F9_public.sel, Tester3_private,crossPlan = "testcross") # Fall

Hybrid = setPheno(Hybrid,reps=repTC3,p=pgy) # Winter

Hybrid = selectInd(Hybrid,nInd=2, selectTop = TRUE) # Winter


######  Year 4  #######

# Fall = Self F6 -> F7 // TestCrosses 2

# Winter = Evaluate TC2 +++ Select F7

F8_public = self(F7_public.sel, nProgeny = 1, parents = NULL, keepParents = FALSE) #
Fall

TC2 = setPhenoGCA(F7_public.sel, Tester2_private,reps=repTC2,p=pgy) # Fall/Winter

TC2 = selectInd(TC2,nInd=15, selectTop = TRUE) # Winter

F9_public = self(F8_public, nProgeny = 1, parents = NULL, keepParents = FALSE) #
Winter

F8_public.sel = F8_public[F8_public@mother%in%TC2@id]

F9_public.sel = F9_public[F9_public@mother%in%F8_public.sel@id] # Winter
```

```
######  Year 3  #######

# Fall = Self F5 -> F6 // TestCrosses 1 +++ Evaluate F5 and make selection

# Winter = Evaluate TC1 +++ Select F6

F6_public = self(F5_public.sel, nProgeny = 3, parents = NULL, keepParents = FALSE) # Fall

TC1 = setPhenoGCA(F5_public.sel,Tester1_private,reps=repTC1,p=pgy) # Fall/Winter

TC1 = selectInd(TC1,nInd=100, selectTop = TRUE) # Winter

F7_public = self(F6_public, nProgeny = 1, parents = NULL, keepParents = FALSE) # Winter

F6_public.s0 = F6_public[F6_public@mother%in%TC1@id]

F6_public.sel = selectWithinFam(F6_public.s0, nInd = 1, selectTop = TRUE) # Fall

F7_public.sel = F7_public[F7_public@mother%in%F6_public.sel@id] # Fall


######  Year 2  #######

# Fall = Self F3 -> F4 +++ Evaluate F3 and make selection

# Winter = Self F4 -> F5 ++++ Evaluate F4 and make selection

F4_public = self(F3_public.sel, nProgeny = 4, parents = NULL, keepParents = FALSE) # Fall

F5_public = self(F4_public, nProgeny = 4, parents = NULL, keepParents = FALSE) # Winter

F4_public.s0 = setPheno(F4_public,reps=2, p=pgy) # Winter

F4_public.sel = selectWithinFam(F4_public.s0, nInd = 2, selectTop = TRUE) # Winter

F4_public.sel = selectInd(F4_public.sel, nInd = 300, selectTop = TRUE) # Winter

F5_public.s0 = F5_public[F5_public@mother%in%F4_public.sel@id]

F5_public.s1 = setEBV(F5_public.s0,gsModel) # Fall (genotyping the leaves)

F5_public.sel = selectWithinFam(F5_public.s1, nInd = 1, selectTop = TRUE,use="ebv") # Fall
```

```
F5_public.sel = selectInd(F5_public.sel, nInd = 150, selectTop = TRUE, use="ebv") # Winter


######  Year 1  #######

### Trainning population

TS_pop = randCross(Parents_public_update, nCrosses_TS, ignoreSexes = TRUE) # Spring

TS_pop = setPheno(TS_pop,reps=2,p=pgy) # Winter

gsModel = RRBLUP(TS_pop, use = "gv", snpChip = 1) # Winter

### Advance population

F1_public = randCross(Parents_public_update, nCrosses_public, ignoreSexes = TRUE) # Spring

F2_public = self(F1_public, nProgeny = 10, parents = NULL, keepParents = FALSE) # Fall

F3_public = self(F2_public, nProgeny = 12, parents = NULL, keepParents = FALSE) # Winter

F2_public = setPheno(F2_public,reps=2,p=pgy) # Winter

F2_public.sel = selectWithinFam(F2_public, nInd = 4, selectTop = TRUE) # Winter

F3_public.s0 = F3_public[F3_public@mother%in%F2_public.sel@id] # Winter

F3_public.s1 = setEBV(F3_public.s0,gsModel) # Winter (genotyping the leaves)

F3_public.sel = selectWithinFam(F3_public.s1, nInd = 2, selectTop = TRUE,use="ebv") # Fall
```

- **Sc.2_UpdateParents.GS**

```
Parents_public_update = F7_public.sel

nP = nParents_public - Parents_public_update@nInd

if(nP>0){

  Parents_public_update = c(Parents_public_update, selectInd(F6_public.sel, nP))
```

```
}

Parents_public_update = selectInd(Parents_public_update, nInd= nParents_public,
selectTop = TRUE)
```

- **Sc.2_UpdateResults.GS**

```
hybridMean[year] = meanG(Hybrid)

hybridVar[year] = varG(Hybrid)

parentsMean[year] = meanG(Parents_public_update)

parentsVar[year] = varG(Parents_public_update)

acc1[year] = cor(F3_public.s1@gv,F3_public.s1@ebv)

acc2[year] = cor(F5_public.s1@gv,F5_public.s1@ebv)

geneticGainHyb[year] = hybridMean[year]-hybridMean[burninYears]

geneticGainPar[year] = parentsMean[year]-parentsMean[burninYears]

GainHyb_perCycle[year] = geneticGainHyb[year]/5

GainPar_perCycle[year] = geneticGainPar[year]/5

Efficiency[year] = (parentsMean[year])/(parentsVar[year])
```

- **Sc.3_Advanced_cycle.DH**

```
########  Year 4  ########

TC3_DH = setPhenoGCA(TC2_DH,Tester3_private,reps=repTC3,p=pgy)

TC3_DH = selectInd(TC3_DH,nInd=2)

Hybrid = hybridCross(TC2_DH,Tester3_private,crossPlan = "testcross")

Hybrid = setPheno(Hybrid,reps=repTC3,p=pgy)

Hybrid = selectInd(Hybrid,nInd=2)


########  Year 3  ########
```

```
TC2_DH = setPhenoGCA(TC1_DH,Tester2_private,reps=repTC2,p=pgy)
TC2_DH = selectInd(TC2_DH,nInd=20)

######## Year 2 ########
# Spring = Multiply DH seeds in optimized conditions
DH_Pub2 = self(DH_Pub1, nProgeny = 1, parents = NULL, keepParents = FALSE)
# Fall = Multiply seeds and exclude the bad DH plants
# Fall = Generate testcross 1
DH_Pub2.1 = setPheno(DH_Pub2,reps=1,p=pgy)
DH_Pub3 = selectInd(DH_Pub2.1,nInd=800)
TC1_DH = setPhenoGCA(DH_Pub3,Tester1_private,reps=repTC1,p=pgy)
# Winter = Evaluate testcross 1
TC1_DH = selectInd(TC1_DH,nInd=100)

######## Year 1 ########
# Spring = Cross Parental Lines
F1_public = randCross(Parents_public_update, nCrosses_public, ignoreSexes = TRUE)
# Fall and Winter = Generate DH seeds
DH_Pub1 = makeDH(F1_public, nDH = pub_DH, keepParents = FALSE)
```

- **Sc.3_UpdateParents.DH**

```
Parents_public_update = TC1_DH
Parents_public_update = selectInd(Parents_public_update, nInd= nParents_public)
```

- **Sc.3_UpdateResults.DH**

```
hybridMean[year] = meanG(Hybrid)
hybridVar[year] = varG(Hybrid)
parentsMean[year] = meanG(Parents_public_update)
parentsVar[year] = varG(Parents_public_update)
acc1[year] = cor(DH_Pub2.1@gv,DH_Pub2.1@pheno)
acc2[year] = cor(DH_Pub2.1@gv,DH_Pub2.1@pheno)
geneticGainHyb[year] = hybridMean[year]-hybridMean[burninYears]
geneticGainPar[year] = parentsMean[year]-parentsMean[burninYears]
GainHyb_perCycle[year] = geneticGainHyb[year]/4
GainPar_perCycle[year] = geneticGainPar[year]/4
Efficiency[year] = (parentsMean[year])/(parentsVar[year])
```

- **Sc.4_Advanced_cycle.DHGS**

```
########  Year 4  ########
TC3_DH = setPhenoGCA(TC2_DH,Tester3_private,reps=repTC3,p=pgy) # Fall
TC3_DH = selectInd(TC3_DH,nInd=2) # Winter
Hybrid = hybridCross(TC2_DH,Tester3_private,crossPlan = "testcross") # Fall
Hybrid = setPheno(Hybrid,reps=repTC3,p=pgy) # Winter
Hybrid = selectInd(Hybrid,nInd=2) # Winter
########  Year 3  ########
TC2_DH = setPhenoGCA(TC1_DH,Tester2_private,reps=repTC2,p=pgy)
TC2_DH = selectInd(TC2_DH,nInd=20) # Winter

########  Year 2  ########
# Spring = Multiply DH seeds in optimized conditions
```

```
DH_Pub2 = self(DH_Pub1, nProgeny = 1, parents = NULL, keepParents = FALSE)

# Fall = Multiply seeds and exclude the bad DH plants

# Fall = Generate testcross 1

DH_Pub2.1 = setEBV(DH_Pub2,gsModel)

DH_Pub3 = selectInd(DH_Pub2.1,nInd=800,use="ebv")

TC1_DH = setPhenoGCA(DH_Pub3,Tester1_private,reps=repTC1,p=pgy)

# Winter = Evaluate testcross 1

TC1_DH = selectInd(TC1_DH,nInd=100)


########  Year 1  ########

### Training population

TS_pop = randCross(Parents_public_update, nCrosses_TS, ignoreSexes = TRUE) #
Spring

TS_pop = setPheno(TS_pop,reps=2,p=pgy) # Winter

gsModel = RRBLUP(TS_pop, use = "gv", snpChip = 1) # Winter

### Advance population

# Spring = Cross Parental Lines

F1_public = randCross(Parents_public_update, nCrosses_public, ignoreSexes = TRUE)

# Fall and Winter = Make DH

DH_Pub1 = makeDH(F1_public, nDH = pub_DH, keepParents = FALSE)
```

- **Sc.4_UpdateParents.DHGS**

```
Parents_public_update = TC1_DH

Parents_public_update = selectInd(Parents_public_update, nInd= nParents_public)
```

- **Sc.4_UpdateResults.DHGS**

```
hybridMean[year] = meanG(Hybrid)

hybridVar[year] = varG(Hybrid)

parentsMean[year] = meanG(Parents_public_update)

parentsVar[year] = varG(Parents_public_update)

acc1[year] = cor(DH_Pub2.1@gv,DH_Pub2.1@ebv)

acc2[year] = cor(DH_Pub2.1@gv,DH_Pub2.1@ebv)

geneticGainHyb[year] = hybridMean[year]-hybridMean[burninYears]

geneticGainPar[year] = parentsMean[year]-parentsMean[burninYears]

GainHyb_perCycle[year] = geneticGainHyb[year]/4

GainPar_perCycle[year] = geneticGainPar[year]/4

Efficiency[year] = (parentsMean[year])/(parentsVar[year])
```

- **Sc.5_Advanced_cycle.DH**

```
########  Year 3  ########

# Fall = Make TestCross 2 crosses + Advance Private DH in selfing

TC3_DH = setPhenoGCA(TC2_DH,Tester3_private,reps=repTC3,p=pgy)

#Winter = Evaluate TC3 in Belle-Glade  ((Test-Cross 3 = 5 testers (best/oldest elite
inbreds) in 60 environments))

TC3_DH = selectInd(TC3_DH,nInd=2)

Hybrid = hybridCross(TC2_DH,Tester3_private,crossPlan = "testcross")

Hybrid = setPheno(Hybrid,reps=repTC3,p=pgy)

Hybrid = selectInd(Hybrid,nInd=2)


########  Year 2  ########

# Spring = Multiply DH seeds through selfing

DH_Pub2 = self(DH_Pub1, nProgeny = 1, parents = NULL, keepParents = FALSE)
```

```
# Fall = Multiply seeds and exclude the bad DH plants
# Fall = Generate testcross 2
DH_Pub2.1 = setPheno(DH_Pub2,reps=2,p=pgy)
DH_Pub3 = selectInd(DH_Pub2.1,nInd=200)
TC2_DH = setPhenoGCA(DH_Pub3,Tester2_private,reps=repTC2,p=pgy)
# Winter = Evaluate TC2
TC2_DH = selectInd(TC2_DH,nInd=30)

########  Year 1  ########
# Spring = Cross Parental Lines
F1_public = randCross(Parents_public_update, nCrosses_public, ignoreSexes = TRUE)
# Fall and Winter = Make DH
DH_Pub1 = makeDH(F1_public, nDH = pub_DH, keepParents = FALSE)
```

- **Sc.5_UpdateParents.DH**

```
nP=nParents_public-nInd(TC2_DH)
tmp = DH_Pub3[!(DH_Pub3@id%in%TC2_DH@id)]
Parents_public_update = c(TC2_DH,selectInd(tmp, nInd= nP))
```

- **Sc.5_UpdateResults.DH**

```
hybridMean[year] = meanG(Hybrid)
hybridVar[year] = varG(Hybrid)
parentsMean[year] = meanG(Parents_public_update)
parentsVar[year] = varG(Parents_public_update)
acc1[year] = cor(DH_Pub2.1@gv,DH_Pub2.1@pheno)
```

```
acc2[year] = cor(DH_Pub2.1@gv,DH_Pub2.1@pheno)
geneticGainHyb[year] = hybridMean[year]-hybridMean[burninYears]
geneticGainPar[year] = parentsMean[year]-parentsMean[burninYears]
GainHyb_perCycle[year] = geneticGainHyb[year]/3
GainPar_perCycle[year] = geneticGainPar[year]/3
Efficiency[year] = (parentsMean[year])/(parentsVar[year])
```

- **Sc.6_Advanced_cycle.DHGS**

```
########  Year 3  ########
TC3_DH = setPhenoGCA(TC2_DH,Tester3_private,reps=repTC3,p=pgy) # Fall
TC3_DH = selectInd(TC3_DH,nInd=2) # Winter
Hybrid = hybridCross(TC2_DH,Tester3_private,crossPlan = "testcross") # Fall
Hybrid = setPheno(Hybrid,reps=repTC3,p=pgy) # Winter
Hybrid = selectInd(Hybrid,nInd=2) # Winter

########  Year 2  ########
# Spring = Multiply DH seeds in optimized conditions
# Spring = Make selection of the top DH based on EBV
DH_Pub2 = self(DH_Pub1, nProgeny = 1, parents = NULL, keepParents = FALSE)
DH_Pub2.1 = setEBV(DH_Pub2,gsModel)
DH_Pub3 = selectInd(DH_Pub2.1,nInd=200,use="ebv")
# Fall = Generate testcross 2
TC2_DH = setPhenoGCA(DH_Pub3,Tester2_private,reps=repTC2,p=pgy)
# Winter = Evaluate testcross 2
TC2_DH = selectInd(TC2_DH,nInd=30)
```

```
########  Year 1  ########

### Trainning population

TS_pop = randCross(Parents_public_update, nCrosses_TS, ignoreSexes = TRUE) #
Spring

TS_pop = setPheno(TS_pop,reps=2,p=pgy) # Winter

gsModel = RRBLUP(TS_pop, use = "gv", snpChip = 1) # Winter

### Advance population

# Spring = Cross Parental Lines

F1_public = randCross(Parents_public_update, nCrosses_public, ignoreSexes = TRUE)

# Fall and Winter = Make DH

DH_Pub1 = makeDH(F1_public, nDH = pub_DH, keepParents = FALSE)
```

- **Sc.6_UpdateParents.DHGS**

```
nP=nParents_public-nInd(TC2_DH)

tmp = DH_Pub3[!(DH_Pub3@id%in%TC2_DH@id)]

Parents_public_update = c(TC2_DH,selectInd(tmp, nInd= nP))
```

- **Sc.6_UpdateResults.DHGS**

```
hybridMean[year] = meanG(Hybrid)

hybridVar[year] = varG(Hybrid)

parentsMean[year] = meanG(Parents_public_update)

parentsVar[year] = varG(Parents_public_update)

acc1[year] = cor(DH_Pub2.1@gv,DH_Pub2.1@ebv)
```

```
acc2[year] = cor(DH_Pub2.1@gv,DH_Pub2.1@ebv)

geneticGainHyb[year] = hybridMean[year]-hybridMean[burninYears]

geneticGainPar[year] = parentsMean[year]-parentsMean[burninYears]

GainHyb_perCycle[year] = geneticGainHyb[year]/3

GainPar_perCycle[year] = geneticGainPar[year]/3

Efficiency[year] = (parentsMean[year])/(parentsVar[year])
```