APPLICATIONS AND EVALUATION OF PHENOMIC TECHNOLOGIES IN MAIZE

BREEDING

A Thesis

by

HOLLY MARIE LANE

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Seth C. Murray
Committee Members,	Jamie Foster
	Russell W. Jessup
	Cristine L. S. Morgan
Head of Department,	David D. Baltensperger

December 2019

Major Subject: Plant Breeding

ABSTRACT

High-throughput phenotyping technologies can generate large volumes of data at low costs and are useful in a wide range of plant sciences, including plant breeding. Many phenotyping technologies increase the rate of currently measured traits, while others can detect completely novel traits yet to be fully explored.

In addition to replacing current manual measures, higher throughput technologies may be used to indirectly predict yield. Here, we explore this concept, using high-throughput phenotype information from Fourier Transformed near-infrared reflectance spectroscopy (NIRS) of harvested kernels to predict parental grain yield in maize (*Zea mays* L.). A dataset of 2,563 kernel samples from a diversity panel of 346 hybrid testcrosses were scanned using an FT-NIRS interferometer, measuring 3,076 wavenumbers (bands) in the range of 4,000-10,000 cm^{-1}. Corresponding grain yield for each sample was used to train predictive models using three types of statistical learning: (a) partial least square regression (PLSR), (b) NIRS best linear unbiased predictors (NIRS BLUP) and (c) functional regression. These results found NIRS spectral data to be a useful tool in predicting maize grain yield of unknown samples and showed promising results for evaluating genetically independent breeding populations. High correlations between predicted and observed values demonstrated value for grain NIRS in ranking variety yields relative to one another, even where yield predictions were not accurate. More research in this area will provide better understanding for how NIRS and other phenomic technologies can be used to predict phenotypes in breeding programs.

The second project of this thesis explored the tradeoffs that exist between using high throughput phenotyping technologies with a higher error rate compared to a traditional phenotyping method with higher accuracy, with lower sampling rates. To accomplish this, populations were simulated to include variance from genetic, environmental, other effects, and error; different analysis scenarios were then created and assessed. It was determined that additional environmental sampling could compensate for greater measurement error; while for genetic mapping, increasing population size was most important to correctly predict genetic loci. This analysis provided an understanding of the threshold for high-throughput sample number increases needed to obtain a benefit over more accurate traditional measures.

DEDICATION

This thesis is dedicated to all the mentors who encouraged me to pursue science; especially to Dr. Michael Neff for sparking my journey into the world of plant genetics, to Dr. Emilio Oyarzabal for the push to pursue a graduate degree, to Gabriela Burian for her continued encouragement and positive spirit, and to Dr. Seth Murray for shaping me into the researcher I am today. Also, to my parents, sisters, grandparents, and all of my family members who have always believed in me and supported me to follow my dreams. Thank you also to my dear friends, Joseph Burke, Andrea Fox, Shelby Westenskow, and Andrea Welchons for being there for me throughout this degree, near or far.

ACKNOWLEDGEMENTS

CONTRIBUTORS AND FUNDING SOURCES

Contributors

Funding Sources

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Aims

This work primarily aims to develop and explore the value of phenomics towards improving trait estimations in plant breeding. To this end, a phenomic prediction approach using Near Infrared Reflectance Spectroscopy (NIRS) analogous to the genotyping data in a genomic prediction model was explored in maize. Ultimately, the goal is to develop models which could correctly predict which experimental varieties (hybrids or parental inbreds) should be kept and advanced, and those which should be discarded. However, for any such model, it is not-clear if there is sufficient discriminating information in non-target traits to accurately predict target traits, like yield. If successful, it is also unclear whether trait prediction is based on biological process relationships, or based on identifying genetic relationships between lines. Like many statistical prediction approaches, these questions require large amounts of data to evaluate. However, this large volume of phenomic data is likely to be of lower precision and accuracy than traditional measures. This creates a need to explore the trade-offs between high-throughput, and high accuracy or precision in phenotypic data collection. In other words, can low accuracy in sampling be overcome via a high rate of sampling? To address this, modeling is required to understand measurement trade-offs, and how they could affect the evaluation methods of a breeding program.

Plant Breeding

Maize Domestication

Globally, the most important food class is cereal grains, contributing to almost half of human caloric intake. Of those, maize, wheat, and rice make up most of the consumption, totaling 94% (Ranum et al., 2014). In particular, the domestication of maize, or corn (*Zea mays L.*), likely began about 9,000 years ago (Kennett et al., 2017). According to molecular data, it appears a single domestication event resulted in the maize grown today, whose current closest relative is a subspecies of teosinte, *Zea mexicana* (Schrad.) Kuntze (Matsuoka et al., 2002). Since domestication, humankind has continued to improve maize yield and other qualities of value for consumption and production. According to the FAO, an estimated one billion tonnes of corn were produced globally in 2016 (FAOSTAT, 2016). Maize and other crops yields have increased in the US as a result of numerous factors, including improved management, increased irrigation, improved weed and pest management techniques, and most relevant to this study, breeding (Duvick, 2005; Assefa et al., 2017). The best estimate is that 50% of maize improvement has been contributed by genetics and breeding, using Duvick's so-called era studies (Duvick, 1992).

A New Era of Plant Breeding

There have been at least three defined eras of plant breeding since the first plant domestication of over 10,000 years ago (Li et al., 2012; Lee and DeVore, 1968). The first era consisted of our ancestors using only traits that were visible to the naked eye, forming the basis of phenotypic selection. In the 20th century, there was a shift into the scientific era, where the use of heritability was first applied to the study of plants. Finally, this was followed

by the genomics era, when the focus shifted to molecular techniques, aimed at better elucidating the genetics behind heritability of phenotypic traits (Li et al., 2012). In maize, yields did not have substantial improvements until the late 1930s, where seed production moved from open pollination into the double hybrid, and then single cross hybrid breeding schemes, this is referred to as the era of hybrids, in maize (Hallauer et al., 2010). Biotech traits further improved yields, beginning in the 1990s, enabled by both improved genetics techniques and germplasm (Hallauer et al., 2010).

The genomics era has dominated the attention of the field of plant breeding since the 1990s. Improvements in molecular technologies have consumed large amounts of resources and contributed to large amounts of data. Presently, it seems there is yet another shift in the world of breeding, where once again, phenotypes are becoming a major focus of analysis. This new *phenomics* era is characterized by an increased shift in emphasizing phenotypic traits on a large scale. While a phenotype describes a characteristic that is morphological, biogeochemical, etc., the word *phenome* refers to the phenotype as a whole, or the expression of a genome in an environment, and usually results in high-dimensional data (Kumar et al., 2015). Many phenomic technologies improve upon existing data collection methods (i.e. using a drone to measure plant height), whereas some investigate novel phenotypes altogether (i.e. using near-infrared spectroscopy or hyperspectral cameras). These phenomic datasets have the potential to complement, or even eventually replace, many technologies of the genomics era.

High Throughput Phenotyping

In addressing phenotyping bottlenecks experienced in breeding programs, technologies are aiming to increase the throughput of traditional methods. The concepts of phenotyping have even evolved from the direct measurement of the trait of interest, to investigations surrounding more indirect measures of overall plant growth or spectral signatures, which may be inferred via remote or proximal sensing (Cabrera-Bosquet et al., 2012). Reductions in costs and technological hurdles have increased agricultural interest in the use of unoccupied aerial systems over the past decade (UAS, i.e. drones) opening up opportunities for new types of data collection by researchers and farmers (Shi et al., 2016), though field data collection remains challenging to collect (Araus and Cairns, 2014). When UAS campaigns successfully collect images over breeding programs, especially temporally over a growing season, many features from the images can be extracted. Some of these features may correspond to traditionally measured phenotypes such as plant height (Pugh et al., 2018), which can be incorporated into physiological growth models to extract novel, more robust, phenotypes (Anderson et al., 2019). Researchers have excitingly begun to incorporate many features into statistical models that can predict end yield, such as plant height (Anderson et al., 2019) and vegetation indices (Sun et al., 2019), already known to have some relationship with yield. In future work, it is likely that hundreds or thousands of features from UAS temporal flights and hyperspectral sensors could be identified that consistently differentiate between cultivars in the field; most such features have not yet been evaluated for their abilities to predict end yield (Shakoor et al., 2019).

As technologies for capturing phenotypic data continue to advance, the size of the datasets increases. These increasingly large datasets bring novel data management challenges, and new methods are becoming necessary for appropriately utilizing phenomic data. High density datasets demand improved data storage, processing, and analysis methodologies. Lots of data can provide lots of information, given the best protocols are established to mine that data.

Furthermore, this increase of high throughput phenotyping has brought with it the promise of reductions in resource-intensive manual measurements. However, the accuracy of these new methods must be properly explored for these hopes to be realized. Can manual measurements truly become a technique of the past? These developing technologies offer the capacity to increase the data collected while limiting human labor, but the possible tradeoffs between accuracy and throughput have yet to be fully explored or understood in the context of breeding programs.

References

Anderson, S.L., S.C. Murray, L. Malambo, C. Ratcliff, S. Popescu, et al. 2019. Prediction of Maize Grain Yield before Maturity Using Improved Temporal Height Estimates of Unmanned Aerial Systems. The Plant Phenome Journal 2(1). doi: 10.2135/tppj2019.02.0004.

Assefa, Y., P.V.V. Prasad, P. Carter, M. Hinds, G. Bhalla, et al. 2017. A New Insight into Corn Yield:Trends from 1987 through 2015. Crop Science 57(5): 2799–2811. doi: 10.2135/cropsci2017.01.0066.

Duvick, D.N. (Iowa S.U. of S. and T. 1992. Genetic contributions to advances in yield of US maize. Maydica (Italy) 37: 69–79.

Duvick, D.N. 2005. The Contribution of Breeding to Yield Advances in maize (Zea mays L.). Advances in Agronomy. Academic Press. p. 83–145

FAOSTAT data. 2016. Food and Agriculture Organization of the United Nations.

Hallauer, A.R., J.B.M. Filho, and M.J. Carena. 2010. Breeding Plans. In: Carena, M.J., Hallauer, A.R., and Miranda Filho, J.B., editors, Quantitative Genetics in Maize Breeding. Springer New York, New York, NY. p. 577–653

Kennett, D.J., H.B. Thakar, A.M. VanDerwarker, D.L. Webster, B.J. Culleton, et al. 2017. High-precision chronology for Central American maize diversification from El Gigante rockshelter, Honduras. PNAS 114(34): 9026–9031. doi: 10.1073/pnas.1705052114.

Kumar, J., A. Pratap, and S. Kumar. 2015. Plant Phenomics: An Overview. In: Kumar, J., Pratap, A., and Kumar, S., editors, Phenomics in Crop Plants: Trends, Options and Limitations. Springer India, New Delhi. p. 1–10

Lane, H.M., S.C. Murray, J. Crossa, O.A. Montesinos-López, A. Montesinos-López, et al. *In Review*. Phenomic Selection and Prediction of Maize Grain Yield from Near-infrared Reflectance Spectroscopy of Kernels. The Plant Phenome Journal.

Lee, R.B., and I. DeVore. 1968. Problems in the Study of Hunters and Gatherers. Man The Hunter. ALDINE DE GRUYTER, New York, NY. p. 3

Li, X., C. Zhu, J. Wang, and J. Yu. 2012. Chapter six - Computer Simulation in Plant Breeding. In: Sparks, D.L., editor, Advances in Agronomy. Academic Press. p. 219–264

Matsuoka, Y., Y. Vigouroux, M.M. Goodman, J.S. G, E. Buckler, et al. 2002. A single domestication for maize shown by multilocus microsatellite genotyping. PNAS 99(9): 6080–6084. doi: 10.1073/pnas.052125199.

Ranum, P., J.P. Peña-Rosas, and M.N. Garcia-Casal. 2014. Global maize production, utilization, and consumption. Annals of the New York Academy of Sciences 1312(1): 105–112. doi: 10.1111/nyas.12396.

Rincent, R., J.-P. Charpentier, P. Faivre-Rampant, E. Paux, J.L. Gouis, et al. 2018. Phenomic Selection Is a Low-Cost and High-Throughput Method Based on Indirect Predictions: Proof of Concept on Wheat and Poplar. G3: Genes, Genomes, Genetics 8(12): 3961–3972. doi: 10.1534/g3.118.200760.

CHAPTER II

PHENOMIC SELECTION AND PREDICTION OF MAIZE GRAIN YIELD FROM NEAR-

INFRARED REFLECTANCE SPECTROSCOPY OF KERNELS

Introduction

Advancement of plant breeding

Modern plant breeding is conducted very differently from the crop domestication of over 10,000 years ago, and even from the first scientific breeding of 100 years ago (Lee and DeVore, 1968). As the science of plant breeding has progressed, a major focus has been on genomic tools for crop improvement, with key technological innovations being the primary driver of advancements. Over time, breeding has become faster, by decreasing cycle times (i.e. growing plants in off-season nurseries to reach multiple generations within a year), and has become more targeted, through marker-assisted selection and other molecular techniques (Brummer et al., 2011). Most recently, genomic prediction and selection approaches have shown to have practical value and have become extensively relied on for decision making in private and public sector breeding programs (Zhao et al., 2015; Cooper et al., 2004; Crossa et al., 2017). In general, technologies to better understand and collect genetic data have vastly improved over the past few decades (Levy and Myers, 2016).

In contrast, the methods employed to collect and quantify phenotypic data, have not seen the same rate of technological advancement (Araus and Cairns, 2014). Rincent et al. (2012) and others have noted that phenotyping is the limiting factor in genomic selection and other breeding methods. Evaluating plants in the field is the single most costly and complex aspect of a breeding program, in terms of time, labor, and money, but is a key step in most

8

accurately estimating genetic yield potential and training genomic models (Bernardo, 2008; Bernardo and Yu, 2007). In addressing phenotyping bottlenecks, efforts are expanding to better collect and analyze phenotypic data on a wide scale. Recent findings suggest that improving phenotyping methods may not just simply improve genomic selection or other genetics-based techniques, but could potentially reduce or eliminate the need for genotyping in some cases (Rincent et al., 2018). Evaluating the thousands of features provided by Near-Infrared Reflectance Spectroscopy (NIRS) may serve as a guide to utilizing many kinds of dense phenotypic datasets.

Near-Infrared Reflectance Spectroscopy (NIRS)

Near-infrared reflectance spectroscopy (NIRS) has a long history in agricultural settings for evaluating chemical composition since the 1960s, with some of the first mentions of the technology being used to quantify seed oil composition (Morris and Holman, 1961). NIRS is now commonly used to estimate characteristics of grain or seeds including protein, starch, and oil content (Chen et al., 2013; Ferreira et al., 2013; Murray et al., 2008a; Silva et al., 2008) as well as new composition traits such as grain phenolics (Meng et al., 2015 and many others). NIRS composition estimates are commonly made on plant tissue, such as forage and silage, to estimate bioenergy potential or livestock feeding value (Wolfrum et al., 2013; Murray et al., 2008b). NIRS has a sound grounding in physics, as near-infrared light at specific wavelengths is absorbed by specific chemical bonds that make up the components of living tissues. Furthermore, the relative proportion of each of these bonds within the tissue quantitatively influence the nature of the absorbance of light at different wavenumbers (Foley et al., 1998). Typically however, no single region of the near-infrared spectrum can be

attributed to a single chemical component in living tissues, because plant and animal tissues exist as heterogeneous mixtures of compounds, making direct inference from spectra challenging (Foley et al., 1998). Statistical analysis is needed to extract meaningful information from NIRS on biological samples. Key among the benefits of NIRS is that there is no need to contact, damage, or destroy samples. Consequently, samples can be analyzed and still be available for planting or other analyses.

Recently, NIRS has been shown to be useful as a high throughput phenotype for predicting grain yield and other end quality traits (Rincent et al., 2018). At the molecular level, it has been shown that NIRS can be an effective cell-wall mutant screening tool in *Arabidopsis thaliana* as well as in sorghum (*Sorghum bicolor* L.) or maize (*Zea mays* L.) research programs (Chen et al., 1998; Vermerris et al., 2007). While previous studies have shown the ability of NIRS to quantify composition, understand chemical bonds, differentiate between species, varieties, or even genetics (Bertrand et al., 1985; Espinoza et al., 2012; Lang et al., 2017), using NIRS information to perform selections in the context of breeding for grain yield remains relatively novel, since grain yield is not a chemical component. However, NIRS has been previously shown to correlate with yield. Ferrio et al. (2004) investigated grain yield correlation with NIRS of durum wheat flour. Despite high r values, the slopes of their predicted and measured yield values were not in unity, causing them to conclude that the ability of NIRS to predict grain yield in wheat was not an accurate way to achieve estimations under their conditions. They partially attributed these challenges to a lack of strong previous calibrations under the same growing conditions of the samples.

The concept of treating NIRS data as markers, and using them to make inferences about relatedness, is a recent phenomenon in breeding. Rincent et al. (2018) used NIRS wavelength data on wheat (*Triticum aestivum* L.) grain and leaf tissue as well as poplar (*Populus nigra* L.) wood to develop NIRS best linear unbiased predictions (NIRS BLUP, an NIRS similarity matrix); the authors surprisingly obtained better prediction accuracy than using G-BLUP (genomic similarity matrix). Krause et al. (2019) had success in wheat as well, developing relationship matrices from hyperspectral data collected via UAS and using them in yield predictions alongside markers and pedigree information.

Phenomic prediction and selection for grain yield from grain or plant spectra

The theory and the term of phenomic selection was first introduced by Rincent et al. (2018), as the practice of evaluating and selecting lines via phenotypic variables. Essentially, phenotypic, or phenomic, variables replace genetic marker information in traditional selection methodology. The present study serves to independently validate and generalize some of the concepts of Rincent et al.'s (2018) findings about the predictive capacity of NIRS, using phenomic prediction (plot basis) and phenomic selection (entry basis).

The main objective of this study was to apply models and methods to the NIRS data for the phenomic prediction and selection of grain yield. Specifically, our aim was to predict parental grain yield from existing grain scans of a hybrid maize diversity panel grown in several environments (Barrero-Farfan et al., 2015; Warburton et al., 2013; Flint-Garcia et al., 2005) and apply these calibrations on additional independent breeding populations.

We sought to evaluate different statistical models: (1) partial least square regression (PLSR); (2) NIRS BLUP, employing the values of NIRS instead of markers as proposed by

Rincent et al. (2018); and (3) functional regression analyses using the NIRS values as covariates in a G×E, multi-environment model. In this case, PLSR was used as a *phenomic prediction*, where yield is predicted on a plot basis, with NIRS being used for each individual plot. This is analogous to how composition of grain samples would be predicted. The NIRS BLUP and functional regression models were used as *phenomic selection*, where predictions were made on an entry basis, with NIRS being averaged within pedigrees. This is analogous to how a breeding program predicts breeding values using marker information.

Different cross-validations schemes were used to ensure prediction accuracy estimates would be relevant to future breeding activities. These schemes addressed 1) predicting known genotypes in unknown environments, 2) predicting unknown genotypes in known environments, and 3) so-called sparse testing, where known genotypes are predicted in known environments in which they are not observed.

If NIRS can be used to estimate the performance (grain yield) of a genotype, either on a plot or entry basis, it therefore stands to reason that the spectra also capture some of the value of that material. Successful models for a trait of interest suggest that phenomic selection methods using NIRS could be useful in maize, or other hybrid crop, breeding programs.

Functional regression analyses for high throughput phenotyping data

High throughput phenotyping technologies, including NIRS, are increasingly available in agriculture and generate large volumes of data. Many such phenotypes are multi-dimensional and these data can be represented as functions. Functional data analysis is a field of study that deals with the analysis and theory of data whose units of observation are functions (curves) defined as continuous domain (Morris, 2015). Recent studies by

Montesinos-Lopez et al. (2017a, b) present functional regression models to develop prediction equations for wheat grain yield and other traits using hyperspectral crop image data from the field. This work showed that functional regression models could provide yield predictions with similar and, in some cases, higher predictive power than that of conventional regression techniques. Montesinos-Lopez et al. (2018) described details for implementing Bayesian functional regression using the developed genomic functional regression package.

Materials and Methods

Experimental Design and Germplasm

A set of 346 diverse hybrid lines were grown as a structured genome-wide association study (GWAS) to assess aflatoxin resistance, drought tolerance, and other agronomic traits such as yield in 2011 and 2012 (Barrero-Farfan et al., 2015). These lines originated from a subset of the USDA-Goodman maize association panel (Flint-Garcia et al., 2005), as well as the Southern subtropical Williams/Warburton panel (Warburton et al., 2013). This panel was crossed to two isogenic lines of Tx714, a high-yielding, Southern U.S. bred stiff-stalk line that is more than 95% identical to its stiff-stalk relative, B73 (Romay et al., 2013). Each isogenic hybrid was grown under one or two experimental conditions (well-watered (WW) and limited irrigation, or water stress (WS)) with two replicates in a randomized complete block design, where seed was available. A full description of the hybrids and experimental design is in Barrero-Farfan et al. (2015). Since the two isogenic lines did not differ in grain yield, the data from both were combined and treated as the same hybrid. This population was used for all model training and for validation.

Other elite hybrid breeding trials in the Texas A&M breeding program were also used as an independent validation set for evaluating prediction accuracy. These trials, grown in 2011, consisted of hybrids with no known or expected relatedness to the original material presented above, most being sub-tropical derived lines crossed to U.S. commercial stiff stalk hybrids (Murray et al., 2019). These tests consist of relevant breeding material within the program, as opposed to the training data, which represented a genetic study. NIRS was collected for each test on a plot basis. These tests were grown to assess aflatoxin (AF) resistance. Material from within the program represented four tests, and are referred to in this study as 1AF, 2AF, 3AF, and 4AF. Two other tests were also assessed, which include breeding material from other programs. The first is Southeast Regional Aflatoxin Trials (SERAT; Wahl et al., 2017), and the next consists of germplasm enhancement of maize (GEM) lines.

Three of these validation tests (4AF, SERAT, and GEM) were grown in College Station, TX under similar conditions, while, 1AF and 2AF were grown in Weslaco, TX, and 3AF was grown in Corpus Christi, TX. All together, these datasets combined represent 200 pedigrees across three Texas locations from 2011 (679 scans total), most of which is breeding material. This set of tests was used as a practical validation of how broadly the trained models could be used in a breeding program, or if more extensive calibration would likely be necessary.

Spectral Data

NIRS was performed on harvested grain from Barrero-Farfan et al. (2015) hybrids during the completion of the initial study. In total, 2,563 such whole kernel samples were scanned using Fourier transformed near-infrared reflectance spectroscopy. Spectra were

14

collected with a Thermo Anteris II FT Interferometer, where approximately 175g of bulked grain, acclimated to a climate-controlled room, were scanned 128 times, resulting in an average (full description of methodology in Meng et al., 2015). The spectra consist of every other wavenumber (i.e. band) from 4,000 to 10,000 cm^{-1} (a range of 1,000-2,500 nm).

Data Processing

In 2011, very low yields were observed in all experiments due to record heat and drought. The lowest yielding of the plots appeared remarkably atypical, so a threshold of at least 62 g m^{-2} (10 bu ac^{-1}) per plot was set. Applying the threshold, 47 plots were removed, leaving 2,516 samples with an average yield of 559 g m^{-2}, and a range of 63 to 1462 g m^{-2}. Furthermore, as an objective method to eliminate bad NIRS, an algorithm was implemented in R (R Core Team, 2018) to eliminate scans that had shapes perceptively different from other scans. First, the difference of each spectra from the average spectra across each wavenumber was calculated; next, the sum of the absolute value of those differences were taken; and finally scans with large, outlying differences from the average were removed. This method successfully eliminated spectra with abnormal shapes ("bad scans"), as opposed to abnormal predictions within the model. This, presumably, was an unbiased approach to identifying these bad scans. This procedure removed 24 sample scans leaving 2,492 samples for final analyses.

The cleaned scans were loaded into R and processed with the *prospectr* package (Stevens and Ramirez-Lopez, 2014) to obtain the first and second derivatives, using the Savitsky-Golay method and a window size of 37 (Fig. 1). This transformation served as a

normalization from the raw scans, and mainly accounted for differences in albedo (overall reflectance) between samples. Interferometers, as used in this study, generally have a higher spectral resolution than other methods, often associated with a smaller ratio of signal-to-noise. Therefore, it has been recommended to use smoothing derivation methods such as the Savitsky-Golay method as opposed to a simpler, finite-difference derivative method (Rinnan *et al*, 2009). Ultimately, the first derivative proved to offer the best results in cross validation, so this was used across all model types.
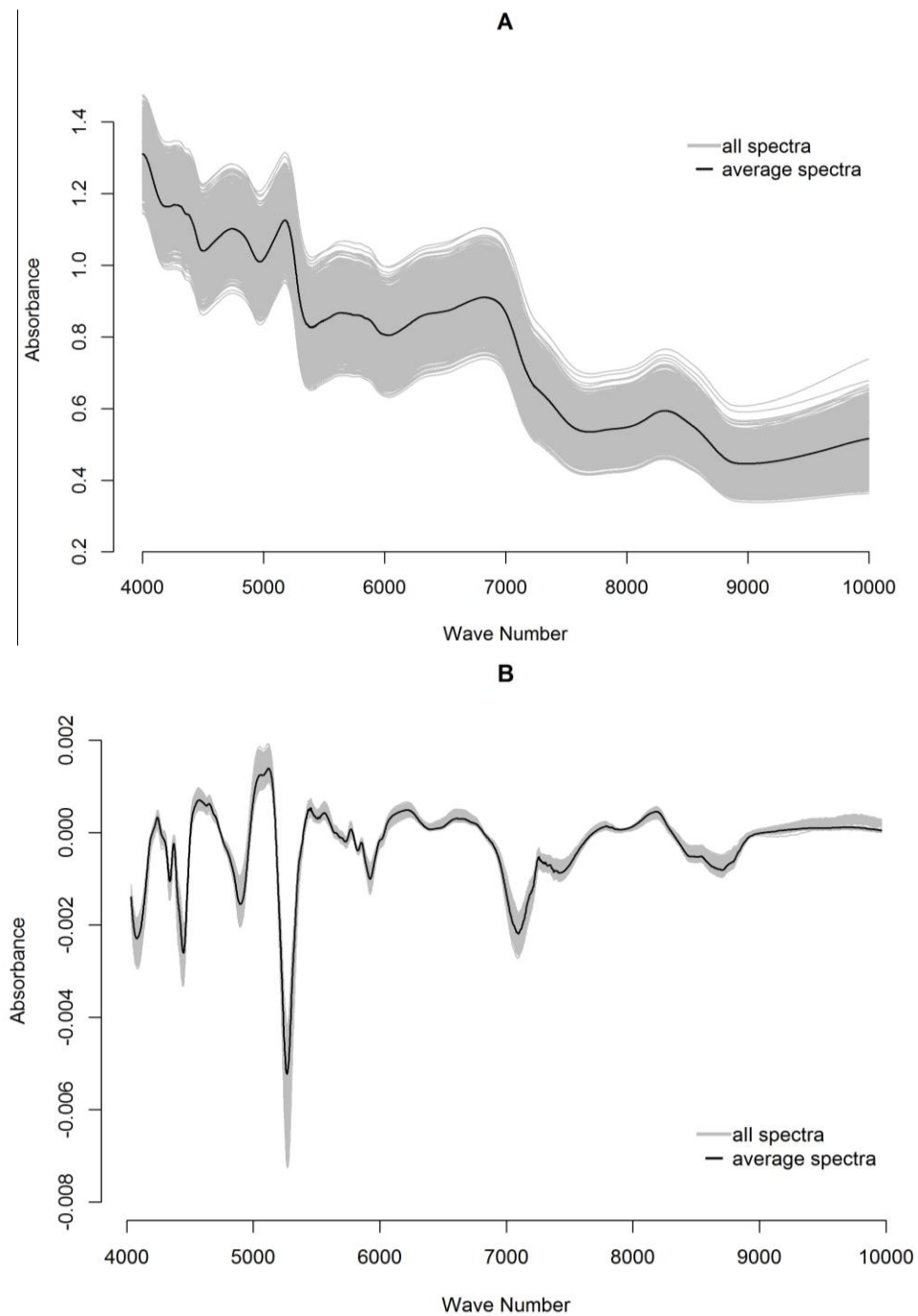
Figure 1. All analyzed spectra from the set of whole kernel scans are shown in grey for A) untreated scans, or raw spectra and the B) the Savitsky-Golay first derivative of the scans. The black line in each graph is the average of all scans.

Statistical Models to Predict Grain Yield using NIRS information

Multiple statistical approaches were investigated for grain yield prediction accuracy. Both phenomic prediction models, where data was assessed on a plot basis, and phenomic selection models, where data is assessed on an entry basis, are described. The models using the NIRS and phenotypic information are described below.

Partial least squares regression (PLSR)

Partial least squares regression (PLSR) was used to predict yield as implemented in the *pls* package in R (Wehrens and Mevik, 2007). Models were cross-validated using the default of 10 segments, and then further validated on a held out test set. When training this phenomic prediction model, random cross validation holds out observations on a plot basis, meaning hybrids can remain in both the training and testing sets. To mitigate this, a held out test set was selected to ensure that entire instances of a hybrid were held out. These test sets were selected using the DUPLEX method available in the *prospectr* package (Stevens and Ramirez-Lopez, 2014), which functions by selecting the two most distant points (based on principal components) iteratively for the sample size desired. This serves to select a spectrally representative set of samples to use for validation or calibration. If a hybrid was selected by the function to be held out in the test set, all replicates of that hybrid were held out for training the model; this is similar to predicting untested genotypes in known environments (Saint Pierre et al., 2016).

For example, for the model trained on both years (PLSR 1, Table 1 in Results), the training set consisted of 1573 samples, representing 253 pedigrees. The validation test set was

18

composed of 848 samples, representing 91 pedigrees. All PLSR models were validated in this way, though the number of samples used to train and test the models varied (see Tables 1 and 2 in Results).

Additionally, PLSR models were built to investigate the ability for the model to be trained on one year to predict the same year, or to predict the other year. Models were evaluated either for their ability to predict known or unknown hybrids in an unknown year. Evaluating the ability of the model to predict yield for a separate experiment from another year, consisting of different hybrids, or predicting unknown genotypes in unknown environments (Saint Pierre et al., 2016), is also of great practical interest in breeding programs.

The number of PLSR components used in each model were selected based on the root mean squared error prediction (RMSEP) curve using the one-sigma heuristic and permutation approaches as implemented in the *pls* package in R (Wehrens and Mevik, 2007). R code used for PLSR models can be found in Appendix B.

Baseline single environment and across environment models for implementing NIRS BLUP and Functional Regression

In this subsection, we explain the six phenomic selection models implemented using the phenotypic and NIRS information under a NIRS BLUP and functional regression approach. Before describing the six models used under BLUP and functional regression we briefly describe the model used for obtaining the BLUEs (best linear unbiased estimators) for grain yield as well as for NIRS over each of the wavenumbers with the intention of removing the design. This analysis was the first stage analyses that produced the BLUEs used in the

second stage analysis where models M1 to M6 are implemented. BLUEs of each hybrid and

NIRS in each environment were obtained using the following model

$$y_{jk} = \mu_j + b_k + H_j + \epsilon_{jk} \qquad \text{(Eq. 1)}$$

where y_{jk} is the response variable (grain yield, or wavenumber measurements) of the jth

hybrid in the kth complete block; μ is the overall mean; b_k is the random effect of the kth

complete block assuming independent and identically distributed (iid) $N(0, \sigma_b^2)$; H_j is the

fixed effect of the jth hybrid, and ϵ_{jk}, assuming iid $N(0, \sigma_e^2)$, represents the random residual

plot error associated with the observation y_{jk}.

Then in the second stage analysis these BLUEs of grain yield were used as response

variable and the BLUEs of each NIRS were incorporated as covariates to implement

improved version of the following model that takes into account the hybrid $\times$ environment

interaction

$$y_{ij} = \mu + E_i + H_j + HE_{ij} + \epsilon_{ij} \qquad \text{(Eq. 2)}$$

where E_i is the fixed main effect of the ith environment; H_j is main random effect of the jth

hybrid identical and independent normal distributed (iid) $N(0, \sigma_L^2)$; HE_{ij} is the random

interaction effect between the ith environment and the jth hybrid; it is assumed iid $N(0, \sigma_{HE}^2)$;

and ϵ_{ij} is the residual error term assumed iid $N(0, \sigma^2)$. For the sake of clarity, later we named

the model including hybrid $\times$ environment interactions as model M5.

The NIRS BLUP model with interaction – M1

Wavenumber can be introduced in the previous baseline model such that the effect of hybrid (H_j) can be replaced by w_j, which is expressed as a linear regression of phenotypic data on the wavenumber covariates that approximates the value of the jth hybrid such that the vector of NIRS random effect $\boldsymbol{w} = [w_1, \ldots, w_J]^T$ is assumed $\boldsymbol{w} \sim N(\boldsymbol{0}, \mathbf{W}\sigma_w^2)$, where σ_w^2 is the spectral variance component, and $\mathbf{W}$ is a scaled wavenumber relationship matrix that was computed using the matrix of wavenumber information ($\boldsymbol{N}$) from data of order $J \times w$ (for a total of J lines and w wavenumbers) as $\boldsymbol{W} = \frac{NN^T}{w}$.

It is important to point out that the final $\boldsymbol{W}$ reported for lines was calculated as the average of the wavenumber relationship matrices calculated in each environment since the wavenumber reflectance values were obtained for each hybrid in each environment. Similarly, for the interaction terms, when wavenumber information was used, the hybrid $\times$ environment interaction HE_{ij} of the lines across environments baseline model was replaced by wE_{ij} the random effect of the interaction term of the ith environment and the jth wavenumber and $\boldsymbol{wE} = [wE_{11}, \ldots, wE_{IJ}]^T \sim N(\boldsymbol{0}, \mathbf{W}_A \sigma_{wE}^2)$, where σ_{wE}^2 is the variance component associated with the wavenumber $\times$ environment interaction, but $\mathbf{W}_A$ was calculated as $\mathbf{W}_A = \frac{N_A N_A^T}{w}$ where $\boldsymbol{N_A}$ is the matrix of wavenumber information of order $JI \times w$ where I denotes the number of environments assuming that the same number of hybrids, J, were evaluated in each environment.

Therefore, using basic wavenumber information, the NIRS BLUP model including environments, wavenumber, and their interactions becomes model M1, which is

$$y_{ij} = E_i + w_j + wE_{ij} + \epsilon_{ij}. \qquad\qquad \text{(M1, eq. 3)}$$

M1 establishes relationships between hybrids based on wavenumber information, which is similar to the conventional G-BLUP. In a traditional G-BLUP approach, dense molecular markers are used (instead of wavenumber), and covariates are incorporated to develop the genomic relationship matrix (G) establishing the similarities between hybrids based on genomic markers. The main difference in this scenario is that that wavenumbers are measured for all hybrids in each environment, as opposed to DNA markers, which are measured once for each hybrid.

Functional regression model with interaction – M2

The previously defined baseline model, $y_{ij} = E_i + H_j + HE_{ij} + \epsilon_{ij}$, can be combined with the information from the wavenumber, but not as a wavenumber matrix as introduced in model M1. Instead, wavenumber values are combined using models displaying wavenumbers as functional covariates. These models are called functional regression models, where the response variable is a scalar and some of the covariates are functions. Thus, this combination will be named model M2 and is represented as

$$y_{ij} = E_i + H_j + HE_{ij} + \int_{4000\ cm-1}^{10000\ cm-1} w_{ij}(k)\beta_1(k)dk + \epsilon_{ij}. \qquad \text{(M2, eq. 4)}$$

Here, $w_{ij}(k)$ is the functional predictor and represents the value of a continuous underlying process evaluated at wavenumber k; $\beta_1(k)$ is the functional regression beta coefficient for the functional part of the model; and ϵ_{ij} is defined as in model M1 (Montesinos-López et. al., 2017b).

The functional regression models add the 3,076 wavenumbers to M2 as functional

covariates constructed over the interval between 4,000 and 10,000 cm^{-1}. Model M2 was

implemented using a total number of $L = 101$, Fourier basis, to approximate the functions,

$w_{ij}(k)$ and $\beta_1(k)$ in model M2. The Fourier basis is a set of predefined functions in terms of

sine and cosine. If in model M2, we replace the integral of the product by the sum of products

($\sum_{k=1}^{p} w_{ijk}\beta_k$), then model M2 is reduced to a conventional linear regression model. In this

model, w_{ijk} represents the kth wavenumber data measured on the jth hybrid in the ith

environment with $k = 1,2,\dots,3076$; w_{ijk} are BLUEs obtained in the first stage analysis; and

β_k is the beta regression coefficient for the kth wavenumber. Note that we implemented model

M2 to avoid implementing a model with 3076 beta coefficients corresponding to each of the

wavenumbers. This model was implemented using a Bayesian approach.

The NIRS BLUP model with only main effects of hybrids – M3

Model M3 is similar to M1 but only with main effects of hybrids (with wavenumbers

incorporated through a covariance matrix) applied for each environment.   This model is equal

to

$$y_j = w_j + \epsilon_j. \tag{M3, eq. 5}$$

In M2 eq. 4, $\boldsymbol{w} \sim N(\boldsymbol{0}, \boldsymbol{W}\sigma_w^2)$, where $\boldsymbol{W}$ was calculated with the wavenumber information of

each environment. Note that model M3 is obtained from model M1, ignoring the main effect

of environment, E_i, and the interaction term, wE_{ij}. It is reduced to $y_j = w_j + \epsilon_j$, with $\boldsymbol{w} = [w_1, \dots, w_J]^T$ and $\boldsymbol{w} \sim N(\boldsymbol{0}, \boldsymbol{W}\sigma_w^2)$.

Functional regression model with only main effects of hybrids – M4

Model M4 is similar to M2, but with only the main effects of hybrids, $y_j = H_j + \epsilon_j$. When using the wavenumbers as covariates, model M4 is described as

$$y_j = H_j + \int_{4000\ cm-1}^{10000\ cm-1} w_j(k)\beta_1(k)dk + \epsilon_{ij}. \qquad \text{(M4, eq. 6)}$$

Models without NIRS – M5 and M6

As previously mentioned, the model of the hybrid $\times$ environment interactions is named model M5,

$$y_{ij} = \mu + E_i + H_j + HE_{ij} + \epsilon_{ij} \qquad \text{(M5, eq. 7)}.$$

Furthermore, when model M5 includes only the random effect of hybrids, this yields model M6;

$$y_j = H_j + \epsilon_{ij} \qquad \text{(M6, eq. 8)}.$$

Assessing the models' prediction accuracy

For all validation schemes, we report the Pearson´s correlation (r) and RMSEP between the observed and predicted values of grain yield for the testing set. For PLSR models, these results are reported for the held out validation test sets as described above. For

functional regression and NIRS BLUP models, we report the average of the five folds resulting from the random partitions implemented.

The phenomic selection models (functional regression and NIRS BLUP) were subjected to three different cross-validations schemes; a) cross-validation predicting one environment using the other environments as the training set; b) random cross-validation within each environment where 20% of the maize hybrids are predicted using 80% of the other hybrids as the training population; c) random cross-validation for multi-environments where 20% of the lines were observed in some environments and predicted in other environments using the other 80% of lines.

The first scheme consists of the prediction of all hybrids in one environment (in turn, WW_2011, WW_2012, WS_2011, and WS_2012) using all the hybrids combined from the other three environments as the training set. This prediction is named cross-validation 0 (CV0) (predicting known genotypes in pseudo-unknown environments). In this experiment, even if the model was not trained on the environment, since the NIRS data is collected in the environment, the environment is therefore only considered to be pseudo-unknown. The PLSR, functional regression, and NIRS BLUP models were all assessed with the CV0 scheme, to test the models' abilities to perform in unknown environments.

The second cross-validation scheme is a random cross-validation within each environment, where a 5 fold random partitioning was implemented. In each partition, 20% of the hybrids are predicted by the remaining 80% of hybrids; this is named (CV1) (unknown genotypes in known environments).

25

Finally, the last cross-validation strategy was a 5 fold random cross-validation (CV2) where 20% of the hybrids were predicted by 80% of the other hybrids, similar to CV1. However, in this case, 20% of hybrids are not observed in some environments, and thus predicted in those environments, but are observed in other environments (known genotypes in known environments). This CV2 mimics a prediction problem faced by breeders in incomplete field trials where hybrids are evaluated in some, but not all, target environments (usually called *sparse testing*). Note that in CV2, all environments were used in training the model. In this cross validation, some hybrids can never be part of the training set. This is because some hybrids were not observed in all environments. This CV2 strategy used sampling with replacement, which means that one observation can appear in more than one partition. Training and testing partitions were obtained as follows: since the total number of records per trait available for the data set with multi-environments is $N = J \times I$ observations comprising of J hybrids and I environments, to select lines in the test data set, we fixed the percentage of data to be used for test (Testing=20%). Then we chose $0.20 \times N$ (hybrids) at random, and subsequently, one environment per hybrid was randomly picked from I environments. The resulting cells (ij) were assigned to the test data set, while cells not selected through this algorithm were allocated to the training data set. Hybrids were sampled without replacement if $J \geq 0.20 \times N$, and with replacement otherwise (López-Cruz et al., 2015).

The PLSR models were tested to predict unknown genotypes via their ability to predict a held out validation set of genotypes the model was not trained on, as described

above in the PLSR section. To test their ability to predict in new environments, on either

known or unknown hybrids, held out test sets were designed (see Table 2 in results). PLSR

models were also subjected to the CV0 scheme.

Other analysis

To investigate if yield was simply correlating with some compositional aspect of the

kernels, predictions for crude protein, starch, and oil content were obtained using existing

calibrations from these components built within the research program (Meng et al., 2015;

Christman, 2017). To assess the correlation of composition to yield, simple linear models

(LM) were built using the lm function in R (R Core Team, 2018). These LM were compared

to the results of the PLSR model trained and tested on the same samples sets, as both predict

on a plot basis. Repeatability for grain yield on an entry basis across all environments were

estimated as explained by Anderson et al. (2018).

Results

The results are presented in two main sections. The first section presents the

predictions under the PLSR and LM methods. The second section describes the predictions

based on the phenomic selection models (NIRS BLUP and functional regression), for single

environment and G×E multi-environment approaches under the CV0, CV1, and CV2

schemes.

Prediction accuracy of PLSR and LM

Yield predictions from models using PLSR within the same years demonstrated a high

Pearson's correlation to maize grain yield on a plot basis (r = 0.84; Table 1, PLSR 1). The

model successfully used spectra alone to predict the yield of a sample on a plot basis, based

on error in prediction. With only spectral data, the PLSR model using 15 PLS components

predicted yield with a RMSEP of 163.67 g m^{-2}.

To investigate if kernel compositional traits were correlated with yield, NIRS

composition predictions were used to predict yield using simple LM. A PLSR model as above

was run, but using only the 2,155 samples with composition predictions to build the training

and testing sets (Table 1; PLSR 2); this had an r of 0.82. Looking at individual components,

crude protein was the only measured component with a strong correlation with yield (r = 0.58)

(Table 1; LM 1). Both starch and fat had very low correlations (r ≤ 0.17, Table 1; LM 2-3).

Combining all compositional predictions into a model to predict yield improved predictions

over protein alone, but only slightly, yielding an r of 0.64 (Table 1; LM 4).

Table 1. Description of factors used in training PLSR and LM models for grain yield on all GWAS data and their results. The number of samples used in the training and testing sets as well as the number of genotypes (pedigrees) represented by those samples are reported. The number of Partial Least Squares components used for each model are listed (PLS) when relevant. Pearson´s correlation (r) and root mean square error of prediction (RMSEP) in g m^{-2} were calculated between observed and predicted values of the test set.

| Scenario | Training Data | | Testing Data | | Predictor | PLS | r | RMSEP |
	Samples	Pedigrees	Samples	Pedigrees				
PLSR 1	1573	253	848	91	NIRS	15	0.84	163.67
PLSR 2[†]	1389	254	734	90	NIRS	11	0.82	174.52
LM 1	1389	254	734	90	Protein[‡]	-	0.59	247.25
LM 2	1389	254	734	90	Starch[‡]	-	0.21	296.38
LM 3	1389	254	734	90	Oil[‡]	-	0.12	301.86
LM 4	1389	254	734	90	Protein/starch/oil[‡]	-	0.63	238.52

† The full data set did not have corresponding composition estimations for each sample, so a baseline PLSR was run with the subset of the data for which composition was available.
‡ Protein, starch, and oil values were estimations based on previously trained NIRS models calibrated with wet chemistry.

Next, PLSR scenarios were developed to predict yield on samples from a year unknown to the model, on known or unknown hybrids (see Table 2). Results showed that a model built on 2011 predicted unknown hybrids in 2011 better than a 2012 model predicting unknown hybrids in 2012 (Table 2; PLSR 3-4). Results also showed that a model trained on 2011 predicted 2012 well, and better than the reverse (PLSR 5 and 8); even though the predicted hybrids were known to the model in both cases. In predictions between years, the best performing model was trained on all available samples from 2011 to predict 2012 (PLSR 6). PLSR 6 had more training data than PLSR 5, as this training data represented all available 2011 data (including the USDA and other breeding populations from 2011). We also investigated the ability of a model to predict only unknown hybrids from a new year (PLSR 7), which was comparable to when most of the hybrids were already known from a previous year (PLSR 5).

Table 2. Description of factors used in training PLSR scenarios to carry out year-to-year predictions on grain yield and their results. The number of samples used in the training and testing sets as well as the number of genotypes (pedigrees) represented by those samples are reported. The number of partial least squares components used for each model are listed (PLS). Pearson´s correlation (r) and root mean square error of prediction (RMSEP) in g m^{-2} were calculated between observed and predicted values of the testing sets.

Scenario	Training Data			Testing Data			PLS	r	RMSEP
	Year	Samples	Pedigrees	Year	Samples	Pedigrees			
PLSR 3	2011	913	213	2011	363	61	12	0.81	121.03
PLSR 4	2012	768	239	2012	448	85	15	0.72	195.03
PLSR 5	2011	1276	275[†]	2012	1216	324[†]	9	0.62	247.58
PLSR 6	2011[‡]	1962	475[†]	2012	1216	324[†]	15	0.69	209.64
PLSR 7	2011	1276	275	2012	201	69	10	0.60	244.38
PLSR 8	2012	1216	324[†]	2011	1276	275[†]	11	0.19	329.79

† Indicates the pedigrees were not completely unique between years.
‡ Indicates data from separate 2011 experiments were included.

Cross-validation under CV0, CV1, and CV2 schemes

Prediction of observed hybrids in new environments (CV0)

Results of cross-validation CV0, where one environment is predicted using the other three environments (year by irrigation) as the training data, for PLSR, NIRS BLUP (M1), and functional regression (M2) models are shown in Tables 3 and 4. PSLR had slightly higher prediction accuracies across all scenarios than both M1 and M2 (r = 0.55; Table 3). Model M2, using functional regression with wavelength as covariates, gave higher prediction on average when compared to model M1, using NIRS BLUP (r = 0.53 versus 0.40; Table 4). Error, as presented via the average RMSEP in g m^{-2} were nearly identical between functional regression and PLSR models (272.12 and 272.66, respectively). In general, all hybrids predicted in individual environments had higher prediction accuracies for model M2 than for model M1 (e.g. WS_2012 = 0.68 for M1 versus WS_2012 = 0.74 for M2). Results showed that the prediction of hybrids unobserved in WW using observed hybrids in WS is better than the reverse, and this is much higher for M2 than for M1 (WS = 0.42 and WW = 0.75 for M2 versus WS = 0.20 and WW = 0.28 for M1; Table 4). PLSR performed similarly to M2 overall (Table 3). Similarly, hybrids grown in 2012 were more predictable than hybrids in 2011, which makes sense given 2011 was a drought year, presenting more overall stress to plants. Based on the r values between samples' observed and predicted values, and the RMSEP results, functional regression model M2 and PLSR models had substantially higher prediction accuracies of unobserved hybrid performance than NIRS BLUPs model M1.

Table 3. Pearson´s correlation (r) between predicted and observed values for environments water stress (WS) and well-watered (WW) during years 2011 and 2012 for PLSR phenomic prediction models. The number of PLS components are also reported (PLS). Root mean squared error prediction (RMSEP) is reported in g m^{-2}. Cross-validation CV0 predicted one environment using the other three environments, one irrigation regime (WW, WS) using the other irrigation regime, or one year (2011, 2012) using the other year.

Model	Predicted Environment	PLS	r	RMSEP	Validation
PLSR	WS_2011	12	0.43	402.97	CV0
PLSR	WS_2012	12	0.64	206.12	CV0
PLSR	WW_2011	13	0.72	232.09	CV0
PLSR	WW_2012	13	0.47	235.22	CV0
PLSR	WS	12	0.48	343.08	CV0
PLSR	WW	10	0.81[†]	183.49	CV0
PLSR	2011	11	0.19	329.79	CV0
PLSR	2012	12	0.66	248.89	CV0
	Mean	--	0.55	272.66	

† Model with highest correlation (r)
‡ Model with lowest RMSEP

Table 4. Pearson´s correlation (r) between predictive and observed value for environments water stress (WS) and well-watered (WW) during years 2011 and 2012 for phenomic selection models M1 and M2 including interactions between wavenumber × environment. Root-mean squared error prediction (RMSEP) is reported in g m^{-2}. Cross-validation CV0, predicted one environment using the other three environments, one irrigation regime (WW, WS) using the other irrigation regime, or one year (2011, 2012) using the other year.

Model	Predicted Environment	r	RMSEP	Validation
M1	WS_2011	0.41	1905.22	CV0
M1	WS_2012	0.68[†]	1533.79	CV0
M1	WW_2011	0.62	101.55[‡]	CV0
M1	WW_2012	0.45	1581.57	CV0
M1	WS	0.19	762.47	CV0
M1	WW	0.28	3969.53	CV0
M1	2011	0.23	638.25	CV0
M1	2012	0.31	340.07	CV0
	Mean	0.40	1354.06	
M2	WS_2011	0.45	322.53	CV0
M2	WS_2012	0.74	148.28[‡]	CV0
M2	WW_2011	0.67	215.82	CV0
M2	WW_2012	0.44	220.89	CV0
M2	WS	0.42	395.73	CV0
M2	WW	0.75[†]	197.21	CV0
M2	2011	0.13	354.82	CV0
M2	2012	0.64	321.69	CV0
	Mean	0.53	272.12	

[†] Model with highest correlation (r)
[‡] Model with lowest RMSEP

Prediction of unobserved hybrids within environment (CV1)

Results of models M3 and M4 under CV1 for the predictions of hybrids within each environment are shown in Table 5. Models M3 and M4 are the counterpart of models M1 and M2, respectively, but include only main effects of hybrids. In general, results showed that NIRS BLUP model M3 with only main effects predicts the unobserved hybrids better (0.57) than the main effect functional regression model M4 (0.46). The prediction accuracy of unobserved hybrids in WS_2012 for M3 (0.54) was lower than that achieved in WS_2012 for M4 (0.64).

Table 5. Pearson´s correlation (r) between predictive and observed values for environments water stress (WS) and well-watered (WW) during years 2011 and 2012 for models M3 and M4 within only a single environment. Root-mean squared error prediction (RMSEP) is reported in g m^{-2}. CV1 predicted 20% of unobserved lines in that single environment using the remaining 80% of lines. SE_r denotes standard error under r and SE_RMSEP denotes standard error under RMSEP.

Model	Environment	r	SE_r	RMSEP	SE_RMSEP	Validation
M3	WS_2011	0.28	0.07	75.97[‡]	18.6	CV1
M3	WS_2012	0.54	0.05	148.45	51.12	CV1
M3	WW_2011	0.63	0.02	101.04	33.14	CV1
M3	WW_2012	0.30	0.06	184.56	58.31	CV1
M3	WS	0.85[†]	0.02	132.64	40.1	CV1
M3	WW	0.77	0.03	170.97	57.57	CV1
M3	2011	0.75	0.02	108.36	30.03	CV1
M3	2012	0.37	0.02	203.56	40.28	CV1
	Mean	0.57	---	1780.3	---	
M4	WS_2011	0.30	0.11	74.93[‡]	1.73	CV1
M4	WS_2012	0.64[†]	0.15	135.06	2.44	CV1
M4	WW_2011	0.63	0.15	103.72	2.23	CV1
M4	WW_2012	0.25	0.20	189.47	1.57	CV1
M4	WS	0.47	0.21	109.22	3.1	CV1
M4	WW	0.44	0.26	152.74	3.04	CV1
M4	2011	0.47	0.21	90.47	2.99	CV1
M4	2012	0.45	0.26	164.53	3.14	CV1
	Mean	0.46	---	132.77	---	

† Model with highest correlation (r)
‡ Model with lowest RMSEP

Within an irrigation treatment, prediction of unobserved hybrids tested under WS using observed hybrids evaluated under WW was higher than the prediction of hybrids under WW using hybrids in WS by means of models M3 and M4, if both years were included (Table 5). The prediction of hybrids in WS environment was of 0.85 with model M3 and of only 0.47 for model M4. In this case, unobserved hybrids in 2011 were more predictable than unobserved hybrids in 2012. Based on r between observed and predicted values, results indicated that functional regression model, M4, had lower prediction accuracy than NIRS BLUP model M3. However, based on RMSEP values, results indicated that functional regression model M4 had higher prediction accuracy than NIRS BLUP model M3.

Prediction of unobserved hybrids in some environments (sparse testing) (CV2)

Table 6 shows the random cross-validation CV2 fitting models M1 and M2. When including the environment × hybrid interaction the functional regression using the NIRS as covariates (M4) gave much higher prediction accuracy of the unobserved hybrids (r = 0.73; Table 6) than the NIRS BLUP (r = 0.38). Model M2 with functional regression covariates in the G×E multi-environment modes gave very good predictions of unobserved hybrids in WS (r = 0.91; Table 6) as well as in WW (r = 0.87); on the contrary, NIRS BLUP with G×E multi-environment (M1) gave very poor prediction of unobserved hybrids in WS (r = 0.07) and in WW (r = 0.14) (Table 6).

Table 6. Pearson´s correlation (r) between predictive and observed values for environments water stress (WS) and well-watered (WW) during years 2011 and 2012 for models M1 and M2 including interaction wavenumber × environment. Root-mean squared error prediction (RMSEP) is reported in g m^{-2}. Random cross-validation CV2 for predicting 20% of the lines not observed in listed environments but observed in others. SE_r denotes standard error under r and SE_RMSEP denotes standard error under RMSEP.

Model	Environment	r	SE_r	RMSEP	SE_RMSEP	Validation
M1	WS_2011	0.50	0.05	78.06‡	27.81	CV2
M1	WS_2012	0.65	0.05	130.58	31.55	CV2
M1	WW_2011	0.70†	0.02	92.4	29.66	CV2
M1	WW_2012	0.47	0.13	170.61	58.07	CV2
M1	WS	0.07	0.09	254.6	45.87	CV2
M1	WW	0.14	0.03	266.92	56.57	CV2
M1	2011	0.12	0.05	166.05	48.31	CV2
M1	2012	0.37	0.07	206.34	49.67	CV2
	Mean	0.38	---	182.69	---	
M2	WS_2011	0.48	0.15	80.1‡	2.51	CV2
M2	WS_2012	0.74	0.10	120.52	2.08	CV2
M2	WW_2011	0.71	0.10	88.08	2.01	CV2
M2	WW_2012	0.52	0.10	167.28	103.16	CV2
M2	WS	0.91†	0.01	103.11	41.61	CV2
M2	WW	0.87	0.03	134.85	69.51	CV2
M2	2011	0.86	0.02	84.65	26.16	CV2
M2	2012	0.75	0.07	147.87	73.97	CV2
	Mean	0.73	---	119.66	---	

† Model with highest correlation (r)
‡ Model with lowest RMSEP

The unobserved hybrids of WS environments in both years were less predictable than the unobserved hybrids in WW environments, and hybrids in 2012 were more predictable than those tested in 2011 for model M1, whereas the opposite was true for model M2 (Table 6). Based on r values and RMSEP, results indicated that functional regression model M2 had substantially higher prediction accuracy than NIRS BLUP model M1. Models M5 and M6, which do not include the NIRS data (Table 7), gave a much lower prediction accuracy than models M1, M2, M3, and M4 that include NIRS, for all three cross-validations schemes (CV0, CV1, and CV2).

Table 7. Pearson´s correlation (r) between predictive and observed values for environments water stress (WS) and well-watered (WW) from 2011 and 2012 for model M4, as well as for models M5 and M6, which do not include NIRS information. Root-mean squared error prediction (RMSEP) is reported in g m^{-2}. SE_r denotes standard error under r and SE_RMSEP denotes standard error under RMSEP. For M4 and M6 models the main effects of environments were considered.

Model	Environment	r	SE_r	RMSEP	SE_RMSEP	Validation
M4	WS_2011	0.32	---	492.54	---	CV0
M4	WS_2012	0.70	---	427.09	---	CV0
M4	WW_2011	0.48	---	200.7[‡]	---	CV0
M4	WW_2012	0.59[†]	---	3135.51	---	CV0
M4	WS	0.28	---	4095.16	---	CV0
M4	WW	0.33	---	482.74	---	CV0
M4	2011	0.26	---	1182.65	---	CV0
M4	2012	0.39	---	941.75	---	CV0
	Mean	0.42	---	1369.77	---	
M5	WS_2011	0.01	0.06	78.45[‡]	17.12	CV1
M5	WS_2012	0.04	0.09	175.46	62.18	CV1
M5	WW_2011	0.04	0.05	128.83	45.48	CV1
M5	WW_2012	0.06[†]	0.03	194.6	60.29	CV1
M5	WS	-0.10	0.04	251.91	54.39	CV1
M5	WW	-0.09	0.05	268.33	49.75	CV1
M5	2011	-0.09	0.05	161.69	49.89	CV1
M5	2012	-0.10	0.03	220.83	37.88	CV1
	Mean	-0.03	---	185.01	---	
M6	WS_2011	0.34	0.05	92.28[‡]	33.68	CV2
M6	WS_2012	0.66[†]	0.03	142.67	58.21	CV2
M6	WW_2011	0.46	0.08	114.52	39.88	CV2
M6	WW_2012	0.52	0.07	169.42	58.32	CV2
M6	WS	-0.30	0.02	263.42	48.64	CV2
M6	WW	-0.27	0.03	281.41	55.38	CV2
M6	2011	-0.15	0.03	176.22	37.02	CV2
M6	2012	0.20	0.02	216.95	61.52	CV2
	Mean	0.18	---	182.11	---	

† Model with highest correlation (r)

‡ Model with lowest RMSEP

Predicting on elite breeding material

The PLSR 1 model (Table 1 describes training set) was applied to other breeding tests from 2011 to investigate the ability for this calibration to predict other genetic material, relevant for breeding. This model had an ability to predict not only unknown pedigrees from within the experiment (a genetics study where all hybrids share the same tester line Tx714), but also showed ability in predicting genetically distinct material from different experiments grown in 2011 on a plot basis (Fig. 2). These breeding experiments were grown across three locations in the state of Texas. Among the best predicted were 3AF (r = 0.68), a low yielding, late planted diverse test grown in Corpus Christi, SERAT (r = 0.53, Fig. 2B) a moderate yielding late planted test in College Station, while the other breeding tests from Weslaco (1AF and 2AF) and College Station (4AF) were not particularly well correlated (r = 0.23, 0.25, 0.21, respectively), as well as GEM (Fig. 2B; r = 0.33). In the GEM predictions, two plots were predicted erroneously, having a negative yield prediction. These plots were removed from plotting and analysis. It is important to note that while these plots were not accurately predicted, they were also some of the lowest yielding plots from the test.

Various phenomic selection models were also investigated for their ability to perform on this unrelated material, though results were not as impressive, even when 10% of the new material was included into the training set.
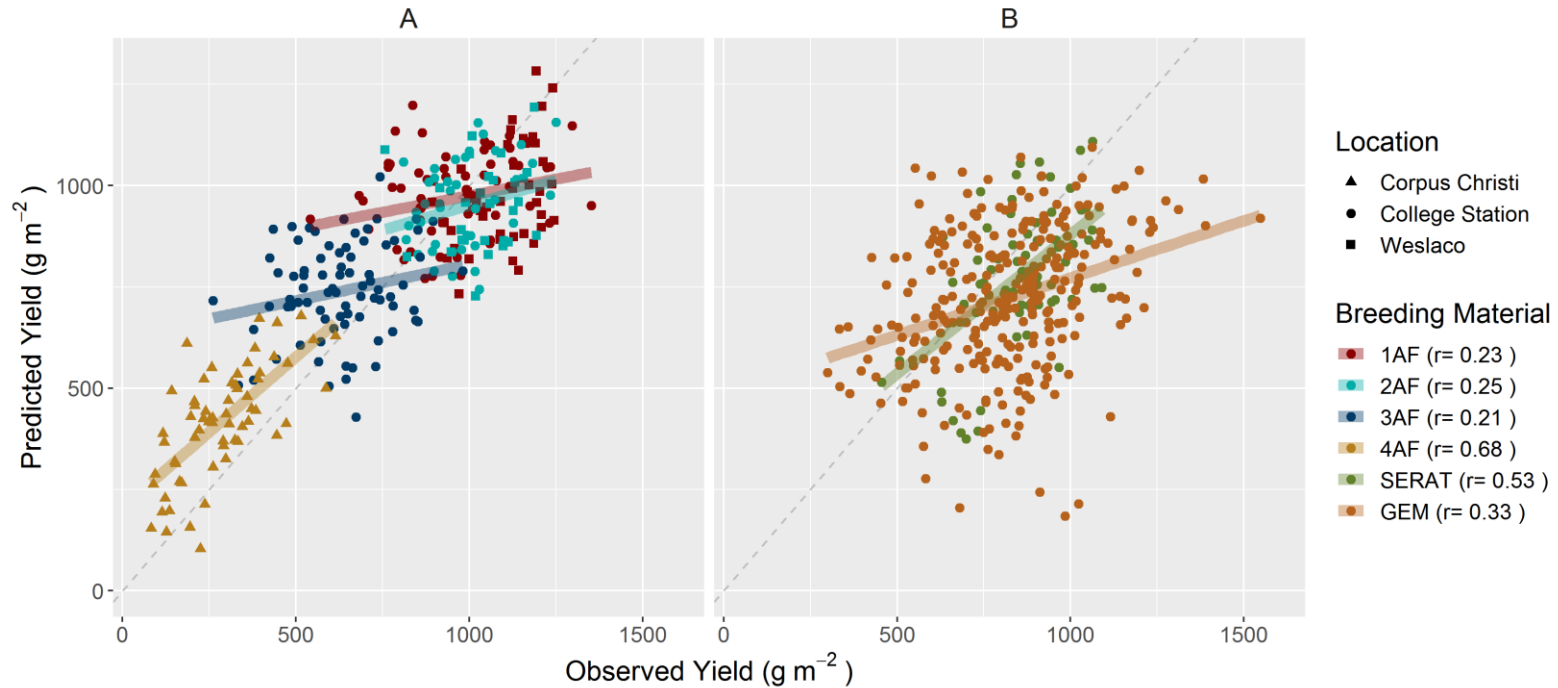
Figure 2. Ability for PLSR1 (Table 1) model trained on original samples to predict yield on a plot basis of A) genetically unrelated samples of breeding trials from the Texas A&M Corn Program in 2011 B) genetically unrelated tests representing two tests of material from outside of the program (note: two negatively predicted plots were removed from GEM test). The dotted lines show the 1:1 line.

The repeatability of yield across and within environments are presented in Table 8. Note that the lowest repeatability is within WS_2011 (0.55), and that overall repeatability was relatively high (0.77).

Table 8. Repeatability for yield on an entry basis across all environments and within each environment

Environment	Repeatability
All	**0.7745**
WS_2011	0.5559
WW_2011	0.6369
WS_2012	0.7421
WW_2012	0.7534

Discussion

Our aims were to determine if information from NIRS of maize grain could be used to estimate grain yield in both phenomic prediction and selection scenarios. If the spectra can capture value of genetic material, there seems to be a great potential for these methods to be useful within the context of breeding programs. The models used in this experiment represented two methods of mining the spectral data, phenomic prediction and phenomic selection. The relative success of both approaches offers promise for future techniques using NIRS beyond its traditional composition analysis.

It was clear that the NIRS information was the basis of most prediction capability for each of the model types used in this study. Models not including NIRS information (Table 7) had very little correlation between predicted and observed values. The mode of action for the strong predictive relationships across genotypes and years is not yet understood.

43

One hypothesis is that spectra could be indicative of some compositional aspect of the kernels that is correlated with yield. This could be investigated by comparing composition prediction model results, and only crude protein appeared to have any significant correlation on its own (Table 1; LM 1). Combining all measured compositional predictions (starch, oil, protein; LM 4) into a model to predict yield improved predictions over a single compositional component alone, demonstrating that composition is relevant but likely not sufficient in predicting yield. However, adding more components could further improve correlations. While crude protein seemed to have a significant inverse correlation with yield, using composition in any combination was never as accurate as using NIRS data.

Alternatively, the spectra could be explaining genetic relatedness between samples. If the NIRS was capturing this alone, the phenomic selection models, would have significantly outperformed the PLSR phenomic prediction models, and this was not the case either. However, given that the NIRS BLUP and functional regression models could not predict the unrelated genetic material as well as the PLSR models suggests that they likely rely more on genetics than a plot-based method. The plot-based nature of the PLRS phenomic prediction method may mean it relies more on composition, which could be why it is more flexible and adaptable to predict new genetic material (Fig. 2).

Phenomic prediction with PLSR analysis

In the case of the PLSR models, which predicted on each sample on a plot basis, genetic relatedness did not appear to have any discernible correlation to prediction accuracy (Table 2; Fig. 2), since many genetically unrelated samples were predicted quite well (Fig 2). Because composition is likely a factor in the PLSR model's prediction capacity, this may

44

expand the prediction capacity to a wider range of germplasm, since composition may be a more robust measure than kinship, at least on unrelated material.

The GEM dataset from 2011 was predicted with poor accuracy (r = 0.33), though the SERAT test was predicted relatively well (r =0.53; Fig. 2B). Both of these tests represented material from outside of the A&M Corn Breeding Program, and therefore represent different genetics than the AF tests from within the program. Most likely, some combination of composition and relatedness is what was being captured in the NIRS spectra. The fact that using this plot basis method to predict new, unrelated material (Fig. 2) was much better than the entry basis methods suggests that perhaps composition becomes more important for the prediction capacity in these cases.

The composition data used in the LM came from the NIRS in some capacity, so it may be that the compositional predictions capture some of the spectral data important in predicting yield. It is more likely that the spectra in its entirety is able to pick up the composition of the kernel as a whole, including protein, in relation to how it compares to other samples. In addition, yield predictions were surprisingly better than what is usually observed or expected in predictions of compositional traits (Christman, 2017).

By predicting or at least improving yield estimates, grain NIRS has the potential to reduce the number of field replications needed. Reducing replications or using less expensive plot evaluation methodology has potential to save breeding resources. Improving phenotypic data accuracy also has potential to improve selection ability. For new technologies to improve plant breeding they need to either reduce evaluation resources, improve data accuracy, or both. High-throughput phenotyping tools like NIRS offer potential for inexpensive, fixed-cost

methods of evaluating best yielding varieties within a breeding program. It is also important to note that the repeatability for yield in this study was relatively high, which may have impacted the success of NIRS being able to capture yield. One surprising factor here, however, is that the most stressed environment (WS_2011), which seemed to be best to include in training sets, had the lowest repeatability.

Advantages of NIRS BLUPS

The NIRS BLUP model mimics the genomic best linear unbiased prediction (G-BLUP) model by building a relationship matrix with the bands of the spectral data. This mimicry made NIRS BLUP models very parsimonious despite large p (independent variables, bands in this case) and small n (number of observations). The G-BLUP is one of the most popular methods in genomic selection because of high prediction accuracies of sample data, and because it is easy to process and implement in conventional statistical software. Thus, these benefits are also applicable when computing the NIRS BLUPS via generation of a NIRS similarity matrix between entries.

Advantages of functional regression analyses

Functional regression analyses accounts for spectral bands as transformed covariates of the original bands, which are of considerably fewer dimensions than the original number of bands. Because of this, functional regression is parsimonious and practical to implement in conventional software for phenomic selection. The functional regression analyses was implemented under a Bayesian approach using conventional genomic selection software (Montesinos-Lopez, et al., 2018).

One problem with the functional regression model is that it is not easy to choose the required number of basis functions (*L*) that give rise to the transformed covariates used for final implementation. This process is somewhat analogous to selecting the number PLS components to use. The correct method for selecting the hyperparameter *L* is via a tuning process. The advantage is that when the bands (original covariates) are strongly correlated with few transformed covariates (*L*=25), it must be enough to capture most of the information existing in the original spectral data.

Advantages of phenomic selection

NIRS BLUP and functional regression both offered parsimonious models for directly including all the spectral data (bands). NIRS BLUP included all the spectral data for building a NIRS relationship matrix, which mimics the genomic relationship matrix of the G-BLUP method of genomic selection. The functional regression analyses uses only a few transformed covariates (*L*=101 in our case) that represent the 3,076 bands originally measured.

Furthermore, the obvious advantage over a simple plot-based prediction method is that the selection methods aim to determine the yield value of a variety, which more easily fits the objectives of most breeders.

Strong potential of NIRS-based phenomic methods

Models traditionally used in NIRS, such as PLSR, as well as more novel approaches implemented by NIRS BLUP and functional regression, showed surprising promise for extracting meaningful information from NIRS of kernels. This discovery aligns with findings by Rincent et al. (2018) for predicting end quality traits in wheat.

In all cases, NIRS in the model allowed a useful prediction ability for grain yield. Some information is available within the NIRS that can be attributed to yield on a plot or entry basis. It is not necessarily important to understand why the NIRS successfully predicts, simply that the prediction is reliable and consistent within the context that it is used. Still, these results pose some interesting, novel fundamental research questions about how NIRS can help understand information that connects to and predicts the yields of plants; it is conceivable there is a novel biological basis to this prediction ability that could open up new areas of research.

Beyond predicting for unknown pedigrees with that share a tester with the training set, there also is an ability to predict unknown pedigrees that are genetically distinct from those in the model calibration. These results suggest that NIRS prediction models have promise to predict unknown pedigrees that have breeding use and value in a program. In our case, the results are limited to the year on which the model is calibrated. However, as results in Fig. 2 show, some of the datasets were poorly predicted, and understanding the underlying reason for this could hold the key for why NIRS phenomic prediction works (or in some cases, does not work). Factors outside of simple genetic relatedness are unlikely to be the main cause, given that plot-based methods sometimes outperformed entry-based methods, especially on genetically unrelated material.

Ultimately, high r values, even when the yield predictions are not highly accurate, suggest that NIRS based models offer a strong potential to be used in ranking varieties relative to one another, as is often done in breeding programs. Furthermore, as suggested by Rincent et al. (2018), there is potential for NIRS to assist in negative selection, to eliminate

varieties with confidence that none of the best lines are being lost, before subjecting a population to more expensive or rigorous screening. The most practical future application would be an eventual relative ranking model to be designed to assist breeders in selection.

One recurrent question surrounding NIRS prediction methods are the optimization of the environment in which to capture scans. Spectra collection in environments separate from the environment in which calibrations are made has presented challenges in the past (Ferrio et al., 2004), and remains a key question to be answered in optimizing selection models using NIRS. It has been shown that NIRS can be used to predict in new environments (Rincent et al., 2018), but understanding what makes the best calibration site is yet to be determined. In this study, where entire environments were held out from the training of the model (CV0), it was always better to build the model using the more stressed environment, which is consistent with the findings of Rincent et al. (2018). Evidence from this experiment and previous studies suggest a more stressful growing condition is likely better for training models using NIRS information.

This current approach has a number of potential direct applications. First, it would be useful in improving plot combine yield data, which would be particularly easy to incorporate into programs with NIRS-capable combines. There is also the potential to carry out single-ear, hand-harvested yield estimates to reduce the amount of plants grown or plots combined. This would be most useful when increasing the environments where limited or unskilled labor might be available for collecting some ears but not the entire plot.

One of the expected pitfalls, as opposed to the successful wheat study, is that maize is an outcrossing crop grown as a hybrid, meaning the grain likely is not a pure representation of

the mother plant, unlike wheat. Fortunately, xenia effects are not usually impactful in maize grain (Seka and Cross, 1995).

It is also likely that the most robust models in a breeding program will be those calibrated with data from multiple years, across a wide diversity of genotypes. This is supported by the improvement made in prediction accuracy between PLSR 5 and PLSR 6 (Table 2), where more genetic material across more environments added to the training set improved prediction results.

Genomic prediction has been rapidly adopted by industry in the improvement of major field crops. The genomic selection approach requires building prediction models using genotyping information to better estimate the breeding value of different varieties within a breeding program (Bernardo et al, 1994; Meuwissen et al., 2001; Bernardo and Yu, 2007; Crossa et al., 2010, 2017; Rincent et al., 2012). Our results suggest that phenomic prediction (such as NIRS of grain yield) might also be useful, although it will be under more restrictive scenarios where the grain composition can be readily assessed, such as in hand harvested experiments and where NIRS is mounted within grain combines, since NIRS was needed for each sample in each environment. Our results also suggest that phenomic selection, where the NIRS is used to make inferences about the value or performance of a variety, may be even more useful, once protocols for the conditions to best collect NIRS calibration information are better established and understood.

Conclusions

This study supports previous evidence from wheat that the use of grain spectral data is useful for predicting grain yield, but for the first time, shows that this approach is valid in

maize. For this reason, these results offer promise for studies in other crops, and will likely guide users to choose a statistical analysis method which best suits their goals. PLSR is routinely used with NIRS, and showed promise for predicting a non-compositional trait on a plot basis. However, the prediction performance obtained with the functional regression method was competitive, and may offer even more value for breeders once protocols are established.

Furthermore, there is reason to suggest that building NIRS prediction models including multiple years will strengthen the prediction accuracy. Future experiments or applications of these methods should aim for more than two years of data to predict subsequent years. Low cost of implementation and evidence provided in this work suggest that breeders interested in this technology should scan as much relevant genetic material as possible when building calibrations, and continually add and update the models just as in genomic selection.

Author Contributions

H.M. Lane: conceptualization, formal analysis, investigation, methodology, software, writing (original draft), review and editing (lead), visualization; S.C. Murray: conceptualization, funding acquisition, methodology, project administration, supervision, resources, review and editing (supporting); O.A. Montesinos-López: software, formal analysis, writing, review and editing (supporting); A. Montesinos-López: software, formal analysis, writing; J. Crossa: software, formal analysis, writing, resources, supervision, review and editing (supporting); D.K. Rooney: data curation; I.D. Barrero-Farfan: data curation; G.N. De La Fuente: data curation; C.L.S. Morgan: methodology, review and editing (supporting).

Conflicts of Interest

The authors declare there are no conflicts of interest.

References

Anderson, S.L., A.L. Mahan, S.C. Murray, and P.E. Klein. 2018. Four parent maize (FPM) population: effects of mating designs on linkage disequilibrium and mapping quantitative traits. The Plant Genome 11(2): 0. doi: 10.3835/plantgenome2017.11.0102.

Anderson, S.L., S.C. Murray, L. Malambo, C. Ratcliff, S. Popescu, et al. 2019. Prediction of maize grain yield before maturity using improved temporal height estimates of unmanned aerial systems. The Plant Phenome Journal. doi: 10.2135/tppj2019.02.0004

Araus, J.L., and J.E. Cairns. 2014. Field high-throughput phenotyping: the new crop breeding frontier. Trends in Plant Science 19(1): 52–61. doi: 10.1016/j.tplants.2013.09.008.

Barrero-Farfan, I.D.B., S.C. Murray, S. Labar, and D. Pietsch. 2013. A multi-environment trial analysis shows slight grain yield improvement in Texas commercial maize. Field Crops Research 149: 167–176. doi: 10.1016/j.fcr.2013.04.017.

Barrero-Farfan, I.D.B., G.N.D.L. Fuente, S.C. Murray, T. Isakeit, P.-C. Huang, M. Warburton, P. Williams, G.L. Windham, and M. Kolomiets. 2015. Genome wide association study for drought, aflatoxin resistance, and important agronomic traits of maize hybrids in the sub-tropics. PLOS ONE 10(2): e0117737. doi: 10.1371/journal.pone.0117737.

Bernardo R.1994. Prediction of maize single-cross performance using RFLPS and information from related hybrids. Crop Science 34:20-25.

Bernardo, R., and J. Yu. 2007. Prospects for genomewide selection for quantitative traits in maize. Crop Science 47(3): 1082–1090. doi: 10.2135/cropsci2006.11.0690.

Bernardo, R. 2008. Molecular markers and selection for complex traits in plants: learning from the last 20 years. Crop Science 48(5): 1649–1664. doi: 10.2135/cropsci2008.03.0131

Bertrand, D., P. Robert, and W. Loisel. 1985. Identification of some wheat varieties by near infrared reflectance spectroscopy. Journal of the Science of Food and Agriculture 36(11): 1120–1124. doi: 10.1002/jsfa.2740361114.

Brummer, E.C., W.T. Barber, S.M. Collier, T.S. Cox, R. Johnson, et al. 2011. Plant breeding for harmony between agriculture and the environment. Frontiers in Ecology and the Environment 9(10): 561–568. doi: 10.1890/100225.

Cabrera-Bosquet, L., J. Crossa, J. von Zitzewitz, M.D. Serret, and J.L. Araus. 2012. High-throughput phenotyping and genomic selection: the frontiers of crop breeding converge. Journal of Integrative Plant Biology 54(5): 312–320. doi: 10.1111/j.1744-7909.2012.01116.x.

Chen, J., X. Ren, Q. Zhang, X. Diao, and Q. Shen. 2013. Determination of protein, total carbohydrates and crude fat contents of foxtail millet using effective wavelengths in NIR spectroscopy. Journal of Cereal Science 58(2): 241–247. doi: 10.1016/j.jcs.2013.07.002.

Chen, L., N.C. Carpita, W.-D. Reiter, R.H. Wilson, C. Jeffries, et al. 1998. A rapid method to screen for cell-wall mutants using discriminant analysis of Fourier transform infrared spectra. The Plant Journal 16(3): 385–392. doi: 10.1046/j.1365-313x.1998.00301.x.

Christman, J.L. 2017. Development of near infrared reflectance spectroscopy (NIRS) calibrations of maize kernel phosphorus for the identification of useful breeding material. https://oaktrust.library.tamu.edu/handle/1969.1/161508 (accessed 3 July 2019).

Cooper, M., O.S. Smith, G. Graham, A. Lane, and et al. 2004. Genomics, genetics, and plant breeding: a private sector perspective. Crop Science; Madison 44(6): 1907–1913.

Crossa, J., de los Campos, G., Pérez-Rodríguez, P., Gianola, D., Burgueño, J., et al. 2010. Prediction of genetic values of quantitative traits in plant breeding using pedigree and molecular markers. Genetics 186 713–724.

Crossa, J. Pérez-Rodríguez, P., Cuevas, J., Montesinos-López, O.A., Jarquín, D., et al. 2017. Genomic selection in plant breeding: methods, models, and perspectives. Trend in Plant Science. 11(22): 961-975. doi.org/10.1016/j.tplants.2017.08.011. ISSN: 1360-1385.

Espinoza, J.A., G.R. Hodge, and W.S. Dvorak. 2012. The potential use of near infrared spectroscopy to discriminate between different pine species and their hybrids. Journal of Near Infrared Spectroscopy 20(4): 437–447. doi: 10.1255/jnirs.1006.

Ferreira, D.S., J.A.L. Pallone, and R.J. Poppi. 2013. Fourier transform near-infrared spectroscopy (FT-NIRS) application to estimate Brazilian soybean [Glycine max (L.) Merril] composition. Food Research International 51(1): 53–58. doi: 10.1016/j.foodres.2012.09.015.

Ferrio, J.P., E. Bertran, M.M. Nachit, J. Català, and J.L. Araus. 2004. Estimation of grain yield by near-infrared reflectance spectroscopy in durum wheat. Euphytica 137(3): 373–380. doi: 10.1023/B:EUPH.0000040523.52707.1e.

Flint-Garcia, S.A., A.-C. Thuillet, J. Yu, G. Pressoir, S.M. Romero, et al. 2005. Maize association population: a high-resolution platform for quantitative trait locus dissection. The Plant Journal 44(6): 1054–1064. doi: 10.1111/j.1365-313X.2005.02591.x.

Foley, W.J., A. McIlwee, I. Lawler, L. Aragones, A.P. Woolnough, et al. 1998. Ecological applications of near infrared reflectance spectroscopy – a tool for rapid, cost-effective prediction of the composition of plant and animal tissues and aspects of animal performance. Oecologia 116(3): 293–305. doi: 10.1007/s004420050591.

Krause, M.R., L. González-Pérez, J. Crossa, P. Pérez-Rodríguez, O. Montesinos-López, et al. 2019. Hyperspectral reflectance-derived relationship matrices for genomic prediction of grain yield in wheat. G3: Genes, Genomes, Genetics: g3.200856.2018. doi: 10.1534/g3.118.200856.

Lang, C., D.R.A. Almeida, and F.R.C. Costa. 2017. Discrimination of taxonomic identity at species, genus and family levels using Fourier Transformed Near-Infrared Spectroscopy (FT-NIR). Forest Ecology and Management 406: 219–227. doi: 10.1016/j.foreco.2017.09.003.

Lee, R.B., and I. DeVore. 1968. Problems in the study of hunters and gatherers.

Levy, S.E., and R.M. Myers. 2016. Advancements in next-generation sequencing. Annual Review of Genomics and Human Genetics 17(1): 95–115. doi: 10.1146/annurev-genom-083115-022413.

Meng, Q., S.C. Murray, A. Mahan, A. Collison, L. Yang, et al. 2015. Rapid estimation of phenolic content in colored maize by near-infrared reflectance spectroscopy and its use in breeding. Crop Science 55(5): 2234–2243. doi: 10.2135/cropsci2014.11.0767.

Meuwissen, T.H.E., B.J. Hayes, and M.E. Goddard. 2001. Prediction of total genetic value using genome-wide dense marker maps. Genetics 157(4): 1819–1829.

Morris, J. S. (2015). Functional regression. Annual Review of Statistics and Its Application, 2, 321-359.

Morris, L.J., and R.T. Holman. 1961. Naturally occurring epoxy acids: II. Detection and measurement of long-chain epoxy acids by near infrared spectrophotometry. J. Lipid Res. 2(1): 77–82.

Montesinos-López, O. A., Montesinos-López, A., Crossa, J., de los campos, G., Alvarado, G., et al. 2017a. Predicting grain yield using canopy hyperspectral reflectance in wheat breeding data. Plant Methods, 13(4): 1-23. DOI 10.1186/s13007-016-0154-2

Montesinos-López, A., Montesinos-López, O. A., Cuevas, J., Mata-López, W. A., Burgueño, J., et al. 2017b. Genomic Bayesian functional regression models with interactions for predicting wheat grain yield using hyper-spectral image data. Plant Methods. Plant Methods 13 (62): 1-29. DOI 10.1186/s13007-017-0212-4.

Montesinos-Lopez A, Montesinos-Lopez OA, de los Campos G, Crossa, J, Burgueño J, Luna-Vazquez FJ. 2018. Bayesian functional regression as an alternative statistical analysis of high-throughput phenotyping data of modern agriculture. Plant Methods. 14:46. Plant Methods (2018) 14:46. https ://doi.org/10.1186/s1300 7-018-0314-7.

Murray, S.C., W.L. Rooney, S.E. Mitchell, A. Sharma, P.E. Klein, et al. 2008a. Genetic improvement of sorghum as a biofuel feedstock: II. QTL for stem and leaf structural carbohydrates. Crop Science 48(6): 2180–2193. doi: 10.2135/cropsci2008.01.0068.

Murray, S.C., A. Sharma, W.L. Rooney, P.E. Klein, J.E. Mullet, et al. 2008b. Genetic improvement of sorghum as a biofuel feedstock: I. QTL for stem sugar and grain nonstructural carbohydrates. Crop Science 48(6): 2165–2179. doi: 10.2135/cropsci2008.01.0016.

Murray, S.C., K. Mayfield, J. Pekar, P. Brown, A. Lorenz, et al. 2019. Tx741, Tx777, Tx779, Tx780, and Tx782 inbred maize lines for yield and southern United States stress adaptation. Journal of Plant Registrations 13(2): 258–269. doi: 10.3198/jpr2017.07.0044crp.

Pugh, N.A., D.W. Horne, S.C. Murray, G. Carvalho, L. Malambo, et al. 2018. Temporal estimates of crop growth in sorghum and maize breeding enabled by unmanned aerial systems. The Plant Phenome Journal 1(1). doi: 10.2135/tppj2017.08.0006.

R Core Team. 2018. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Rinnan, Å., F. van den Berg, and S.B. Engelsen. 2009. Review of the most common pre-processing techniques for near-infrared spectra. TrAC Trends in Analytical Chemistry 28(10): 1201–1222. doi: 10.1016/j.trac.2009.07.007.

Rincent, R., D. Laloë, S. Nicolas, T. Altmann, D. Brunel, et al. 2012. Maximizing the reliability of genomic selection by optimizing the calibration set of reference individuals: comparison of methods in two diverse groups of maize inbreds (*Zea mays L.*). Genetics 192(2): 715–728. doi: 10.1534/genetics.112.141473.

Rincent, R., J.-P. Charpentier, P. Faivre-Rampant, E. Paux, J.L. Gouis, et al. 2018. Phenomic selection is a low-cost and high-throughput method based on indirect predictions: proof of concept on wheat and poplar. G3: Genes, Genomes, Genetics 8(12): 3961–3972. doi: 10.1534/g3.118.200760.

Romay, M.C., M.J. Millard, J.C. Glaubitz, J.A. Peiffer, K.L. Swarts, et al. 2013. Comprehensive genotyping of the USA national maize inbred seed bank. Genome Biol. 14(6): R55. doi: 10.1186/gb-2013-14-6-r55.

Saint Pierre, C., J. Burgueño, J. Crossa, G. Fuentes Dávila, P. Figueroa López, et al. 2016. Genomic prediction models for grain yield of spring bread wheat in diverse agro-ecological zones. Sci Rep 6. doi: 10.1038/srep27312.

Seka, D., and H.Z. Cross. 1995. Xenia and maternal effects on maize agronomic traits at three plant densities. Crop Science 35(1): 86–90. doi: 10.2135/cropsci1995.0011183X003500010015x.

Silva, C.F.L., S.C.K. Milach, S.D.A. Silva, and C.R. Montero. 2008. Near infrared reflectance spectroscopy (NIRS) to assess protein and lipid contents in Avena sativa L. Cropp Breeding and Applied Biotechnology 8(2): 127–133. doi: 10.12702/1984-7033.v08n02a05.

Stevens, A., and L. Ramirez–Lopez. 2014. An introduction to the prospectr package. R Package Vignette, Report No.: R Package Version 0.1 3.

Vermerris, W., A. Saballos, G. Ejeta, N.S. Mosier, M.R. Ladisch, et al. 2007. Molecular breeding to enhance ethanol production from corn and sorghum stover. Crop Science 47(Supplement_3): S-142-S-153. doi: 10.2135/cropsci2007.04.0013IPBS.

Wahl, N., S.C. Murray, T. Isakeit, M. Krakowsky, G.L. Windham, et al. 2017. Identification of resistance to aflatoxin accumulation and yield potential in maize hybrids in the Southeast Regional Aflatoxin Trials (SERAT). Crop Science 57(1): 202–215. doi: 10.2135/cropsci2016.06.0519.

Warburton, M.L., W.P. Williams, G.L. Windham, S.C. Murray, W. Xu, et al. 2013. Phenotypic and genetic characterization of a maize association mapping panel developed for the identification of new sources of resistance to *Aspergillus flavus* and aflatoxin accumulation. Crop Science 53(6): 2374–2383. doi: 10.2135/cropsci2012.10.0616.

Wehrens, R., and B.-H. Mevik. 2007. The pls package: principal component and partial least squares regression in R.

Wolfrum, E., C. Payne, T. Stefaniak, W. Rooney, N. Dighe, et al. 2013. Multivariate calibration models for sorghum composition using near-infrared spectroscopy. National Renewable Energy Lab. (NREL), Golden, CO (United States).

Zhao, Y., M.F. Mette, and J.C. Reif. 2015. Genomic selection in hybrid breeding. Plant Breeding 134(1): 1–10. doi: 10.1111/pbr.12231

CHAPTER III

MODELING TRADEOFFS BETWEEN HIGH THROUGHPUT AND HIGH ACCURACY

PHENOTYPING IN A PLANT BREEDING PROGRAM

Introduction

The History of Plant Breeding

The history of plant breeding can be divided into three relatively defined eras: 1) the early phenotypic selection of our ancestors, based only on visualizable traits; 2) the scientific era, which began in the 20th century, when the underlying heritability of traits first began to be investigated; 3) the genomics era, in which molecular techniques, such as markers, have become abundant and well-utilized (Li et al., 2012).

The latest era has been defined by different genetics methodologies, one of the most notable being quantitative genetics studies, which aimed to identify and map loci which contribute to the complex traits of interest (Tanksley et al., 1982). These quantitative trait loci (QTL) studies focused on pinning down locations of chromosomes important for targeted traits, as well constructing genetic maps and full genome sequences (Cooper et al., 2004). Since then, genomic selection, where genetic markers are used more generally for their predictive capacity within a population, has had much more success, especially in the commercial sectors of both plant and animal breeding (Meuwissen et al., 2001; Heffner et al., 2009; Zhao et al., 2015). Phenotyping capacity remains a bottlenecking, yet key step in training genomic selection models (Bernardo and Yu, 2007; Bernardo, 2008, Rincent et al., 2012).

Tools and technologies may now allow scientific plant breeding to enter a new era: phenomics. This new era will be defined, in part, by novel challenges of data collection and analysis. An increase in data quantity, regardless of quality, also requires resources to be allocated to data storage and the development of data processing pipelines. To do this, a decrease in human labor may be offset in the increase of skill needed in those human resources, and an increase in computing power requirements. Work is needed to address how tradeoffs in resource allocation will affect breeding programs.

High Throughput Phenotyping (HTP)

Remote sensing applications in plant breeding are continuing to both improve and drop in price, with the most commonly used method of data capture being RGB cameras (Araus et al., 2018). Flying over a field and collecting images can increase throughput capacity for data collected to unprecedented levels. Using an unoccupied aerial system (UAS, aka drones), allows the data collection process to be automated and throughput greatly increased, meaning more plants can be measured than before.

Plant height, usually collected at the end of the season to capture terminal growth, is relatively simple to collect manually, though involves a significant amount of manual labor and time, and is therefore a good candidate to benefit by automated collection methods (Anderson et al., 2019). Hand collection for plant height and other data has been previously limited to the labor available, how quickly plants can be measured, and the subjectivity of different individuals' measurements. For example, rice (*Oryza sativa*) plots can be measured manually for plant height at a rate of 45 plots per hour, though a UAS can fly at a rate of 3,000 plots per hour (Tanger et al., 2017), excluding the time needed to extract heights from

the images, which may be substantial. The UAS approach also has the added benefits of measuring an entire plot of plants, as opposed to only a few "representative" plants in a plot that when measured by hand. While UAS data extraction is an added step requiring highly skilled labor, for the time being, this work is often less biologically time-sensitive, and can be done after the original collection out of the field. Additionally, while more computing power is required, fewer people are needed. Flights also can be made more easily throughout the growing season, bringing a temporal dimension to a formerly single-point trait (Anderson et al., 2019).

Plant height is valuable not just as a standalone phenotype, but also shows predictive capacity for grain yield in some environments (Katsvairo et al., 2003; Machado et al., 2002; Mallarino et al., 2003; Barrero-Farfan et al., 2013). Additionally, unlike yield data, collected using machinery such as a combine, collecting plant height via a drone or other UAS, or other HTP methods, is nondestructive, can be done relatively rapidly, or earlier in the season or breeding process (Rincent et al., 2018; Anderson et al., 2019).

Not only can HTP methods improve how we analyze traditional traits of interest, these technologies are also allowing novel phenotypes to be collected invisible to the human eye, and therefore impossible before. This includes using hyperspectral cameras to track plant stress or fertility status (Behmann et al., 2014; Zhao et al., 2005), or using near-infrared reflectance spectroscopy (NIRS) to make inferences about plant performance or breeding value (Rincent et al., 2018; Lane et al., *in review*).

Near Infrared Reflectance Spectroscopy (NIRS)

NIRS is among the oldest tools to be used to increase throughput at the expense of accuracy. Once the initial capital investment has been made for a NIRS machine that fits a program's needs, the cost of scanning samples using NIRS is very inexpensive, limited to labor costs, and the results very repeatable (Shenk and Westerhaus, 1991). While current costs to outsource a sample for wet chemistry typically ranges from $11 to $33 per sample depending on the number of components analyzed, (Ward Laboratories Inc., 2019), the estimated costs per NIRS scan is around $0.40, depending on operator speed. NIRS methods need to be properly calibrated through the appropriate use of wet chemistry. More than 150 wet chemistry samples are recommended to ensure a broadly applicable calibration for a given species, and these should be selected to be spectrally, and perhaps genetically, representative of the populations of samples you wish to analyze (Windham et al., 1989; Wolfrum et al., 2013). This means that after the initial investment of the machine, and calibration development costs, about fifty times the number of samples could be predicted using NIRS. This could be more informative in capturing valuable information about a population, if it allows resources to be extended so that an entire population to be evaluated, or that additional environments can be evaluated. The accuracy of NIRS methods is related to the accuracy of the chemical methods used for calibration, but cost savings over 80% or more can be observed over normal laboratory methods (Aragones, 1997; Foley et al., 1998). Over time, a decrease in equipment costs for NIRS interferometers and other types of machines is making these methods even less cost-prohibitive. One other major advantage is that multiple phenotypes can be predicted simultaneously for maize, for instance, with one scan we can get

estimates on starch, protein, oil, phenolic antioxidants (Meng et al., 2015), among others. Furthermore, we can go back to predict past samples by applying any newly developed calibrations to old spectra (Lane et al., *in review*).

<div align="center">*Simulations and Plant Breeding*</div>

To investigate how HTP can impact a breeding program, there are a few empirical datasets, which would require both HTP and ground truth to be collected on all samples. Computer simulations have been important in scientific research for both generating and testing hypotheses, and began around the same time that computers were invented, in 1945 (Winsburg, 2010). Plant breeding is costly, in terms of both resources and time, making it useful to model experiments to determine the value of a method or breeding scheme before this is conducted (Li et al., 2012). Assessing methods *in silico* is nearly considered a necessary step in determining the efficacy of a potential new method (Sun and Mumm, 2016). Essentially, generating a population for which the 'truth' is known (because it was set in the simulation), allows validation of how well a method can accurately or precisely capture that truth. Such simulations were essential to the development of quantitative genetics and genomics, where such phenomenon as the 'Beavis Effect' (Beavis, 1994; Beavis 1998; Xu, 2003) was discovered, and statistical tools, such as permutation analysis and the genome wide association Q+K model were developed (Churchill and Doerge 1994; Yu et al., 2005; Yu et al., 2006). However, a simulation approach has not, to our knowledge, been applied to HTP tools, to evaluate the experimental advantages that they can bring.

Objectives

In plant breeding, it is usually more desirable to be precise, or to have high repeatability of a measurement to differentiate the best genotypes to keep from those to be discarded, rather than to be accurate, or close to the true value. In the agronomic sense it is important to be accurate, to estimate the true yield of a genotype. In both cases there may be value in sacrificing precision and accuracy of individual plots in exchange for a greater number of measurements, which statistically may increase a genotypes experiment-wise precision and accuracy. In other words, quantity of data may be more usefully predictive than quality of data in a breeding program. For example, using NIRS to predict seed composition, instead of wet chemistry, is less accurate, but much less resource intensive to perform allowing an order of magnitude more samples to be screened. Increasingly, the use of unoccupied aerial systems (UAS aka drones) will be used to take manual agronomic field measurements such as plant height (Anderson et al., 2019; Ashapure et al., 2019).

In this study, we sought to investigate under which circumstances, if any, it is better to use a lower accuracy higher throughput HTP method to capture phenotypes, and at what levels of throughput and error benefits are realized. We investigate both the ability to capture a phenotype (e.g. height) on an entry number basis, as well as the ability to accurately detect genetic effects via single quantitative trait loci (QTL) analysis. This study is directly relevant to the current directions of plant breeding, which appears to be moving from hand measurements to measurements collected via UAS or other automated systems on wider scales.

Materials and Methods

Building genotypes - An F_2 population of 300 individuals was simulated to have 100 single nucleotide polymorphisms (SNPs) with an allele frequency uniformly distributed around 0.5 (between 0.4 and 0.6). Because these alleles were randomly assigned, they are unlinked and without LD. Allele effects were simulated using a geometric distribution with the probability set to 0.3, this yielded a result where approximately one third of the SNPs, on average, would have no effect on the trait of interest (e.g. numbers relevant to maize [*Zea mays* L.] plant height in this case). To compare analysis across population types, F_3 and inbred populations were also assessed, and the population size was expanded to 1000 genotypes via random selection of parents from the F_2. The F_3 generation was generated via simulated selfing of the F_2 population. Unlike an inbred population, heterozygous loci are present and therefore various dominance effects exist. These effects were simulated via a Poisson distribution to have either no dominance (therefore additive) effect, a standard dominance effect where the heterozygote and homozygote have the same effect, or an over-dominance effect, where the heterozygote has a larger effect than the homozygote. To void all dominance effects, the population was immediately advanced to homozygosity into a recombinant inbred line (RIL) population of 1000 individuals, similar to what would be achieved with double haploid technology. The effects of the additive and dominance effects for all loci in each individual were summed to generate that individual's genotypic value.

Building phenotypes - Populations were simulated to be grown in ten environments, where an environmental effect was randomly generated from a normal distribution, centered on zero, with a standard deviation of ten (cm for plant height). Genotype by environment

64

(GxE) effects were randomly generated for each genotype in each environment via a random distribution; the standard deviation for each GxE interactions simulation was itself randomly generated from a normal distribution centered at five with a standard deviation of ten. Next, four replicate effects were added for each rep within each environment. Again, these were normally distributed, centered at zero, with a standard deviation of five. A baseline random error term was then generated for each observation to be equally proportional to the genetic effects based on their variance as presented in code by Tang et al. (2016).

Adding HTP measurement error - To compare error rates in measurement, four residual levels were used. The first level, treated as the more accurate was manual measurements, simply the baseline residual generated above, where genetic and residual variance are proportional to one another. The additional levels tested various levels of additional error that might be added with HTP methods. The baseline residual is increased, via an additional error term, generating using the method described above. The three levels of increased error add an *additional* error value where the residual is: 1) proportional to the size of the genetic effects, making total error variance twice the size of the genetic variance 2) four times the size of the genetic effects, making total error variance five times the size of the genetic variance; 3) is nine times the size of the genetic effects, making total error variance ten times the size of genetic variance. These three levels represent three different HTP methods with varying accuracy. The four levels of error, or measurement types respectively are HTP2x, HTP5x, and HTP10x in addition to the manual measurement.

A multitude of scenarios were tested. The number of genotypes, environments, and replications were varied by taking subsamples from the generated dataset described above.

This process looked at 100, 300, 500, and 1000 genotypes, grown in 1, 2, 5, and 10 environments, replicated 2 and 3 times, in both F_3 or RIL populations. This entire process of simulating and analyzing the population was iterated 50 times.

In each scenario, a best linear unbiased estimator (BLUE) was extracted for each genotype using the lme4 package in R (Bates et al., 2015). Single marker analysis was also conducted for each SNP. This analysis used an ANOVA test to assess if there was a significant effect contributed by the SNP (p-value <0.05), and then an estimate for the size of the effect of the SNP.

To compare each scenario, root mean squared error (RMSE) was calculated based on the simulated, true genetic heights, and the estimated genetic values (BLUEs) from the mixed linear model applied to the phenotypes generated previously. This permitted an assessment of the ability of capturing true genetic heights on an entry number basis. The single-marker analysis served to detect quantitative trait loci (QTL), and then these scenarios were compared via the proportion of true QTL that were detected, as well as what proportion of all of the determined QTL were false (an effect was detected where none was simulated).

Initial analysis indicated that increasing the number of genotypes and reps had a minimal effect on reducing RMSE. For this reason, a second loop was designed to simulate and assess scenarios where 300 genotypes of both F_3 and RIL populations were analyzed in up to 25 environments.

Datasets and models were simulated in R (R Core Team, 2018), using custom scripts (Appendix C) and packages *lme4* and *dplyr* (Bates et al., 2015; Wickham et al., 2018).

Results and Discussion

Replicates and environments effects on genotypic accuracy

Increasing replicates and environments improved the ability of statistical model to detect the true genetic value of an individual genotype, on an entry number basis, as shown by a decrease in RMSE (Fig.3). The increase in throughput needed to overcome an increased error rate for the same RMSE level was dependent on the level of error added (HTP2x, HTP5x, HTP10x). Overall, the biggest improvement in accuracy, in terms of RMSE for genotype predictions was seen in our scenarios via an increase in the number of environments in which the population was assessed (Fig. 3), while the number of replicates made only a marginal improvement, though this was directly correlated to the size of replicate effect generated in the simulation code.
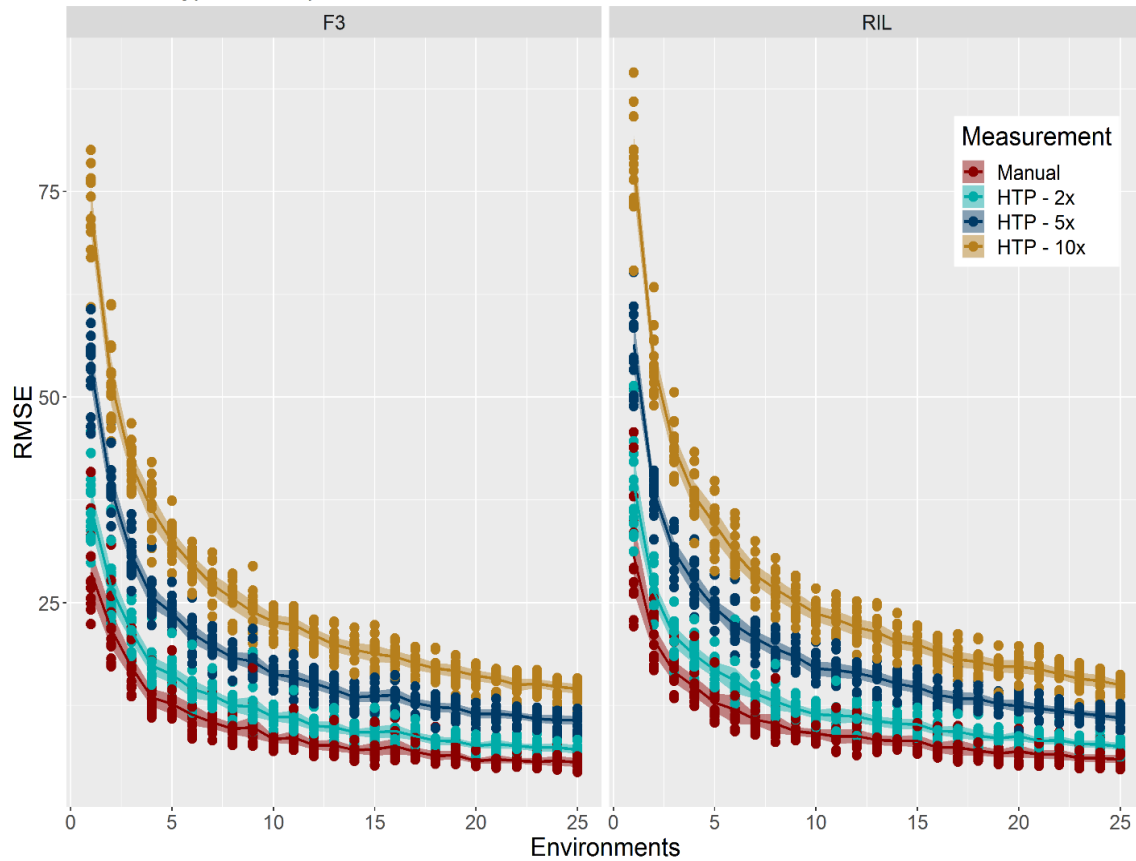
Figure 3. Four different error rates of measurement (manual, HTP with 2x error, HTP with 5x error, and HTP with 10x error) compared across a differing number of environments. The left panel shows the results of the scenarios of an F₃ population, while the right shows a recombinant inbred line (RIL) population, both had 300 genotypes replicated two times in each environment. Each scenario was repeated ten times. The shaded area shows a 95% confidence interval for the mean, the lines show the mean RMSE for each scenario.

Accuracy of QTL estimation

To capture QTL, results showed that combinations of increasing the number of environments and the number of genotypes was important in capturing QTL. Similar results have been found both by other studies in both simulation (Beavis et al., 1998; Yu et al., 2005) and empirically (Beavis et al., 1994; Anderson et al. 2018). The proportion of true QTL that were detected in each scenario (Fig. 4) increased around 10% more greatly by doubling the genotypes than by doubling the environments: this would require the same increased number of observations, but would increase development costs to create these larger populations. When using one environment, error rate had a very large effect on the ability to identify QTL, but increasing the number of environments decreased the effects of measurement methods, where HTP methods with lower accuracy were nearly as effective as the more accurate manual methods. Increasing genotypes was the best way to capture more QTL, especially if the number of environments was held constant.
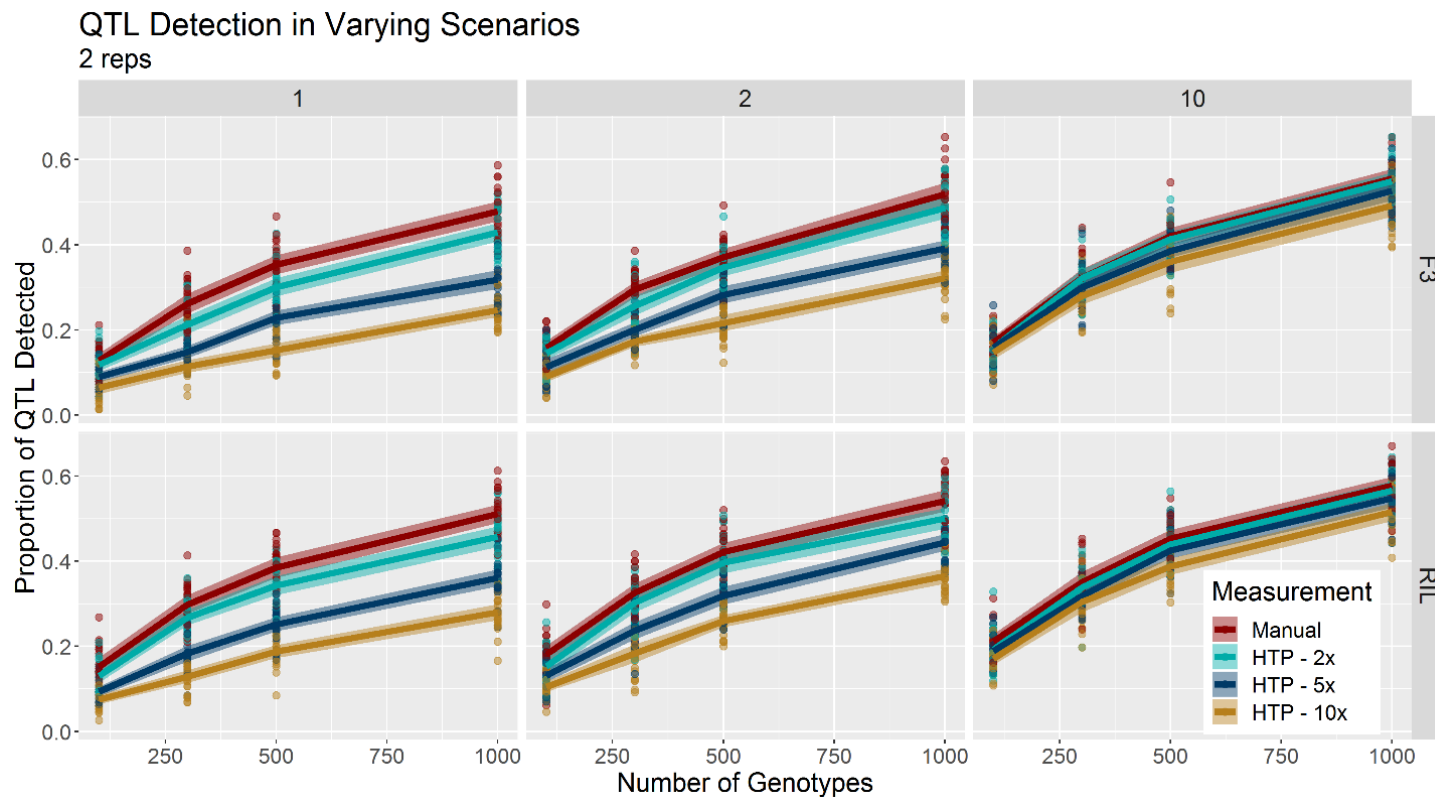
Figure 4. The proportion of the simulated, true quantitative trait loci accurately captured across varying numbers of genotypes assessed. The figure is faceted along the y-axis to compare F_3 and recombinant inbred line (RIL) populations, and faceted along the x-axis to compare increasing numbers of environments. The lines show the mean across each scenario after replication, the shading represents the standard error of the mean.

The simulated QTL were compared to the statistically detected QTL to assess what proportion of the detected QTL were real versus which were false positives, where the model detected an additive effect on height by a SNP when there was no effect (Fig. 5). It was nearly impossible to completely reduce the risk of capturing false QTL, but increasing both the number of genotypes and environments decreased chances greatly. Again, the method of measurement did not greatly impact the risk of false QTL detection, except in the cases of the fewest number of genotypes in a single environment. Even with the lowest error rate (manual measurements), growing a small population (100-300 genotypes), in one environment presented the likeliness that in some cases, more than 20% of detected QTL would not have a true effect on the trait of interest. This risk was mainly present for small effect loci (effect < 4 cm), and was less of an issue in large effect SNPs (effect > 6 cm). This propensity in small populations, and especially those with heterozygosity (F_2 and F_3) has been termed the Beavis effect (Beavis et al., 1998; Xu 2003).
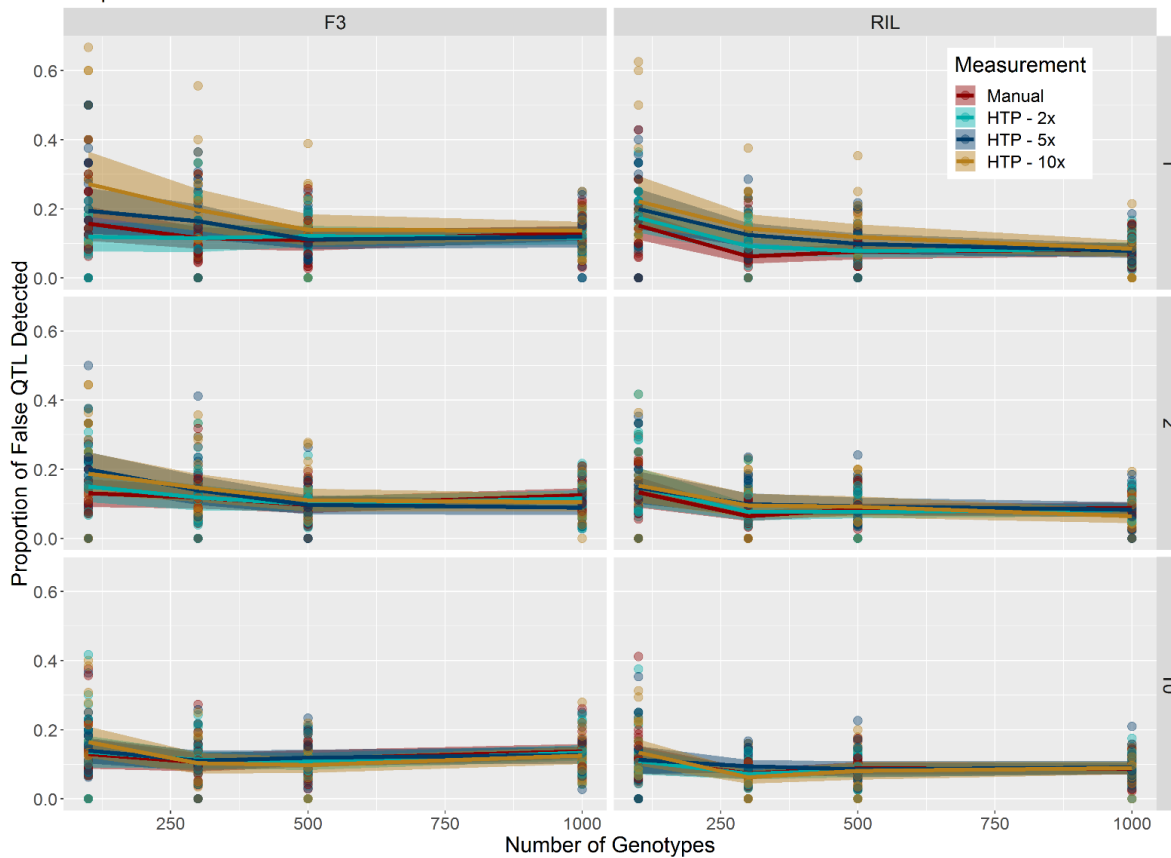
Figure 5. The proportion of detected quantitative trait loci in varying scenarios that in reality had no effect on the trait across increasing numbers of genotypes analyzed. The figure is faceted along the x axis to compare F_3 and recombinant inbred line (RIL) populations, and faceted along the y axis to compare increasing numbers of environments. The lines show the mean across each scenario after replication, the shading represents the standard error of the mean.

Throughout the process of running different simulations under the described parameters, it became clear that increasing the number of environments had the largest impact on improving the accuracy of a scenario in terms of the ability to accurately predict a genotype's simulated, true genetic height on an entry number basis, under both population

types assessed (Fig. 4). It is not unrealistic, and it is nearly mandatory, for a breeding program to test genotypes in many environments. These environments are occasionally obtained within the same location through varied planting dates and/or input and irrigation regimens. Across five locations, 16 environments were reported in one heterosis study in maize (Li et al., 2018), and breeding programs may even subject genotypes to hundreds of environments over the years (Gaffney et al., 2015). At minimum, a program could plant irrigated and unirrigated trials, or an optimal and late planting. The results of these simulations show that increasing environments is more impactful than increasing replications beyond two.

Increasing the number of genotypes analyzed within a population did not improve RMSE values, but rather made RMSE more consistent between iterations. Therefore, we hypothesize that if the goal is the most accurate estimate of genotypic values, resources would be better allocated in increasing the number of environments in an experiment, rather than spending them on increasing genotypes or replications within fewer experiments.

In contrast, when the ability to identify, and correctly predict the effects of QTL is of the most interest, the number of genotypes plays a significant role, with the number of environments being secondary. Assessing QTL under only one environment resulted in the highest rates of false positives (Fig. 6). Evaluating under only one environment, a higher error rate affects the ability to detect QTL greatly, and there is little difference between measurement types as the number of environments increases (Fig. 4). This suggests that high quantity phenotyping (e.g. HTP) become more valuable over high accuracy phenotyping as more environments are used, because high quantity phenotyping is also more affordable. Furthermore, even under the most rigorous testing scenarios, with 1000 genotypes in 10

environments, the risk of capturing false small effect QTL is never completely eliminated (Fig. 5).

Additionally, according to these results, increasing the accuracy of measurements does not seem to have any effect on reducing the risk of capturing false QTL (Fig. 5). Only increasing the number of genotypes assessed sufficiently had a real decrease in false QTL identified. Ultimately, investigators who are aiming to better understand genetic effects should be aiming to analyze the biggest population feasible under as many environments as possible. This is consistent with other authors empirical findings (Anderson et al. 2018; Beavis et al., 1994) and simulated findings (Beavis et al., 1998; Zhu et al., 2008) It is difficult, and unlikely, to accurately distinguish true genetic effects in limited environmental context, and is more likely that SNPs serve a purpose for predictive capacity. SNPs with relatively large effects were less likely to be falsely identified, so the risk of false identification is higher the smaller the predicted effect. This predictive capacity further suggests why genomic selection is successful when properly calibrated for the population assessed (Heffner et al., 2009).

Advantages of HTP

There are cases where the accuracy of phenotyping cannot be compared to "by-hand" measurements as they are completely novel and not visible without these new phenomic technologies. Furthermore, the "ground truth" measurements taken manually may also not necessarily be more accurate or precise than their high throughput counterpart, just because their values do not align. A higher repeatability of high throughput measurements in some cases (as in Pugh et al., 2018) may indicate that the automated phenotyping may in fact be

more precise. Ashapure et al. (2019) detected differences in UAS-derived phenotypes that were undetectable with infield measures due to a much greater number of measures. In the case of Anderson et al. (2019), the UAS height measurements actually allowed for more of the variation to be explained, binning less into residual.

Limitations

The simulation used here, like all simulations, presents some limitations due its simplistic nature of modeling complex biology and biological interactions. While simplicity offers some generalizability across organisms and applications, it also presents limitations in authenticity and its ability to very accurately simulate real-life, genetic conditions. Despite being a rather basic avenue through which to compare varying error rates, the objectives are still addressed and offer valuable insights to breeders and geneticists looking at expanding the throughput of their phenotyping methods, particularly those who feel their accuracy may be decreased by doing so. Additionally, the simplicity of the simulation and analysis code offers relatively easy expansion or investigation by those who may be more interested in comparing their own conditions.

Conclusions

In general this study demonstrates that that it is efficient and prudent to replace high-accuracy measurements with high-throughput ones where these gains allow more genotypes and environments to be evaluated. Future empirical and simulation studies should consider how linkage and kinship could affect genetics studies, and also should consider how HTP methods may offer advantages distinct from accuracy levels.

References

Anderson, S.L., A.L. Mahan, S.C. Murray, and P.E. Klein. 2018. Four Parent Maize (FPM) Population: Effects of Mating Designs on Linkage Disequilibrium and Mapping Quantitative Traits. The Plant Genome 11(2): 0. doi: 10.3835/plantgenome2017.11.0102.

Anderson, S.L., S.C. Murray, L. Malambo, C. Ratcliff, S. Popescu, et al. 2019. Prediction of Maize Grain Yield before Maturity Using Improved Temporal Height Estimates of Unmanned Aerial Systems. The Plant Phenome Journal 2(1). doi: 10.2135/tppj2019.02.0004.

Araus, J.L., S.C. Kefauver, M. Zaman-Allah, M.S. Olsen, and J.E. Cairns. 2018. Translating High-Throughput Phenotyping into Genetic Gain. Trends in Plant Science 23(5): 451–466. doi: 10.1016/j.tplants.2018.02.001.

Ashapure, A, J. Jung, J. Yeom, A. Chang, M. Maeda, A. Maeda, and J. Landivar. 2019. A novel framework to detect conventional tillage and no-tillage cropping system effect on cotton growth and development using multi-temporal UAS data. ISPRS J Photogramm. 152:49-64.

Barrero Farfan, I.D., S.C. Murray, S. Labar, and D. Pietsch. 2013. A multi-environment trial analysis shows slight grain yield improvement in Texas commercial maize. Field Crops Research 149: 167–176. doi: 10.1016/j.fcr.2013.04.017.

Bates, D., M. Maechler, B. Bolker, S. Walker. 2015. Fitting Linear Mixed-Effects Models Using lme4. Journal of Statistical Software, 67(1), 1-48. doi:10.18637/jss.v067.i01.

Beavis, W. 1994. The power and deceit of QTL experiments: lessons from comparative QTL studies. Proceedings of the forty-ninth annual corn and sorghum industry research conference. Washington, DC. p. 250–266

Beavis, W.D. 1998. QTL analyses: power, precision, and accuracy. Molecular dissection of complex traits. p. 145–162

Behmann, J., J. Steinrücken, and L. Plümer. 2014. Detection of early plant stress responses in hyperspectral images. ISPRS Journal of Photogrammetry and Remote Sensing 93: 98–111. doi: 10.1016/j.isprsjprs.2014.03.016.

Bernardo, R., and J. Yu. 2007. Prospects for Genomewide Selection for Quantitative Traits in Maize. Crop Science 47(3): 1082–1090. doi: 10.2135/cropsci2006.11.0690.

Bernardo, R. 2008. Molecular Markers and Selection for Complex Traits in Plants: Learning from the Last 20 Years. Crop Science 48(5): 1649–1664. doi: 10.2135/cropsci2008.03.0131

Churchill, G.A., and R.W. Doerge. 1994. Empirical threshold values for quantitative trait mapping. Genetics 138(3): 963–971.

Cooper, M., O.S. Smith, G. Graham, A. Lane, and et al. 2004. Genomics, Genetics, and Plant Breeding: A Private Sector Perspective. Crop Science; Madison 44(6): 1907–1913.

Heffner, E.L., M.E. Sorrells, and J.-L. Jannink. 2009. Genomic Selection for Crop Improvement. Crop Science 49(1): 1–12. doi: 10.2135/cropsci2008.08.0512.

Katsvairo, T.W., W.J. Cox, and H.M. Van Es. 2003. Spatial Growth and Nitrogen Uptake Variability of Corn at Two Nitrogen Levels. Agronomy Journal 95(4): 1000–1011. doi: 10.2134/agronj2003.1000.

Lane, H.M., S.C. Murray, J. Crossa, O.A. Montesinos-López, A. Montesinos-López, et al. *In Review*. Phenomic Prediction of Maize Grain Yield from Near-infrared Reflectance Spectroscopy of Kernels. The Plant Phenome Journal.

Li, Z., L. Coffey, J. Garfin, N.D. Miller, M.R. White, et al. 2018. Genotype-by-environment interactions affecting heterosis in maize. PLOS ONE 13(1): e0191321. doi: 10.1371/journal.pone.0191321.

Li, X., C. Zhu, J. Wang, and J. Yu. 2012. Chapter six - Computer Simulation in Plant Breeding. In: Sparks, D.L., editor, Advances in Agronomy. Academic Press. p. 219–264

Machado, S., E.D. Bynum, T.L. Archer, R.J. Lascano, L.T. Wilson, et al. 2002. Spatial and Temporal Variability of Corn Growth and Grain Yield. Crop Science 42(5): 1564–1576. doi: 10.2135/cropsci2002.1564.

Mallarino, A.P., E.S. Oyarzabal, and P.N. Hinz. 1999. Interpreting Within-Field Relationships Between Crop Yields and Soil and Plant Variables Using Factor Analysis. Precision Agriculture 1(1): 15–25. doi: 10.1023/A:1009940700478.

Meng, Q., S.C. Murray, A. Mahan, A. Collison, L. Yang, et al. 2015. Rapid Estimation of Phenolic Content in Colored Maize by Near-Infrared Reflectance Spectroscopy and Its Use in Breeding. Crop Science 55(5): 2234–2243. doi: 10.2135/cropsci2014.11.0767.

Meuwissen, T.H.E., B.J. Hayes, and M.E. Goddard. 2001. Prediction of Total Genetic Value Using Genome-Wide Dense Marker Maps. Genetics 157(4): 1819–1829.

R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL: https://www.R-project.org/.

Rincent, R., D. Laloë, S. Nicolas, T. Altmann, D. Brunel, et al. 2012. Maximizing the reliability of genomic selection by optimizing the calibration set of reference individuals: comparison of methods in two diverse groups of maize inbreds (Zea mays L.). Genetics 192(2): 715–728. doi: 10.1534/genetics.112.141473.

Rincent, R., J.-P. Charpentier, P. Faivre-Rampant, E. Paux, J.L. Gouis, et al. 2018. Phenomic Selection Is a Low-Cost and High-Throughput Method Based on Indirect Predictions: Proof of Concept on Wheat and Poplar. G3: Genes, Genom

Sun, X., and R.H. Mumm. 2016. Method to represent the distribution of QTL additive and dominance effects associated with quantitative traits in computer simulation. BMC Bioinformatics 17(1): 73. doi: 10.1186/s12859-016-0906-z.es, Genetics 8(12): 3961–3972. doi: 10.1534/g3.118.200760.

Tanger, P., S. Klassen, J.P. Mojica, J.T. Lovell, B.T. Moyers, et al. 2017. Field-based high throughput phenotyping rapidly identifies genomic regions controlling yield components in rice. Scientific Reports 7: 42839. doi: 10.1038/srep42839.

Ward Laboratories Inc. Feed/NIR Services. https://www.wardlab.com/feed-nir-services.php (accessed 12 August 2019).

Wickham H., R. François, L. Henry and K. Müller. 2018. dplyr: A Grammar of Data Manipulation. R package version 0.7.8. https://CRAN.R-project.org/package=dplyr

Windham, W., D. Mertens, and F. Barton. 1989. Protocol for NIRS calibration: sample selection and equation development and validation. Near infrared reflectance spectroscopy (NIRS): Analysis of forage quality: 96–103.

Winsberg, E. 2010. Science in the Age of Computer Simulation. University of Chicago Press, Chicago, UNITED STATES.

Wolfrum, E., C. Payne, T. Stefaniak, W. Rooney, N. Dighe, et al. 2013. Multivariate Calibration Models for Sorghum Composition using Near-Infrared Spectroscopy. National Renewable Energy Lab. (NREL), Golden, CO (United States).

Yu, J., M. Arbelbide, and R. Bernardo. 2005. Power of in silico QTL mapping from phenotypic, pedigree, and marker data in a hybrid breeding program. Theor Appl Genet 110(6): 1061–1067. doi: 10.1007/s00122-005-1926-7.

Yu, J., G. Pressoir, W.H. Briggs, I.V. Bi, M. Yamasaki, et al. 2006. A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. Nat Genet 38(2): 203–208. doi: 10.1038/ng1702.

Xu, S. 2003. Theoretical Basis of the Beavis Effect. Genetics 165(4): 2259–2268.

Zhao, D., K.R. Reddy, V.G. Kakani, and V.R. Reddy. 2005. Nitrogen deficiency effects on plant growth, leaf photosynthesis, and hyperspectral reflectance properties of sorghum. European Journal of Agronomy 22(4): 391–403. doi: 10.1016/j.eja.2004.06.005.

Zhao, Y., M.F. Mette, and J.C. Reif. 2015. Genomic selection in hybrid breeding. Plant Breeding 134(1): 1–10. doi: 10.1111/pbr.12231

Zhu, C., M. Gore, E.S. Buckler, and J. Yu. 2008. Status and Prospects of Association Mapping in Plants. The Plant Genome 1(1): 5–20. doi: 10.3835/plantgenome2008.02.0089.

CHAPTER IV

CONCLUSIONS, REFLECTIONS, AND FUTURE DIRECTIONS

Conclusions

Phenomic technologies are increasing in both interest and investment, and offer promise to improve breeding programs of any crop, though maize was the primary species of interest investigated in this research. This work indicates strong capacity for NIRS to expand beyond its traditional use, and that increasing throughput in phenotyping by better allocating resources through these and other technologies will allow breeders to leverage data quantity, even if quality of data cannot be maintained. As throughput is increased, it is likely data quality can be sacrificed to save money and other resources.

Moving Forward

There are future opportunities to explore the genetic marker information for the varieties assessed with NIRS in this work, and to compare the efficacy of phenomic selection approaches to genomic selection approaches in the same sample set, as marker information is available for these lines. Additionally, it is unclear whether or not the predictive capacity of the NIRS stemmed from biological and causal relationships (e.g. starch or protein content), or from detecting genetic relatedness, which was investigated with the NIRS BLUP and functional regression models. Based on the results of plot-based analysis methods, however, it is likely that the compositional predictive capacity of NIRS is also playing a role in success, especially given its superior capacity to predict unfamiliar genetic material. Diving into the genotyping data for the lines in this study may hold the key in better elucidating the nature of these phenomic predictions, and better understanding the best environments to capture NIRS.

As HTP methods and options for breeders continue to improve and increase in number, breeders must weigh whether or not investing in these new technologies will offer significant benefit to their program. This research has shown that increasing throughput stands to greatly increase the capture of genetic information and predictive capacities for various model types, even when the quality of data may not be maintained. It is likely that many breeders in both the public and private sectors stand to benefit from the increased tools made available to them through HTP platforms.

APPENDIX A

NOMENCLATURE

5FRCV	Five-fold random cross validation
AF	Aflatoxin
BLUE	Best linear unbiased estimator
BLUP	Best linear unbiased predictor
CV	Cross validation
CV	Cross validation
CV0	Predicting one environment using all other environments
CV1	20% of hybrids are predicted by the remaining 80% (five-fold), within each environment
CV2	Similar to CV1, but used across environments, where hybrids are seen in 1 environment but predicted in another (mimics sparse testing)
G-BLUP	Genomic best linear unbiased predictor, a genomic similarity matrix
GWAS	Genome-wide association study
GxE	Genotype by environment
HTP	High throughput phenotyping
LM	Simple linear model
NIRS	Near-Infrared Reflectance Spectroscopy
NRMSE	Normalized root mean squared error
r	Pearson's Correlation
PLSR	Partial Least Squares Regression

RMSE   Root mean square error

RMSEP Root mean square error of prediction

SE   Standard error

SNPs   Single nucleotide polymorphisms

UAS   Unoccupied aerial systems

WS   Water stress; unirrigated treatment

WW   Well-watered; irrigated treatment

APPENDIX B

PLSR AND LM R SCRIPTS

Building PLSR Models 1, 3-5, 7, 8:

```
## To convert yield from BU/AC to G/M^2, multiply yield by (6.271)

setwd("C:/Users/holly.lane/Project/Manuscript Files") ## set this for your system
library(reshape); library(reshape2);library(ggplot2); library(dplyr)
library(prospectr);library(RcppArmadillo);library(Rcpp); library(foreach) ; library(iterators)
library(resemble);require(prospectr);library(ggfortify);library(cluster); library(pls)
##############################################################################
#########

# First, we clean the data and transform it with an SG derivative

##############################################################################
#########
wholecorn<- read.csv("Supplemental 1.csv", header=TRUE, check.names = FALSE)
#check.names prevents R from placing an X in front of the wavelength column names
dim(wholecorn)
colnames(wholecorn[,1:83])
##forcing R to accept numeric column names makes the column names in valid
##extract spectra columns before fixing this
wholecornspc <- wholecorn[,83:3194]
wholespc <- data.matrix(wholecornspc)
##now fix column names
valid_column_names <- make.names(names=names(wholecorn), unique=TRUE, allow_ =
TRUE)
names(wholecorn) <- valid_column_names
#select informational columns of interest, then attach matrix of spectra as variable
wholecornfull<- select(wholecorn,c(5,6,7,8,9,21,22,23,25,32,43,77,78))
wholecornfull$year<-as.factor(wholecornfull$year)
wholecornfull$spc<-wholecornspc
###in example dataset, NIRsoil, spectra is variable spc as a matrix within "dataframe"
wholecornfull$raw<-wholespc

#sum(wholecornfull$yield.buac<10) #47 removed

wholecornya<-wholecornfull[!wholecornfull$yield.buac<10,]
##############################################################################
########
```

```
whole.sg <- savitzkyGolay(wholecornya$spc, p = 3, w = 37, m = 1)
whole.sg2 <- savitzkyGolay(wholecornya$spc, p = 3, w = 37, m = 2) ## adjust w
wholecornya$sg1<-whole.sg
wholecornya$sg2<-whole.sg2

wc<-wholecornya

##################################################################################
#########
#create average spectra to compare to as baseline
avewc <- wc [c(10, 50, 100, 222, 500, 300, 1300, 2000),]

avewc1 <- wc[1,]

avewc1$sg1<-t(as.matrix(colMeans(avewc$sg1)))

aveSpec <- avewc1$sg1
####################
aveSpec <-matrix(aveSpec, nrow= nrow(wc$sg1), ncol=ncol(aveSpec), byrow=TRUE)

wc$diff<- wc$sg1 - aveSpec

wc$sumdiff<-rowSums (abs(wc$diff))

hist(wc$sumdiff, breaks=200) #see the distribution of differences
###################
sum(wc$sumdiff>.5)

sum(wc$sumdiff>.4)

wholecornCLEAN<-wc[!wc$sumdiff>.4,]# Final threshold selected to remove plots with poor
scans

wc<-wholecornCLEAN

hist(wc$sumdiff, breaks=200)

rownames(wc) <- seq(length=nrow(wc))

##################################################################################
##################
#we removed two extra odd points, identified here
```

```
wc.pls <- plsr(yield.buac~sg1, data= wc, ncomp=15, validation="CV")
summary(wc.pls)
plot(wc.pls, ncomp = 15, asp = 1, line = TRUE)

plot(RMSEP(wc.pls), legendpos = "topright")

plot(wc.pls, plottype = "scores", comps = 1:10)

plot(x= wc.pls$scores[,5], y= wc.pls$scores[,9])

#identify(x= wc.pls$scores[,5], y= wc.pls$scores[,9], plot=TRUE)

wc2<- wc[-c(1778, 2032),]

wc<-wc2
rownames(wc) <- seq(length=nrow(wc))
#####################################################################################
##################
#####################################################################################
################

hist(wc$yield.buac)
wc11 <- wc[(wc$year=="2011"),]
wc12 <-wc[(wc$year=="2012"),]
rownames(wc11) <- seq(length=nrow(wc11))
rownames(wc12) <- seq(length=nrow(wc12))

## below are histograms to compare the difference between years in yield distribution

hist(wc11$yield.buac,
    main="Distribution of 2011 Yield",
    xlab = expression(paste(" Yield ", "(bu ac"^"-1" ,")" )),
    xlim=c(0,250),
    ylim=c(0,175),
    cex.axis=1.25,
    col="#003A66", #poster color
    #col = "darkred", #presentation color
    las=1,
    breaks=20)
range(wc11$yield.buac)

hist(wc12$yield.buac,
    main="Distribution of 2012 Yield",
```

```
    xlab = expression(paste(" Yield ", "(bu ac"^"-1" ,")" )),
    cex.axis=1.25,
    xlim=c(0,250),
    ylim=c(0,175),
    col="#003A66", #poster color
    #col = "darkred", #presentation color
    las=1,
    breaks=20)
range(wc12$yield.buac)


##############################################################################
##
###
## Importing, cleaning, and filtering 2011 experiments

##############################################################################
#########
corn2011<- read.csv("Supplemental 3.csv", header=TRUE, check.names = FALSE)
#check.names prevents R from placing an X in front of the wavelength column names
dim(corn2011)
##forcing R to accept numeric column names makes the column names in valid
##extract spectra columns before fixing this
colnames(corn2011[,1:80])
corn2011spc <- corn2011[,30:3141]
spc2011 <- data.matrix(corn2011spc)
##now fix column names
valid_column_names <- make.names(names=names(corn2011), unique=TRUE, allow_ =
TRUE)
names(corn2011) <- valid_column_names
#select informational columns of interest, then attach matrix of spectra as variable
corn2011full<- select(corn2011,c(1,2,3,4,9:12,20))
corn2011full$year<-as.factor(corn2011full$year)

corn2011full$spc<-corn2011spc
###in example dataset, NIRsoil, spectra is variable spc as a matrix within "dataframe"
corn2011full$raw<-spc2011

#sum(corn2011full$yield.buac<10) #47 removed

corn2011ya<-corn2011full[!corn2011full$yield.buac<10,]
##############################################################################
########
```

```
whole.sg <- savitzkyGolay(corn2011ya$spc, p = 3, w = 37, m = 1)
whole.sg2 <- savitzkyGolay(corn2011ya$spc, p = 3, w = 37, m = 2) ## adjust w
corn2011ya$sg1<-whole.sg
corn2011ya$sg2<-whole.sg2

MORE2011<-corn2011ya

#MORE2011<-na.omit(MORE2011) #NAs are present in the dataset for different variables

MORE2011<-na.omit(MORE2011, cols= "yield.buac")
#############################################################################
########

aveMORE2011 <- MORE2011

aveMORE20111 <- MORE2011[1,]
aveMORE20111$sg1<-t(as.matrix(colMeans(aveMORE2011$sg1)))


aveSpec <- aveMORE20111$sg1

####################

aveSpec <-matrix(aveSpec, nrow= nrow(MORE2011$sg1), ncol=ncol(aveSpec),
byrow=TRUE)

MORE2011$diff<- MORE2011$sg1 - aveSpec

MORE2011$sumdiff<-rowSums (abs(MORE2011$diff))

hist(MORE2011$sumdiff, breaks=200)

####################

sum(MORE2011$sumdiff>.5)

which(MORE2011$sumdiff>.5)

sum(MORE2011$sumdiff>.35)

which(MORE2011$sumdiff>.4)

MORE2011CLEAN<-MORE2011[!MORE2011$sumdiff>.35,] ##Final threshold selected
```

MORE2011<-MORE2011CLEAN

#hist(MORE2011$sumdiff, breaks=200)

rownames(MORE2011) <- seq(length=nrow(MORE2011))

##
########
again, two odd points identified and removed

pls.11<- plsr(yield.buac~sg1, data= MORE2011, ncomp=15, validation="CV")
summary(pls.11)
plot(pls.11, ncomp = 15, asp = 1, line = TRUE)

plot(RMSEP(pls.11), legendpos = "topright")

plot(pls.11, plottype = "scores", comps = 1:10)

plot(x= pls.11$scores[,7], y= pls.11$scores[,1])

#identify(x= pls.11$scores[,7], y= pls.11$scores[,1], plot=TRUE)

MORE20112<- MORE2011[-598,]

MORE2011<-MORE20112
rownames(MORE2011) <- seq(length=nrow(MORE2011))
MORE20112<- MORE2011[-157,]

MORE2011<-MORE20112
rownames(MORE2011) <- seq(length=nrow(MORE2011))

##
#
Now we can start building testing and training sets for PLSR models
#
##

# Need to select a representative validation set to test our PLS on, using DUPLEX method
wc.duplex<-duplex(X=wc$sg1,k=120,pc=10) #takes awhile to run

```
plot(wc.duplex$pc)
points(wc.duplex$pc[wc.duplex$model, 1],wc.duplex$pc[wc.duplex$model, 2], col = "red",
pch = 19)
points(wc.duplex$pc[wc.duplex$test, 1], wc.duplex$pc[wc.duplex$test, 2], col = "blue", pch
= 19)
legend("topright", legend = c("calibration", "validation"), pch = c(19, 19),
    col = c("red", "blue"))

##result is list with row index for calibration(model)/and validation(test) samples

testrows<-wc.duplex$test
testrows<-sort.int(testrows)

wcTest <- wc[(testrows),]

wcTrain <-wc[-testrows,]

levels(droplevels(wc$pedigree))
# 344 represented -- NOTE: Some may be missing from yield adjustments

levels(droplevels(wcTest$pedigree))
# 91 represented, need to eliminate all from training set

levels(droplevels(wcTrain$pedigree))
#343 represented
##############################################################################
##############
wcTest.pedigree<-wc[wc$pedigree %in% wcTest$pedigree, ]

levels(droplevels(wcTest.pedigree$pedigree))
dim(wcTest)
dim(wcTest.pedigree)
wcTest.pedigree[1:10,1:5]

wcTrain.pedigree<-wc[!wc$pedigree %in% wcTest$pedigree, ]

dim(wcTrain)
dim(wcTrain.pedigree)
wcTrain.pedigree[1:10,1:5]

levels(droplevels(wcTrain.pedigree$pedigree))
#253 pedigrees in training set
```

```
################################################################################
#################
wcTest<-wcTest.pedigree
wcTrain<-wcTrain.pedigree
################################################################################
#################
whole.pls.ya <- plsr(yield.buac~sg1, data= wcTrain, ncomp=50, validation="CV")
#now we have built PLSR 1

summary(whole.pls.ya)
plot(whole.pls.ya, ncomp = 25, asp = 1, line = TRUE)
####
#modelresults<-sort.default(whole.pls.ya$validation$pred[,,15])
##
RMSEP(whole.pls.ya, newdata=wcTest)

plot(RMSEP(whole.pls.ya), legendpos = "topright")

ncomp.onesigma <- selectNcomp(whole.pls.ya, method = "onesigma", plot = TRUE)
ncomp.permut <- selectNcomp(whole.pls.ya, method = "randomization", plot = TRUE)

plot(RMSEP(whole.pls.ya, newdata=wcTest), legendpos = "topright")

plot(R2(whole.pls.ya, newdata=wcTest))

ncomp.onesigma <- selectNcomp(whole.pls.ya, method = "onesigma", plot = TRUE,
                  main = "Component Selection")

##### PLOT
rmsep<-RMSEP(whole.pls.ya, newdata=wcTest, ncomp=10)
R2(whole.pls.ya, newdata=wcTest, ncomp=10)

regline<- lm((predict(whole.pls.ya, ncomp=10, newdata=wcTest))~ wcTest$yield.buac)

predplot(whole.pls.ya, ncomp = 10, newdata = wcTest, line = TRUE,
     cex.axis = 1.5, cex.lab = 1.5,cex.main=2,
     pch = 16, main = "Global Prediction Ability (NIRS)",
     xlab = expression(paste("True Yield ", "(bu ac"^"-1" ,")" )),
     ylab = expression(paste("Predicted Yield ", "(bu ac"^"-1" ,")" )),
     las=1)+
  abline(a=0,b=1,col="firebrick3",lwd=2)
abline(regline, lwd=2)
```

```
legend(x=140, y=60, legend= paste("R2 =", format(summary(regline)$r.squared, digits=2)),
bty="n")
legend(x=140, y=50, legend= noquote(paste("RMSEP=", (round(rmsep$val[1,1,2], digits=2)),
"bu/ac")), bty="n")
legend(x=175, y=20, legend = c("regression", "1:1"),pch= "_",
    col = c("black", "firebrick3"), pt.cex = 2,bty="n" )

################################

## Below are within year and between year models

###############################################################################
##############
## 2011 ###
wc11.duplex<-duplex(X=wc11$sg1,k=70,pc=10)

plot(wc11.duplex$pc)
points(wc11.duplex$pc[wc11.duplex$model, 1],wc11.duplex$pc[wc11.duplex$model, 2], col
= "red", pch = 19)
points(wc11.duplex$pc[wc11.duplex$test, 1], wc11.duplex$pc[wc11.duplex$test, 2], col =
"blue", pch = 19)
legend("topright", legend = c("calibration", "validation"), pch = c(19, 19),
    col = c("red", "blue"))

testrows11<-wc11.duplex$test

testrows11<-sort.int(testrows11)

wc11Test <- wc11[(testrows11),]
wc11Train <-wc11[-testrows11,]
###############################################################################
########

wc11Test.pedigree<-wc11[wc11$pedigree %in% wc11Test$pedigree, ]

levels(droplevels(wc11Test.pedigree$pedigree))

wc11Train.pedigree<-wc11[!wc11$pedigree %in% wc11Test$pedigree, ]

levels(droplevels(wc11Train.pedigree$pedigree))
#213 pedigrees in training set
```

```
###############################################################################
#################
wc11Test<-wc11Test.pedigree
wc11Train<-wc11Train.pedigree
###############################################################################
########
wc11.pls <- plsr(yield.buac~sg1, data= wc11Train, ncomp=15, validation="CV")
#PLSR 3

summary(wc11.pls)
plot(wc11.pls, ncomp = 15, asp = 1, line = TRUE)

RMSEP(wc11.pls, newdata=wc11Test)

plot(RMSEP(wc11.pls), legendpos = "topright")

plot(RMSEP(wc11.pls, newdata=wc11Test), legendpos = "topright")

######################
#plot
regline<- lm((predict(wc11.pls, ncomp=12, newdata=wc11Test))~ wc11Test$yield.buac)

rmsep<-RMSEP(wc11.pls, newdata=wc11Test, ncomp=12)

predplot(wc11.pls, ncomp = 12, newdata = wc11Test, line = TRUE,
      pch = 16, main = "Prediction Ability on 2011 Data Alone",
      xlab = expression(paste("True Yield ", "(bu ac"^"-1" ,")" )),
      ylab = expression(paste("Predicted Yield ", "(bu ac"^"-1" ,")" )),
      las=1, cex.axis=1.5)+
  abline(a=0,b=1,col="firebrick3",lwd=2)
abline(regline, lwd=2)
legend(x=100, y=45, legend= paste("R2 =", format(summary(regline)$r.squared, digits=4)),
bty="n")
legend(x=100, y=35, legend= noquote(paste("RMSEP=", (round(rmsep$val[1,1,2], digits=2)),
"bu/ac")), bty="n")
legend(x=100, y=20, legend = c("regression", "m=1"),pch= "_",
    col = c("black", "firebrick3"), pt.cex = 2,bty="n" )


###############################################################################
########
## 2012 ###

wc12.duplex<-duplex(X=wc12$sg1,k=100,pc=10)
```

```
plot(wc12.duplex$pc)
points(wc12.duplex$pc[wc12.duplex$model, 1],wc12.duplex$pc[wc12.duplex$model, 2], col
= "red", pch = 19)
points(wc12.duplex$pc[wc12.duplex$test, 1], wc12.duplex$pc[wc12.duplex$test, 2], col =
"blue", pch = 19)
legend("topright", legend = c("calibration", "validation"), pch = c(19, 19),
    col = c("red", "blue"))

testrows12<-wc12.duplex$test

testrows12<-sort.int(testrows12)

wc12Test <- wc12[(testrows12),]
wc12Train <-wc12[-testrows12,]
##############################################################################
#########
wc12Test.pedigree<-wc12[wc12$pedigree %in% wc12Test$pedigree, ]

levels(droplevels(wc12Test.pedigree$pedigree))

wc12Train.pedigree<-wc12[!wc12$pedigree %in% wc12Test$pedigree, ]

levels(droplevels(wc12Train.pedigree$pedigree))
#258 pedigrees in training set

##############################################################################
#################
wc12Test<-wc12Test.pedigree
wc12Train<-wc12Train.pedigree
##############################################################################
#########
wc12.pls <- plsr(yield.buac~sg1, data= wc12Train, ncomp=15, validation="CV")
# PLSR4

summary(wc12.pls)
plot(wc12.pls, ncomp = 15, asp = 1, line = TRUE)

RMSEP(wc12.pls, newdata=wc12Test)

plot(RMSEP(wc12.pls), legendpos = "topright")

plot(RMSEP(wc12.pls, newdata=wc12Test), legendpos = "topright")
```

```
######### PLOT

rmsep<-RMSEP(wc12.pls, newdata=wc12Test, ncomp=15)
regline<- lm((predict(wc12.pls, ncomp=15, newdata=wc12Test))~ wc12Test$yield.buac)


predplot(wc12.pls, ncomp = 15, newdata = wc12Test, line = TRUE,
      pch = 16, main = "Prediction Ability on 2012 Data Alone",
      xlab = expression(paste("True Yield ", "(bu ac"^"-1" ,")" )),
      ylab = expression(paste("Predicted Yield ", "(bu ac"^"-1" ,")" )),
      las=1)+
  abline(a=0,b=1,col="firebrick3",lwd=2)
abline(regline, lwd=2)
legend(x=150, y=80, legend= paste("R2 =", format(summary(regline)$r.squared, digits=4)),
bty="n")
legend(x=150, y=65, legend= noquote(paste("RMSEP=", (round(rmsep$val[1,1,2], digits=2)),
"bu/ac")), bty="n")
legend(x=150, y=50, legend = c("regression", "m=1"),pch= "_",
      col = c("black", "firebrick3"), pt.cex = 2,bty="n" )

#############################################################################
######################

###########################################

wc12TestSet.pedigree<-wc12[!wc12$pedigree %in% wc11$pedigree, ]

levels(droplevels(wc12TestSet.pedigree$pedigree))
#69

levels(droplevels(wc11$pedigree))
#275

wcTestSet<-wc12TestSet.pedigree
###################################################
wc12.pls <- plsr(yield.buac~sg1, data= wc11, ncomp=25, validation="CV")
#PLSR 7

summary(wc12.pls)
plot(wc12.pls, ncomp = 15, asp = 1, line = TRUE)

RMSEP(wc12.pls, newdata=wcTestSet)
```

```
plot(RMSEP(wc12.pls), legendpos = "topright")

plot(RMSEP(wc12.pls, newdata=wcTestSet), legendpos = "topright")

######### PLOT

rmsep<-RMSEP(wc12.pls, newdata=wcTestSet, ncomp=10)
regline<- lm((predict(wc12.pls, ncomp=10, newdata=wcTestSet))~ wcTestSet$yield.buac)


predplot(wc12.pls, ncomp = 10, newdata = wcTestSet, line = TRUE,
      pch = 16, main = "2011 Prediction Ability on 2012 Data - Different Pedigrees",
      xlab = expression(paste("True Yield ", "(bu ac"^"-1" ,")" )),
      ylab = expression(paste("Predicted Yield ", "(bu ac"^"-1" ,")" )),
      las=1)+
  abline(a=0,b=1,col="firebrick3",lwd=2)
abline(regline, lwd=2)
legend(x=150, y=80, legend= paste("R2 =", format(summary(regline)$r.squared, digits=4)),
bty="n")
legend(x=150, y=65, legend= noquote(paste("RMSEP=", (round(rmsep$val[1,1,2], digits=2)),
"bu/ac")), bty="n")
legend(x=150, y=50, legend = c("regression", "m=1"),pch= "_",
    col = c("black", "firebrick3"), pt.cex = 2,bty="n" )

####################################################
####################
## 2011 to predict 2012 ##

wc11.pls <- plsr(yield.buac~sg1, data= wc11, ncomp=25, validation="CV")
#PLSR5

summary(wc11.pls)
plot(wc11.pls, ncomp = 9, asp = 1, line = TRUE)

RMSEP(wc11.pls, newdata=wc12)

plot(RMSEP(wc11.pls), legendpos = "topright")

plot(RMSEP(wc11.pls, newdata=wc12), legendpos = "topright")


regline<- lm((predict(wc11.pls, ncomp=9, newdata=wc12))~ wc12$yield.buac)
```

```
predplot(wc11.pls, ncomp = 9, newdata = wc12, line = TRUE,
      pch = 16, main = "2011 Prediction Ability on 2012 Data",
      xlab = expression(paste("True Yield ", "(bu ac"^"-1" ,")" )),
      ylab = expression(paste("Predicted Yield ", "(bu ac"^"-1" ,")" )),
      las=1)+
  abline(a=0,b=1,col="firebrick3",lwd=2)
abline(regline, lwd=2)
legend(x=150, y=75, legend= paste("R2 =", format(summary(regline)$r.squared, digits=4)),
bty="n")
legend(x=150, y=55, legend = c("regression", "m=1"),pch= "_",
      col = c("black", "firebrick3"), pt.cex = 2,bty="n" )

#####################################################################
#####################
## 2012 to predict 2011 ##

wc12.pls <- plsr(yield.buac~sg1, data= wc12, ncomp=15, validation="CV")
#PLSR 8

summary(wc12.pls)
plot(wc12.pls, ncomp = 15, asp = 1, line = TRUE)

RMSEP(wc12.pls, newdata=wc11)

plot(RMSEP(wc12.pls), legendpos = "topright")

plot(RMSEP(wc12.pls, newdata=wc11), legendpos = "topright")

regline<- lm((predict(wc12.pls, ncomp=11, newdata=wc11))~ wc11$yield.buac)

predplot(wc12.pls, ncomp = 11, newdata = wc11, line = TRUE,
      pch = 16, main = "2012 Prediction Ability on 2011 Data",
      xlab = expression(paste("True Yield ", "(bu ac"^"-1" ,")" )),
      ylab = expression(paste("Predicted Yield ", "(bu ac"^"-1" ,")" )),
      las=1)+
  abline(a=0,b=1,col="firebrick3",lwd=2)
abline(regline, lwd=2)
legend(x=100, y=80, legend= paste("R2 =", format(summary(regline)$r.squared, digits=4)),
bty="n")
legend(x=100, y=50, legend = c("regression", "m=1"),pch= "_",
      col = c("black", "firebrick3"), pt.cex = 2,bty="n" )
#####################################################################
```

97

Building PLSR Model 2 and LM 1-4
```
## To convert yield from BU/AC to G/M^2, multiply yield by (6.271)

setwd("C:/Users/holly.lane/Project/Manuscript Files") ## Set this for your system
library(reshape); library(reshape2);library(ggplot2); library(dplyr)
library(prospectr);library(RcppArmadillo);library(Rcpp); library(foreach) ; library(iterators)
library(resemble);require(prospectr);library(ggfortify);library(cluster); library(pls)
#############################################################################
#############
wholecorn<- read.csv("Supplemental 1.csv", header=TRUE, check.names = FALSE)
#check.names prevents R from placing an X in front of the wavelength column names
dim(wholecorn)
##forcing R to accept numeric column names makes the column names in valid
##extract spectra columns before fixing this
wholecornspc <- wholecorn[,83:3194]
wholespc <- data.matrix(wholecornspc)
##now fix column names
valid_column_names <- make.names(names=names(wholecorn), unique=TRUE, allow_ =
TRUE)
names(wholecorn) <- valid_column_names
#select informational columns of interest, then attach matrix of spectra as variable
wholecornfull<- select(wholecorn,c(5,7,8,9,21,22,23,25,32,43,48:61,77,78))
wholecornfull$year<-as.factor(wholecornfull$year)
wholecornfull$spc<-wholecornspc
###in example dataset, NIRsoil, spectra is variable spc as a matrix within "dataframe"
wholecornfull$raw<-wholespc
#############################################################################
############

whole.sg <- savitzkyGolay(wholecornfull$spc, p = 3, w = 37, m = 1)
whole.sg2 <- savitzkyGolay(wholecornfull$spc, p = 3, w = 37, m = 2) ## adjust w
wholecornfull$sg1<-whole.sg
wholecornfull$sg2<-whole.sg2

wcFULL<-wholecornfull

WCcomp<- na.omit(wcFULL)
rownames(WCcomp) <- seq(length=nrow(WCcomp))

################
wc.pls <- plsr(yield.buac~sg1, data= WCcomp, ncomp=15, validation="CV")
summary(wc.pls)
plot(wc.pls, ncomp = 15, asp = 1, line = TRUE)
```

```
plot(RMSEP(wc.pls), legendpos = "topright")

plot(wc.pls, plottype = "scores", comps = 1:10)

plot(x= wc.pls$scores[,6], y= wc.pls$scores[,10])

#identify(x= wc.pls$scores[,6], y= wc.pls$scores[,10], plot=TRUE)

WCcomp2<- WCcomp[-c(1438, 1560),]

WCcomp<-WCcomp2
rownames(WCcomp) <- seq(length=nrow(WCcomp))

################################################################
WCcomp.duplex<-duplex(X=WCcomp$sg1,k=100,pc=10) # takes awhile to run

plot(WCcomp.duplex$pc)
points(WCcomp.duplex$pc[WCcomp.duplex$model,
1],WCcomp.duplex$pc[WCcomp.duplex$model, 2], col = "red", pch = 19)
points(WCcomp.duplex$pc[WCcomp.duplex$test, 1],
WCcomp.duplex$pc[WCcomp.duplex$test, 2], col = "blue", pch = 19)
legend("topright", legend = c("calibration", "validation"), pch = c(19, 19),
     col = c("red", "blue"))
##result is list with row index for calibration(model)/and validation(test) samples

testrows<-WCcomp.duplex$test
testrows<-sort.int(testrows)

WCcompTest <- WCcomp[(testrows),]

WCcompTrain <-WCcomp[-testrows,]

levels(droplevels(WCcomp$pedigree))
# 345 represented -- NOTE: Some may be missing from yield adjustments

levels(droplevels(WCcompTest$pedigree))
# 82 represented, need to eliminate all from training set

levels(droplevels(WCcompTrain$pedigree))
# 345 represented
#########################
```

```
WCcompTest.pedigree<-WCcomp[WCcomp$pedigree %in% WCcompTest$pedigree, ]

levels(droplevels(WCcompTest.pedigree$pedigree))
dim(WCcompTest)
dim(WCcompTest.pedigree)
WCcompTest.pedigree[1:10,1:5]

WCcompTrain.pedigree<-WCcomp[!WCcomp$pedigree %in% WCcompTest$pedigree, ]

dim(WCcompTrain)
dim(WCcompTrain.pedigree)
WCcompTrain.pedigree[1:10,1:5]

levels(droplevels(WCcompTrain.pedigree$pedigree))
#263 pedigrees in training set

###############################################################################
#################
WCcompTest<-WCcompTest.pedigree
WCcompTrain<-WCcompTrain.pedigree
###############################################################################
################
whole.pls.ya <- plsr(yield.buac~sg1, data= WCcompTrain, ncomp=25, validation="CV")
# PLSR 2 built

summary(whole.pls.ya)
plot(whole.pls.ya, ncomp = 15, asp = 1, line = TRUE)
####
#modelresults<-sort.default(whole.pls.ya$validation$pred[,,15])
##
#RMSEP(whole.pls.ya, newdata=WCcompTest)

plot(RMSEP(whole.pls.ya), legendpos = "topright")


ncomp.onesigma <- selectNcomp(whole.pls.ya, method = "onesigma", plot = TRUE)
ncomp.permut <- selectNcomp(whole.pls.ya, method = "randomization", plot = TRUE)

plot(RMSEP(whole.pls.ya, newdata=WCcompTest), legendpos = "topright")

##### PLOT
rmsep<-RMSEP(whole.pls.ya, newdata=WCcompTest, ncomp=13)
```

```
regline<- lm((predict(whole.pls.ya, ncomp=13, newdata=WCcompTest))~
WCcompTest$yield.buac)

predplot(whole.pls.ya, ncomp = 13, newdata = WCcompTest, line = TRUE,
      pch = 16, main = "Global Prediction Ability (NIRS)",
      xlab = expression(paste("True Yield ", "(bu ac"^"-1" ,")" )),
      ylab = expression(paste("Predicted Yield ", "(bu ac"^"-1" ,")" )),
      las=1)+
  abline(a=0,b=1,col="firebrick3",lwd=2)
abline(regline, lwd=2)
legend(x=100, y=75, legend= paste("R2 =", format(summary(regline)$r.squared, digits=2)),
bty="n")
legend(x=100, y=65, legend= noquote(paste("RMSEP=", (round(rmsep$val[1,1,2], digits=2)),
"bu/ac")), bty="n")
legend(x=100, y=30, legend = c("regression", "1:1"),pch= "_",
      col = c("black", "firebrick3"), pt.cex = 2,bty="n" )


###############################################################################
################
#protein
cor.test(WCcomp$yield.buac, WCcomp$crude.protein, method="pearson")

pp <- lm(yield.buac ~ crude.protein, data= WCcompTrain)

predictions<-predict(pp, newdata=WCcompTest)

regline <- lm(predictions ~ WCcompTest$yield.buac)

plot(WCcompTest$yield.buac, predictions)
abline(regline, lwd=2)
legend(x=180, y=75, legend= paste("R2 =", format(summary(regline)$r.squared, digits=2)),
bty="n")
legend(x=180, y=65, legend= paste("R =", format(sqrt(summary(regline)$r.squared),
digits=2)), bty="n")

## need to calculate RMSEP by hand....

WCcompTest$diff<-(predictions) - (WCcompTest$yield.buac)

WCcompTest$diff.sq<-(WCcompTest$diff)*(WCcompTest$diff)

diff.sum<-sum(WCcompTest$diff.sq)
```

```
MSEP<-(diff.sum)/(nrow(WCcompTest))

RMSEP <- sqrt(MSEP)
RMSEP #in bu/ac
##############################################################################
################
#starch.2
cor.test(WCcomp$yield.buac, WCcomp$starch.2, method="pearson")

sp <- lm(yield.buac ~ starch.2, data= WCcompTrain)

predictions<-predict(sp, newdata=WCcompTest)

regline <- lm(predictions ~ WCcompTest$yield.buac)

plot(WCcompTest$yield.buac, predictions)
abline(regline, lwd=2)
legend(x=180, y=75, legend= paste("R2 =", format(summary(regline)$r.squared, digits=2)),
bty="n")
legend(x=180, y=65, legend= paste("R =", format(sqrt(summary(regline)$r.squared),
digits=2)), bty="n")

WCcompTest$diff<-(predictions) - (WCcompTest$yield.buac)

WCcompTest$diff.sq<-(WCcompTest$diff)*(WCcompTest$diff)

diff.sum<-sum(WCcompTest$diff.sq)

MSEP<-(diff.sum)/(nrow(WCcompTest))

RMSEP <- sqrt(MSEP)
RMSEP #in bu/ac
##############################################################################
################
#fat.3
cor.test(WCcomp$yield.buac, WCcomp$fat.3, method="pearson")

fp <- lm(yield.buac ~ fat.3, data= WCcompTrain)

predictions<-predict(fp, newdata=WCcompTest)

regline <- lm(predictions ~ WCcompTest$yield.buac)
```

```r
plot(WCcompTest$yield.buac, predictions)
abline(regline, lwd=2)
legend(x=180, y=90, legend= paste("R2 =", format(summary(regline)$r.squared, digits=2)),
bty="n")
legend(x=180, y=92, legend= paste("R =", format(sqrt(summary(regline)$r.squared),
digits=2)), bty="n")

WCcompTest$diff<-(predictions) - (WCcompTest$yield.buac)

WCcompTest$diff.sq<-(WCcompTest$diff)*(WCcompTest$diff)

diff.sum<-sum(WCcompTest$diff.sq)

MSEP<-(diff.sum)/(nrow(WCcompTest))

RMSEP <- sqrt(MSEP)
RMSEP #in bu/ac
###############################################################################
#################
#all
ap <- lm(yield.buac ~ fat.3+starch.2+crude.protein, data= WCcompTrain)

predictions<-predict(ap, newdata=WCcompTest)

regline <- lm(predictions ~ WCcompTest$yield.buac)


plot(WCcompTest$yield.buac, predictions)
abline(regline, lwd=2)
legend(x=180, y=90, legend= paste("R2 =", format(summary(regline)$r.squared, digits=2)),
bty="n")
legend(x=180, y=85, legend= paste("R =", format(sqrt(summary(regline)$r.squared),
digits=2)), bty="n")

WCcompTest$diff<-(predictions) - (WCcompTest$yield.buac)

WCcompTest$diff.sq<-(WCcompTest$diff)*(WCcompTest$diff)

diff.sum<-sum(WCcompTest$diff.sq)

MSEP<-(diff.sum)/(nrow(WCcompTest))

RMSEP <- sqrt(MSEP); RMSEP #in bu/ac
```

Running PLSR under CV0 Scheme:

```
setwd("C:/Users/holly.lane/Project/Manuscript Files")
library(reshape); library(reshape2);library(ggplot2); library(dplyr)
library(prospectr);library(RcppArmadillo);library(Rcpp); library(foreach) ; library(iterators)
library(resemble);require(prospectr);library(ggfortify);library(cluster); library(pls)
############################################################################
###
## Make sure you have run the other script that cleans and creates the wc object first
############################################################################
###

wc11 <- wc[(wc$year=="2011"),]
wc12 <-wc[(wc$year=="2012"),]
rownames(wc11) <- seq(length=nrow(wc11))
rownames(wc12) <- seq(length=nrow(wc12))

WW<-wc[(wc$treatment =="IRRI"),]
WS<-wc[(wc$treatment =="NIRRI"),]

WW11<-wc[(wc$trial.1 =="CS11_WW"),]
WW12<-wc[(wc$trial.1 =="CS12_WW"),]

WS11<-wc[(wc$trial.1 =="CS11_WS"),]
WS12<-wc[(wc$trial.1 =="CS12_WS"),]

rownames(WW) <- seq(length=nrow(WW))
rownames(WS) <- seq(length=nrow(WS))
rownames(WW11) <- seq(length=nrow(WW11))
rownames(WS11) <- seq(length=nrow(WS11))
rownames(WW12) <- seq(length=nrow(WW12))
rownames(WS12) <- seq(length=nrow(WS12))

############################################################################
################
wcTest<-WS12 ##Update and change plot title for each treatment
wcTrain<-WW
wcTrain<-wc[!(wc$trial.1 =="CS12_WS"),] #Update and change plot title

############################################################################
################
whole.pls.ya <- plsr(yield.buac~sg1, data= wcTrain, ncomp=50, validation="CV")
#summary(whole.pls.ya)
```

```
#plot(whole.pls.ya, ncomp = 25, asp = 1, line = TRUE)
####
#modelresults<-sort.default(whole.pls.ya$validation$pred[,,15])
##
#RMSEP(whole.pls.ya, newdata=wcTest)
plot(RMSEP(whole.pls.ya), legendpos = "topright")

ncomp.onesigma <- selectNcomp(whole.pls.ya, method = "onesigma", plot = TRUE)
ncomp.permut <- selectNcomp(whole.pls.ya, method = "randomization", plot = TRUE)

plot(RMSEP(whole.pls.ya, newdata=wcTest), legendpos = "topright")

##### PLOT
rmsep<-RMSEP(whole.pls.ya, newdata=wcTest, ncomp=12)
R2(whole.pls.ya, newdata=wcTest, ncomp=12)

regline<- lm((predict(whole.pls.ya, ncomp=12, newdata=wcTest))~ wcTest$yield.buac)

predplot(whole.pls.ya, ncomp = 12, newdata = wcTest, line = TRUE,
      cex.axis = 1.5, cex.lab = 1.5,cex.main=2, #for poster
      pch = 16, main = "Fix",
      xlab = expression(paste("True Yield ", "(bu ac"^"-1" ,")" )),
      ylab = expression(paste("Predicted Yield ", "(bu ac"^"-1" ,")" )),
      las=1)+
  abline(a=0,b=1,col="firebrick3",lwd=2)
abline(regline, lwd=2)
legend(x=90, y=80, legend= paste("R2 =", format(summary(regline)$r.squared, digits=2)),
bty="n")
legend(x=90, y=70, legend= noquote(paste("RMSEP=", (round(rmsep$val[1,1,2], digits=2)),
"bu/ac")), bty="n")
legend(x=95, y=50, legend = c("regression", "1:1"),pch= "_",
    col = c("black", "firebrick3"), pt.cex = 2,bty="n" )

summary(regline)$r.squared
```

```
getwd()
setwd("C:/Users/holly.lane/Project 2")
library(lme4); #library(MuMIn);
library(dplyr); library(ggplot2)

iterations<-1:50

results <- matrix(0, 100000,25)
results <- as.data.frame(results)
colnames(results) <-
c("Scenario","Population","Genotypes","Environments","Reps","n","measurement",
          "Percent.Detection.QTL", "False.QTL","Percent.False.QTL", "r2.QTL",
          "r2", "RMSE","NRMSE", "H2", "var.gxe", "var.gen", "var.rep", "var.env",
"var.residual",
          "per.var.gxe", "per.var.gen", "per.var.rep", "per.var.env", "per.var.residual")
row<-0 #leave outside of loop


gen.test<-c(100,300,500,1000)#numbers of genotypes to test
env.test<-c(1,2,5,10) #numbers of environments to test
rep.test<-c(2,3) #numbers of reps to test #numbers of reps to test
measurement<-c("manual", "HTP.50", "HTP.80", "HTP.90") # whether or not error is added,
how much


for (ITER in iterations){
  print(c("Iteration", ITER))

individuals<- 300 #number of indviduals to start at 'F2'
SNPs<-100

## What about for allele frequencies that vary at each SNP? (This would not apply to a RIL
population)
F2<-matrix(data = NA, nrow=individuals, SNPs)

AlleleFreq<-round(abs(runif(SNPs, min = 0.4, max = 0.6)), digits=2)
#want to be close to 0.5, but won't be exact realistically

AlleleFreq # frequency of each allele
```

```
for (i in 1:ncol(F2)) {
  F2[,i]<-rbinom(nrow(F2),2,AlleleFreq[i])
}

F2 #SNP matrix for F2 parent population
##################################################
# defining phenotypic values
baseline.ht <- 0 #cm
# G2F hybrid data from 2017 shows mean height of 213 cm (range 71-345)

AlleleEffect.ht <- rgeom(SNPs, 0.3) # allele effect following geometric distribution

sum(AlleleEffect.ht==0) # how many alleles have no effect
#hist(AlleleEffect.ht)

ht.effect<-t(t(F2) * AlleleEffect.ht)

ht <- rowSums(ht.effect)+baseline.ht
ht

### adding dominance effect

##AlleleEffect.ht.dom <- c(0,2,4,0,0,1,3,0,0,2) ##Figure out how to softcode

AlleleEffect.ht.dom <- rpois(SNPs, 0.5)

dom.effect.ht <-matrix (0,nrow(F2),ncol(F2))

for (SNP in 1:ncol(F2)) {
  for (IND in 1:nrow(F2)) {
    print(c("loop SNP",SNP, "IND", IND,  AlleleEffect.ht.dom[SNP]))
    if(F2[IND,SNP] ==1) {
      dom.effect.ht[IND,SNP]<-(AlleleEffect.ht.dom[SNP] * ht.effect[IND,SNP])}
  }}

dom.effect.ht
F2

ht <- rowSums(ht.effect)+baseline.ht+rowSums(dom.effect.ht)
ht # "true" genetic height of the F2 population
###############################################################################
##########
```

```
#
#
###
####

##############################################################################
###########################
## Now let's self the population - write to expand or contract the population
progeny <- 1000 ## make it as big as possible and then subset
individuals<-progeny #must both be the same length

# defining phenotypic values
environments <- 10
reps<- 3
h2 <- 0.5  # ratio of genetic to residual variance
##################################################################
####
##
##
#
#build progeny matrix
gen3parents<-matrix(NA, nrow= progeny, ncol=ncol(F2))
#choose parents to advance
#in this case, we do not need to select parents from a F1, so this code is a bit overkill

#randomly select and write selected parents to an object (sample function, with replacement)
parents<-sample(1:nrow(F2), progeny, replace = TRUE)

gen3parents<- F2[parents, byrow= TRUE]

gen3<-matrix(NA, nrow= nrow(gen3parents), ncol=ncol(F2))

count<-0
for (PAR in 1:nrow(gen3parents)){
  for (SNP in 1:ncol(gen3parents)){
    count<- count+1
    print(c("genotype",PAR,  "SNP",SNP, count))
    gen3[PAR,SNP]<- ifelse(gen3parents[PAR,SNP]!= 1, (gen3parents[PAR,SNP]),
(rbinom(1,2,0.5)))
  }}

F3<-gen3
```

```
gen4<-matrix(NA, nrow= nrow(gen3), ncol=ncol(gen3))

count<-0
for (PAR in 1:nrow(gen3)){
 for (SNP in 1:ncol(gen3)){
   count<- count+1
   print(c("genotype",PAR,  "SNP",SNP, count))
   gen4[PAR,SNP]<- ifelse(gen3[PAR,SNP]!= 1, (gen3[PAR,SNP]), (rbinom(1,2,0.5)))
 }}


#rapidly advance to RIL

RIL<-matrix(NA, nrow= nrow(gen4), ncol=ncol(gen4))
count<-0
for (PAR in 1:nrow(gen4)){
 for (SNP in 1:ncol(gen4)){
   count<- count+1
   print(c("genotype",PAR,  "SNP",SNP, count))
   RIL[PAR,SNP]<- ifelse(gen4[PAR,SNP]!= 1, (gen4[PAR,SNP]), (rbinom(1,1,0.5)*2))

 }}
RIL[1:10,1:10] #RIL SNP matrix
################################################################

RIL.ht.effect<-t(t(RIL) * AlleleEffect.ht)

RIL.ht <- rowSums(RIL.ht.effect)+baseline.ht
#RIL.ht # "true" genetic height of each genotype

### dominance effect (shouldn't be in RILs)
########################################

F3.ht.effect<-t(t(F3) * AlleleEffect.ht)

F3.ht <- rowSums(F3.ht.effect)+baseline.ht

F3.dom.effect.ht <-matrix (0,nrow(F3),ncol(F3))

for (SNP in 1:ncol(F3)) {
 for (IND in 1:nrow(F3)) {
   print(c("loop SNP",SNP, "IND", IND,  AlleleEffect.ht.dom[SNP]))
   if(F3[IND,SNP] ==1) {
```

109

```
     F3.dom.effect.ht[IND,SNP]<-(AlleleEffect.ht.dom[SNP] * F3.ht.effect[IND,SNP])}
  }}
rowSums(F3.dom.effect.ht)
# F3

F3.ht <- rowSums(F3.ht.effect)+baseline.ht+rowSums(F3.dom.effect.ht)
head(F3.ht) #now that dominance is applied

##############################################################################
##########
## GxE  is an unpredictable effect of the environment on different genotypes
# randomly, normally distributed around 0

## ENVIRONMENT effect, number set above

enveffect<-round(rnorm(environments, mean=0, sd=15), digits= 0)
enveffect

enviroEffects<-matrix(data= enveffect, length(RIL.ht), environments, byrow= TRUE)

head(enviroEffects) #matrix to apply to achieve RIL phenotypes observed

## GxE effect

GxE.effect<-abs(round(rnorm(environments, mean=5, sd=10), digits= 0)) ##CANNOT BE
NEGATIVE
## this effect is used as the std. dev in the script below, so take abs

GxE.effects<-matrix(data= NA, length(RIL.ht), environments)

for (ENV in 1:ncol(GxE.effects)) {
  for (IND in 1:nrow(GxE.effects)) {
    print(c("loop ENV",ENV, "IND", IND,  GxE.effect[ENV]))
    GxE.effects[IND,ENV] <- rnorm(1,0,GxE.effect[ENV])
  }}

#hist(GxE.effects) #may be too small, investigate % variance

## REPLICATE effect ###########################

# number of reps set above

repEffect<-(environments*reps)
```

```
repEffects<-matrix(rnorm(repEffect,0,5),reps,environments)
#repEffects

RIL.obs<-length(enviroEffects)*reps
RIL.obsmatrix <- matrix(,RIL.obs,4)
#RIL.obsmatrix

count<-0
for (ENV in 1:ncol(GxE.effects)) {
  for (IND in 1:nrow(GxE.effects)) {
    for (REP in 1:reps) {
      count<- count +1

      print(c("Environment",ENV, "Individual", IND,  "count",count, "rep", REP, count))
      RIL.obsmatrix[count,1] <- IND
      RIL.obsmatrix[count,2] <- ENV
      RIL.obsmatrix[count,3] <- REP
      RIL.obsmatrix[count,4] <- RIL.ht[IND]+enviroEffects[IND, ENV]
+repEffects[REP,ENV]+ GxE.effects[IND,ENV]
    }}}
head(RIL.obsmatrix) #col 1 = individual, col 2 = environment, col 3 = rep, col 4 = height
###

F3.obs<-length(enviroEffects)*reps
F3.obsmatrix <- matrix(,F3.obs,4)
#F3.obsmatrix

count<-0
for (ENV in 1:ncol(GxE.effects)) {
  for (IND in 1:nrow(GxE.effects)) {
    for (REP in 1:reps) {
      count<- count +1

      print(c("Environment",ENV, "Individual", IND,  "count",count, "rep", REP, count))
      F3.obsmatrix[count,1] <- IND
      F3.obsmatrix[count,2] <- ENV
      F3.obsmatrix[count,3] <- REP
      F3.obsmatrix[count,4] <- F3.ht[IND]+enviroEffects[IND, ENV] +repEffects[REP,ENV]+
GxE.effects[IND,ENV]
    }}}
head(F3.obsmatrix)
##### ERROR ########################################################
```

error is independent for each observation (essentially RepxGxE) (unexplained variation)

following code from Dr. Zhang of Washington State University
# h2 set above
n <- reps*environments*progeny

effectvar <- var(RIL.ht)
#effectvar

residualvar=(effectvar-h2*effectvar)/h2
residualvar
residual=rnorm(n,0,sqrt(residualvar))
# end of adapted script
#hist(residual)
mean(abs(residual))

error<-residual

RIL.obsmatrixerr<-RIL.obsmatrix

RIL.obsmatrixerr[,4]<-RIL.obsmatrix[,4]+error

head(RIL.obsmatrixerr) #col 1 = individual, col 2 = environment, col 3 = rep, col 4 = height (with error effect)

ERROR

error is independent for each observation (essentially RepxGxE) (unexplained variation)

following code from Dr. Zhang of Washington State University
# h2 set above
n <- reps*environments*progeny

effectvar <- var(F3.ht)
#effectvar

residualvar=(effectvar-h2*effectvar)/h2
residualvar
residual=rnorm(n,0,sqrt(residualvar))
# end of adapted script
#hist(residual)
mean(abs(residual))

error<-residual

F3.obsmatrixerr<-F3.obsmatrix

F3.obsmatrixerr[,4]<-F3.obsmatrix[,4]+error

head(F3.obsmatrixerr) #col 1 = individual, col 2 = environment, col 3 = rep, col 4 = height
(with error effect)


```
#####################
#ObservMatrix <-RIL.obsmatrixerr; P<- RIL.ht
ObservMatrix <-F3.obsmatrixerr ; P<- F3.ht

Data<-data.frame(ObservMatrix)
names(Data) <- c("Gen","Env","Rep","Hgt") # variable names

Data$Gen<-factor(Data$Gen)
Data$Env<-factor(Data$Env)
Data$Rep<-factor(Data$Rep)
################################

#ObservMatrix <- ObservMatrix[-which((ObservMatrix[,2]=="X")),] # remove environment
X


## need to do multiple trials of each level (repeated measures) to gain an average
#Loops<-c(1:25) #number of times to replicat each scenario

#for(LOOP in Loops){print(c("Loop",LOOP)) #

  for (GENOTYPE in gen.test){
    genotypes<- sort(sample(progeny, GENOTYPE, replace = FALSE))

    index<-rep(F,nrow(Data))

    for(j in 1:length(genotypes)){

      for(k in 1:nrow(Data)){

        if(index[k]==T){
          next
```

```
    }
    if(as.character(Data$Gen[k])==as.character(genotypes[j])){
     index[k]<-T
    }

  } #### close k loop

}#### Close j loop

Data.sub<-Data[index,]


for (ENVIRONMENT in env.test){

  environments.index<- sort(sample(environments, ENVIRONMENT, replace = FALSE))

  index<-rep(F,nrow(Data.sub))

  for(j in 1:length(environments.index)){

    for(k in 1:nrow(Data.sub)){

     if(index[k]==T){
      next
     }
     if(as.character(Data.sub$Env[k])==as.character(environments.index[j])){
      index[k]<-T
     }

    } #### close k loop

  }#### Close j loop

  Data.sub2<-Data.sub[index,]


  for (REP in rep.test){

   reps.index<- sort(sample(reps, REP, replace = FALSE))

   #print((c(length(environments.index), "environments")))
   #print((c(length(reps.index), "reps")))
```

```
    index<-rep(F,nrow(Data.sub2))

    for(j in 1:length(reps.index)){

      for(k in 1:nrow(Data.sub2)){

        if(index[k]==T){
          next
        }
        if(as.character(Data.sub2$Rep[k])==as.character(reps.index[j])){
          index[k]<-T
        }

      } #### close k loop

    }#### Close j loop
    Data.sub3<-Data.sub2[index,]


    for (m in measurement){
      Data.sub3<-Data.sub2[index,]
      print(paste(c(length(genotypes), "genotypes", length(environments.index),
"environments",length(reps.index), "reps", m )))

      # if (m=="HTP.20"){
      # n2 <- nrow(Data.sub3)
      # h22<-0.8 #adds 20% more error
      # effectvar2 <- var(RIL.ht[genotypes])
      # residualvar2=(effectvar2-h22*effectvar2)/h22
      # residualvar2
      # residual2=rnorm(n2,0,sqrt(residualvar2))
      # error<-residual2

      #  Data.sub3[,4]<-Data.sub3[,4]+error
      # }
      if (m=="HTP.50"){
        n2 <- nrow(Data.sub3)
        h22<-0.5 #adds 50% more error, same size as baseline
        effectvar2 <- var(P[genotypes])
        residualvar2=(effectvar2-h22*effectvar2)/h22
        residualvar2
        residual2=rnorm(n2,0,sqrt(residualvar2))
        error<-residual2
```

115

```r
        Data.sub3[,4]<-Data.sub3[,4]+error
        }

        if (m=="HTP.90"){
         n2 <- nrow(Data.sub3)
         h22<-0.1 #adds 90% more error relative to genetic effects
         effectvar2 <- var(P[genotypes])
         residualvar2=(effectvar2-h22*effectvar2)/h22
         residualvar2
         residual2=rnorm(n2,0,sqrt(residualvar2))
         error<-residual2

         Data.sub3[,4]<-Data.sub3[,4]+error
        }

        if (m=="HTP.80"){
         n2 <- nrow(Data.sub3)
         h22<-.2 #adds 80% more error relative to genetic effects
         effectvar2 <- var(P[genotypes])
         residualvar2=(effectvar2-h22*effectvar2)/h22
         residualvar2
         residual2=rnorm(n2,0,sqrt(residualvar2))
         error<-residual2

         Data.sub3[,4]<-Data.sub3[,4]+error
        }

        DataSub<-Data.sub3
        if (ENVIRONMENT!=1){
         fm1 <- lmer(Hgt ~ (1|Gen) + (1|Env) + (1|Gen:Env) + (1|Env:Rep), data = DataSub)
#all random
        }
        if (ENVIRONMENT==1){
         fm1 <- lmer(Hgt ~ (1|Gen) +  (1|Gen:Env) + (1|Env:Rep), data = DataSub) # if only 1
env
        }

        #fm1 <- lmer(Hgt ~ 1 + Gen +  (1|Gen:Env) + (1|Env:Rep), data = Data)
        #summary(fm1)

        varcomp<-as.data.frame(VarCorr(fm1))
```

```
var<-(select(varcomp, grp, vcov))

var[6,1]<-"Total"
var[6,2]<- sum(var$vcov[1:5])
var$pervar<-0
var$pervar[1:6]<-round((var$vcov[1:6]/var[6,2])*100,digits=3)
###
if (ENVIRONMENT!=1){
  fm1 <- lmer(Hgt ~ 1 + Gen + (1|Env) + (1|Gen:Env) + (1|Env:Rep), data = DataSub)
}
if (ENVIRONMENT==1){
  fm1 <- lmer(Hgt ~ 1 +Gen +  (1|Gen:Env) + (1|Env:Rep), data = DataSub) # if only 1
env
}
#gives estimated effect for each genotype, you could then select best genotypes to
advance

est<-as.data.frame(fixef(fm1))
est$estimated.gen<-0

est$estimated.gen[1]<- est[1,1] # first genotype is the intercept

est$estimated.gen[2:(nrow(est))]<-
(est$`fixef(fm1)`[2:nrow(est)])+est$estimated.gen[1]
#this adds the intercept to the effects to achieve estimated genotypic height

########## SINGLE QTL ANALYSIS
BLUP<-est$estimated.gen #give BLUP to data frame

estimates<- as.data.frame(cbind(F3[genotypes,],BLUP))

single.marker<- matrix(NA,SNPs,6)

single.marker <- as.data.frame(single.marker)
colnames(single.marker) <-
c("SNP","Effect.Present","Effect.Detected","Pvalue","Est.Homozygote","True.Effect")
round<-0

for (MARKER in 1:SNPs){
  QTL<- estimates[,MARKER]
  V<- colnames(estimates[MARKER])

  if(length(levels(as.factor(estimates[,MARKER])))==1){next} ## skip locus if fixed
```

```
        TEST<-(aov(BLUP~as.factor(QTL), data=estimates))
        MEANS<-model.tables(TEST,"means")
        EFFECTS<-model.tables(TEST,"effects")
        summary(TEST)
        round<-round+1
        if(na.omit(summary(TEST)[[1]][["Pr(>F)"]]) < 0.05){
          single.marker$Effect.Detected[round]<- "YES"
        }
        if(na.omit(summary(TEST)[[1]][["Pr(>F)"]]) > 0.05){
          single.marker$Effect.Detected[round]<- "NO"
        }
        if (AlleleEffect.ht[MARKER]==0){
          single.marker$Effect.Present[round]<-"NO"
        }
        if (AlleleEffect.ht[MARKER]!=0){
          single.marker$Effect.Present[round]<-"YES"
        }
        if(length(levels(as.factor(estimates[,MARKER])))==3){ #if heterozygotes exist
          single.marker$Est.Homozygote[round]<-
round((EFFECTS$tables$`as.factor(QTL)`[3]-EFFECTS$tables$`as.factor(QTL)`[1]),4)[[1]]
#homozygote
        }
        if(length(levels(as.factor(estimates[,MARKER])))==2){ #if inbred at locus
          single.marker$Est.Homozygote[round]<-
round((EFFECTS$tables$`as.factor(QTL)`[2]-EFFECTS$tables$`as.factor(QTL)`[1]),4)[[1]]
#homozygote
        }
        single.marker$True.Effect[round]<-AlleleEffect.ht[MARKER]*2 #real homozygous
effect
        single.marker$Pvalue[round]<-na.omit(summary(TEST)[[1]][["Pr(>F)"]]) #pvalue
column
        single.marker$SNP[round]<- MARKER
      }

      single.marker<-na.omit(single.marker)

      #from this results table we can then generate % of false positives, missed QTL,
      #and accuracy of estimations, this then allows us to compare scenarios
      #***********************************
```

```
    Percent.Detection<-
sum(single.marker$Effect.Present=="YES"&single.marker$Effect.Detected=="YES")/sum(si
ngle.marker$Effect.Present=="YES")

    False.QTL <-
sum(single.marker$Effect.Present=="NO"&single.marker$Effect.Detected=="YES")

    Percent.False <-
sum(single.marker$Effect.Present=="NO"&single.marker$Effect.Detected=="YES")/sum(sin
gle.marker$Effect.Detected=="YES")

    r.qtl<-lm((subset(single.marker, Effect.Detected %in% "YES" & Effect.Present %in%
"YES")$Est.Homozygote)~
          (subset(single.marker, Effect.Detected %in% "YES" & Effect.Present %in%
"YES")$True.Effect))

    r2.qtl<-round((summary(r.qtl)$r.squared), digits=2)
    #****************************************
    ### END SINGLE QTL

    est$true.gen<-P[genotypes]

    est$error<- (est$true.gen)-(est$estimated.gen)

    est$error2<- (est$error*est$error)

    RMSE <- sqrt((sum(est$error2))/(nrow(est)))

    #NRMSE<- (RMSE)/(max(DataSub[,4])-min(DataSub[,4]))#skewed due to outliers in
HTP error
    NRMSE<- (RMSE)/(max(P[genotypes])-min(P[genotypes]))

    reg<- lm(est$true.gen~est$estimated.gen)

    row <-row + 1 ; print(row)

    results$Scenario[row]<- row
    results$Population[row]<- "F3"
    results$Genotypes[row]<- GENOTYPE
    results$Environments[row]<-ENVIRONMENT
    results$Reps[row]<-REP
    results$n[row]<-nrow(DataSub)
    results$measurement[row]<-m
```

```r
      results$r2[row] <- round((summary(reg)$r.squared), digits=2)
      results$RMSE[row] <- RMSE
      results$NRMSE[row] <- NRMSE
      results$H2[row] <-
(var[2,2])/(((var[5,2])/REP*ENVIRONMENT)+((var[1,2])/ENVIRONMENT)+(var[2,2]))
      results$Percent.Detection.QTL[row]<-Percent.Detection
      results$False.QTL[row]<- False.QTL
      results$Percent.False.QTL[row]<-Percent.False
      results$r2.QTL[row]<-r2.qtl
      results$var.gxe[row]<-var[1,2]
      results$var.gen[row]<-var[2,2]
      results$var.rep[row]<-var[3,2]
      results$var.env[row]<-var[4,2]
      results$var.residual[row]<-var[5,2]
      results$per.var.gxe[row]<-var[1,3]
      results$per.var.gen[row]<-var[2,3]
      results$per.var.rep[row]<-var[3,3]
      results$per.var.env[row]<-var[4,3]
      results$per.var.residual[row]<-var[5,3]

    }#end of error loop
   }#end of rep loop
  }#end of env loop
 }#end of genotypes loop
#could we add another loop to repeat this multiple times and get an average?


#####################
ObservMatrix <-RIL.obsmatrixerr; P<- RIL.ht
#ObservMatrix <-F3.obsmatrixerr ; P<- F3.ht

Data<-data.frame(ObservMatrix)
names(Data) <- c("Gen","Env","Rep","Hgt") # variable names

Data$Gen<-factor(Data$Gen)
Data$Env<-factor(Data$Env)
Data$Rep<-factor(Data$Rep)
################################

### now loop for RIL population, do not reset 'row' as it will just continue onto existing
# results table

for (GENOTYPE in gen.test){
```

```r
genotypes<- sort(sample(progeny, GENOTYPE, replace = FALSE))

index<-rep(F,nrow(Data))

for(j in 1:length(genotypes)){

 for(k in 1:nrow(Data)){

  if(index[k]==T){
   next
  }
  if(as.character(Data$Gen[k])==as.character(genotypes[j])){
   index[k]<-T
  }

 } #### close k loop

}#### Close j loop

Data.sub<-Data[index,]


for (ENVIRONMENT in env.test){

 environments.index<- sort(sample(environments, ENVIRONMENT, replace = FALSE))

 index<-rep(F,nrow(Data.sub))

 for(j in 1:length(environments.index)){

  for(k in 1:nrow(Data.sub)){

   if(index[k]==T){
    next
   }
   if(as.character(Data.sub$Env[k])==as.character(environments.index[j])){
    index[k]<-T
   }

  } #### close k loop

 }#### Close j loop
```

```r
    Data.sub2<-Data.sub[index,]


  for (REP in rep.test){

    reps.index<- sort(sample(reps, REP, replace = FALSE))

    #print((c(length(environments.index), "environments")))
    #print((c(length(reps.index), "reps")))

    index<-rep(F,nrow(Data.sub2))

    for(j in 1:length(reps.index)){

      for(k in 1:nrow(Data.sub2)){

        if(index[k]==T){
          next
        }
        if(as.character(Data.sub2$Rep[k])==as.character(reps.index[j])){
          index[k]<-T
        }

      } #### close k loop

    }#### Close j loop
    Data.sub3<-Data.sub2[index,]


    for (m in measurement){
      Data.sub3<-Data.sub2[index,]
      print(paste(c(length(genotypes), "genotypes", length(environments.index),
"environments",length(reps.index), "reps", m )))


      if (m=="HTP.50"){
        n2 <- nrow(Data.sub3)
        h22<-0.5 #adds 50% more error, same size as baseline
        effectvar2 <- var(P[genotypes])
        residualvar2=(effectvar2-h22*effectvar2)/h22
        residualvar2
        residual2=rnorm(n2,0,sqrt(residualvar2))
        error<-residual2
```

```
          Data.sub3[,4]<-Data.sub3[,4]+error
        }

      if (m=="HTP.90"){
       n2 <- nrow(Data.sub3)
       h22<-0.1 #adds 99% more error relative to genetic effects
       effectvar2 <- var(P[genotypes])
       residualvar2=(effectvar2-h22*effectvar2)/h22
       residualvar2
       residual2=rnorm(n2,0,sqrt(residualvar2))
       error<-residual2

       Data.sub3[,4]<-Data.sub3[,4]+error
      }

      if (m=="HTP.80"){
       n2 <- nrow(Data.sub3)
       h22<-.2 #adds 80% more error relative to genetic effects
       effectvar2 <- var(P[genotypes])
       residualvar2=(effectvar2-h22*effectvar2)/h22
       residualvar2
       residual2=rnorm(n2,0,sqrt(residualvar2))
       error<-residual2

       Data.sub3[,4]<-Data.sub3[,4]+error
      }

      DataSub<-Data.sub3
      if (ENVIRONMENT!=1){
      fm1 <- lmer(Hgt ~ (1|Gen) + (1|Env) + (1|Gen:Env) + (1|Env:Rep), data = DataSub)
#all random
      }
      if (ENVIRONMENT==1){
      fm1 <- lmer(Hgt ~ (1|Gen) +  (1|Gen:Env) + (1|Env:Rep), data = DataSub) # if only 1
env
      }

      #fm1 <- lmer(Hgt ~ 1 + Gen +  (1|Gen:Env) + (1|Env:Rep), data = Data)
      #summary(fm1)

      varcomp<-as.data.frame(VarCorr(fm1))
```

```
var<-(select(varcomp, grp, vcov))

var[6,1]<-"Total"
var[6,2]<- sum(var$vcov[1:5])
var$pervar<-0
var$pervar[1:6]<-round((var$vcov[1:6]/var[6,2])*100,digits=3)
###
if (ENVIRONMENT!=1){
  fm1 <- lmer(Hgt ~ 1 + Gen + (1|Env) + (1|Gen:Env) + (1|Env:Rep), data = DataSub)
}
if (ENVIRONMENT==1){
  fm1 <- lmer(Hgt ~ 1 +Gen +  (1|Gen:Env) + (1|Env:Rep), data = DataSub) # if only 1
env
}
#gives estimated effect for each genotype, you could then select best genotypes to
advance

est<-as.data.frame(fixef(fm1))
est$estimated.gen<-0

est$estimated.gen[1]<- est[1,1] # first genotype is the intercept

est$estimated.gen[2:(nrow(est))]<-
(est$`fixef(fm1)`[2:nrow(est)])+est$estimated.gen[1]
#this adds the intercept to the effects to achieve estimated genotypic height

########## SINGLE QTL ANALYSIS
BLUP<-est$estimated.gen #give BLUP to data frame

estimates<- as.data.frame(cbind(RIL[genotypes,],BLUP))

single.marker<- matrix(NA,SNPs,6)

single.marker <- as.data.frame(single.marker)
colnames(single.marker) <-
c("SNP","Effect.Present","Effect.Detected","Pvalue","Est.Homozygote","True.Effect")
round<-0

for (MARKER in 1:SNPs){
  QTL<- estimates[,MARKER]
  V<- colnames(estimates[MARKER])

  if(length(levels(as.factor(estimates[,MARKER])))==1){next} ## skip locus if fixed
```

```
TEST<-(aov(BLUP~as.factor(QTL), data=estimates))
MEANS<-model.tables(TEST,"means")
EFFECTS<-model.tables(TEST,"effects")
summary(TEST)
round<-round+1
if(na.omit(summary(TEST)[[1]][["Pr(>F)"]]) < 0.05){
  single.marker$Effect.Detected[round]<- "YES"
}
if(na.omit(summary(TEST)[[1]][["Pr(>F)"]]) > 0.05){
  single.marker$Effect.Detected[round]<- "NO"
}
if (AlleleEffect.ht[MARKER]==0){
  single.marker$Effect.Present[round]<-"NO"
}
if (AlleleEffect.ht[MARKER]!=0){
  single.marker$Effect.Present[round]<-"YES"
}
if(length(levels(as.factor(estimates[,MARKER])))==3){ #if heterozygotes exist
  single.marker$Est.Homozygote[round]<-
round((EFFECTS$tables$`as.factor(QTL)`[3]-EFFECTS$tables$`as.factor(QTL)`[1]),4)[[1]]
#homozygote
}
if(length(levels(as.factor(estimates[,MARKER])))==2){ #if inbred at locus
  single.marker$Est.Homozygote[round]<-
round((EFFECTS$tables$`as.factor(QTL)`[2]-EFFECTS$tables$`as.factor(QTL)`[1]),4)[[1]]
#homozygote
}
single.marker$True.Effect[round]<-AlleleEffect.ht[MARKER]*2 #real homozygous
effect
single.marker$Pvalue[round]<-na.omit(summary(TEST)[[1]][["Pr(>F)"]]) #pvalue
column
single.marker$SNP[round]<- MARKER
}

single.marker<-na.omit(single.marker)

#from this results table we can then generate % of false positives, missed QTL,
#and accuracy of estimations, this then allows us to compare scenarios
#*********************************
```

```
        Percent.Detection<-
sum(single.marker$Effect.Present=="YES"&single.marker$Effect.Detected=="YES")/sum(si
ngle.marker$Effect.Present=="YES")

        False.QTL <-
sum(single.marker$Effect.Present=="NO"&single.marker$Effect.Detected=="YES")

        Percent.False <-
sum(single.marker$Effect.Present=="NO"&single.marker$Effect.Detected=="YES")/sum(sin
gle.marker$Effect.Detected=="YES")

        r.qtl<-lm((subset(single.marker, Effect.Detected %in% "YES" & Effect.Present %in%
"YES")$Est.Homozygote)~
              (subset(single.marker, Effect.Detected %in% "YES" & Effect.Present %in%
"YES")$True.Effect))

        r2.qtl<-round((summary(r.qtl)$r.squared), digits=2)
        #****************************************
        ### END SINGLE QTL
        est$true.gen<-P[genotypes]

        est$error<- (est$true.gen)-(est$estimated.gen)

        est$error2<- (est$error*est$error)

        RMSE <- sqrt((sum(est$error2))/(nrow(est)))

        #NRMSE<- (RMSE)/(max(DataSub[,4])-min(DataSub[,4]))#skewed due to outliers in
HTP error
        NRMSE<- (RMSE)/(max(P[genotypes])-min(P[genotypes]))

        reg<- lm(est$true.gen~est$estimated.gen)

        row <-row + 1 ; print(row)

        results$Scenario[row]<- row
        results$Population[row]<- "RIL"
        results$Genotypes[row]<- GENOTYPE
        results$Environments[row]<-ENVIRONMENT
        results$Reps[row]<-REP
        results$Percent.Detection.QTL[row]<-Percent.Detection
        results$False.QTL[row]<- False.QTL
        results$Percent.False.QTL[row]<-Percent.False
```

```
        results$r2.QTL[row]<-r2.qtl
        results$n[row]<-nrow(DataSub)
        results$measurement[row]<-m
        results$r2[row] <- round((summary(reg)$r.squared), digits=2)
        results$RMSE[row] <- RMSE
        results$NRMSE[row] <- NRMSE
        results$H2[row] <-
(var[2,2])/((((var[5,2])/REP*ENVIRONMENT)+((var[1,2])/ENVIRONMENT)+(var[2,2]))
        results$var.gxe[row]<-var[1,2]
        results$var.gen[row]<-var[2,2]
        results$var.rep[row]<-var[3,2]
        results$var.env[row]<-var[4,2]
        results$var.residual[row]<-var[5,2]
        results$per.var.gxe[row]<-var[1,3]
        results$per.var.gen[row]<-var[2,3]
        results$per.var.rep[row]<-var[3,3]
        results$per.var.env[row]<-var[4,3]
        results$per.var.residual[row]<-var[5,3]


    }#end of error loop
   }#end of rep loop
  }#end of env loop
 }#end of genotypes loop
#}#end of subsetting loop
}#end of iterations loop

results

#results3<-rbind(results[1:row,], results2)

## add results2 and results together


results2<-results[1:row,]
```