Experimental and computational methods to assign gene function to maize genes

Wimalanathan Kokulapalan

Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

Part of the Bioinformatics Commons, Biology Commons, and the Genetics Commons

Recommended Citation
Kokulapalan, Wimalanathan, "Experimental and computational methods to assign gene function to maize genes" (2018). Graduate Theses and Dissertations. 16830.
https://lib.dr.iastate.edu/etd/16830

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Experimental and computational methods to assign gene function to maize genes

by

Wimalanathan Kokulapalan

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Bioinformatics and Computational Biology

Program of Study Committee:
Carolyn Lawrence-Dill, Co-major Professor
Erik Vollbrecht, Co-major Professor
   Julie Dickerson
   Peng Liu
   Paul Scott

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2018

Copyright © Wimalanathan Kokulapalan, 2018. All rights reserved.
DEDICATION

To my mother who has been a role model and an inspiration
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACKNOWLEDGEMENTS</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td>ABSTRACT</td>
<td>viii</td>
</tr>
<tr>
<td>1</td>
<td>GENERAL INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Research Goals</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Dissertations Organization</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Contributions to projects</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Figures and Legends</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>SEQUENCE-BASED METHODS TO FIND THE CAUSAL MUTATION OF MAIZE MUTANTS</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Abstract</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Methods</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Results</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>Discussion</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Author Contributions</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Acknowledgements</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Tables and Legends</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Figures and Legends</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>IDENTIFICATION OF GENOMIC T-DNA INSERTION LOCATIONS USING FLANKING GENOMIC SEQUENCES IN TRANSGENIC MAIZE</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>Abstract</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>Methods</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>Results</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>Discussion</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>Author Contributions</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>Tables and Legends</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>Figures and Legends</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>86</td>
</tr>
<tr>
<td>4</td>
<td>MAIZE GO ANNOTATION- METHODS, EVALUATION, AND REVIEW (maize-GAMER)</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>Abstract</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>Introduction</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>Materials and Methods</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>Results</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>Discussion</td>
<td>114</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>Author Contributions</td>
<td>116</td>
<td></td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>116</td>
<td></td>
</tr>
<tr>
<td>Tables</td>
<td>118</td>
<td></td>
</tr>
<tr>
<td>Figures and Legends</td>
<td>122</td>
<td></td>
</tr>
<tr>
<td>References</td>
<td>129</td>
<td></td>
</tr>
</tbody>
</table>

| CHAPTER 5. GENE ONTOLOGY META ANNOTATOR FOR PLANTS (GOMAP)              | 132  |
| Abstract                                                                | 132  |
| Introduction                                                            | 133  |
| Methods                                                                 | 134  |
| Results                                                                 | 137  |
| Discussion                                                              | 138  |
| Acknowledgements                                                        | 140  |
| Tables                                                                  | 141  |
| Figures and Legends                                                     | 144  |
| References                                                              | 148  |

| CHAPTER 6. GENERAL CONCLUSIONS                                          | 150  |
| Summary                                                                 | 150  |
| Contributions to collaborative projects of significance                 | 151  |
| References                                                              | 155  |

| APPENDIX A. UREA GENOMIC DNA EXTRACTION PROTOCOL                       | 156  |
| APPENDIX B. BSA-GBS PROTOCOL                                            | 158  |
| APPENDIX C. T-DNA FLANK AMPLICON LIBRARY CONSTRUCTION                  | 164  |
| APPENDIX D. maize-GAMER SUPPORTING MATERIALS                            | 172  |
| Detailed Metrics                                                        | 172  |
| Reproducibility                                                         | 174  |
ACKNOWLEDGEMENTS

The completion of this thesis and my Ph.D. would not have been possible without the help from a long list of people, and I am grateful for the support during my graduate studies.

I would like to start by thanking my co-major professors Dr. Carolyn Lawrence-Dill and Prof. Erik Vollbrecht for their kindness and support throughout the PhD. I thank the other committee members Prof. Julie Dickerson, Dr. Peng Liu, and Dr. Paul Scott for providing valuable suggestions, and asking questions that made me critically think about my research throughout. Thank you, Carolyn, for being supportive and nurturing during the early years of my PhD and providing opportunities to attend conferences. Thank you for supporting my ideas, and at times challenging them so I could grow better as a scientist. Thank you for organizing the weekly lab lunches and get-togethers to help us bond outside of work. Thanks for your patience and letting me learn by mistakes. I am grateful to Darwin Campbell for being a great colleague during my time at MaizeGDB and Dill Plant Informatics and Computational lab. Thanks for putting out the occasional fires in the lab. Thank you, Ian Braun, for the thoughtful discussions on ontologies, and computation on ontologies. Thank you, Carla Mann, Mingze He, Antonio Brazleton, Chris Lawrence, Naser Alkhalifah, and Renee Walton for the discussions and the fun times in the lab.

Thank you, Erik, for letting me be part of the lab, and letting me learn at my own pace. Especially, the freedom that allowed me to design and execute my own projects. Thanks for being a sounding board and guiding me in the right directions. Thank you for the patient conversations in the field about genetics and letting me into your exciting world of maize mutants. Thank you for providing the freedom and a platform for graduate students to grow and develop as independent scientists. I would like to thank Erica Unger-Wallace for being an
extraordinary mentor in the lab and teaching me everything I know about wet-lab protocols. Thank you, Erica, for providing me guidance in the world of genetics and molecular biology. The discipline and planning you possessed has allowed me to understand the effort to generate good data for the analysis. The critical thinking and insightful discussion have made projects from not only the wet-lab but also dry-lab more thorough. Thank you for being one of the unofficial counselors when my life was taking unfortunate turns. I am grateful for the time I got to spend in the lab and the field with Dr. Joshua Strable and graduate students. The numerous discussions we have had in the field and over lunches have helped me grow as an individual and a scientist. Thank you, Dr. Rebecca Weeks, for the hard work that set up the projects I worked on. Thank you, Takao Shibamoto, for working with me and learning and letting me learn from our interactions. Thank you, Brian Zebosi, Ryan Arndorfer, and other graduate students, for the help in the lab. Thank you, Sarah Briggs, for the work ethic and responsibility during the time in the lab.

I would like to thank a long list of friends for their support during my PhD. My cohort who joined the BCB program, Kannan Shankar, Jennifer Chang, and Ruolin Liu for the exciting times during the first two years. My dearest friends who were here in the time of need in the BCB program, Bekah Starks, Sweta Roy-Carson, John Hsieh, Gaurav Kandoi, Ashish Jain, and Naihui Zhou. I am thankful for the support provided by the Program Coordinator Trish Stauble during the trying times in the program. I also would like to thank my Sri Lankan friends in Ames for the numerous get-togethers and all the help provided during my stay in Ames. Dulan and Thanuja you have been in my corner all the way from college till now, and I am very grateful for that. Finally, I thank my extended family without whom I would have not been able to pursue my
dream of obtaining a PhD. I am thankful to my mother Vimaladevi Wimalanathan who has been an inspiration to me and has made me a not only a better student also a better human being.
Maize is an important crop species and is the highest produced cereal crop in the world as well as a model species for genetics and genomics research. For this reason, researchers have been very successful in translating understanding of basic biological processes into improved crops for over 100 years. Maize researchers have a long history of utilizing genetic techniques to dissect the function of genes that control biological processes. Characterizing and cloning mutants precisely defines gene function but is a slow process that can take years to accomplish. Alternatively, computational methods provide a faster way to assign predicted function to genes by leveraging the vast knowledge base of gene function gathered by experimental and curatorial efforts in multiple species. Computational methods can be used to predict functions for genes at a genome-wide scale. Ideally, improved computational predictions would narrow and target experiments that would be used to test gene function, thus speeding the process of experimental characterization. We have created methods to improve discrete steps in both experimental characterization and computational prediction of gene function in maize.

For the experimental work, we have developed molecular methods, leveraging the decreasing high-throughput sequencing cost, and bioinformatics analysis pipelines, capitalizing the availability of multiple maize genome assemblies, that improve positional cloning of maize mutants. We have also focused on methods to improve identification of T-DNA integration locations genome-wide for maize. Genes responsible for mutant phenotypes are often studied using transgenic techniques to manipulate function at a molecular level. These techniques typically integrate a transfer DNA (T-DNA) fragment into the host genome, where genome integration context may have crucial effects on transgene expression. Current methods to identify T-DNA integration locations are either cumbersome or imprecise for repetitive rich genomes like
maize. We developed a molecular protocol that utilizes long-read sequencing to enrich genomic T-DNA flanks, thus revealing T-DNA placement more precisely.

Working to identify and characterize genetic variants responsible for specific phenotypes gives insight into how critical the quality of predicted gene function annotations can be to inform and guide experimental investigation. Functional annotation data are used for the interpretation of results from large-scale studies such as transcriptomics and proteomics. In addition, these data are also used to inform and prioritize candidate genes potentially responsible for a phenotype for positional cloning, genetic association, and other studies. To improve the quality of predicted gene functions available for all researchers working in maize, we generated a high-coverage, high-confidence, and reproducible functional annotation dataset for maize genes using the Gene Ontology. Methods we used to generate GO annotations for maize are generic and applicable to other plants. To enable application to other species, we formalized the method used to annotate maize as a containerized pipeline called GOMAP. GOMAP has been optimized for use in high-performance computing environments and has been tested on additional maize lines and other plant species.
CHAPTER 1. GENERAL INTRODUCTION

Introduction

Maize is not only the highest yielding cereal species in the world (Food and Agriculture Organization of the United Nations, 2018), but also an important model species that has been used to study various biological processes such as transposon function and inflorescence development (Bortiri et al., 2006; Lazarow, Doll, & Kunze, 2013; Vollbrecht, Springer, Goh, Buckler, & Martienssen, 2005). Maize has a rich collection of mutants that have been characterized to understand various biological processes (Neuffer, Coe, & Wessler, 1997). Phenotypic characterization of a mutant is essential to understand the biological processes affected in the mutant but doesn't give the complete information for functional characterization. On the other hand, identification of the mutation-affected gene (i.e. cloning of the gene) allows for better experimental design and more powerful genetics to understand the function of the identified gene at a molecular level (Gallavotti & Whipple, 2015).

At present, characterization of a mutant typically begins with the phenotypic characterization. The phenotypic characterization can begin with identification of the effect on different tissues and organs, or it could be a protein or RNA expression profiling of the mutant to identify the various molecular processes that are affected (Whipple et al., 2011). In conjunction with phenotypic characterization, forward genetics is used to clone the mutation-affected gene using positional cloning technique. Positional cloning is the technique used to progressively narrow down the chromosomal location the causal mutation is in, and identify the causal mutation (Gallavotti & Whipple, 2015). Once the gene is cloned, various techniques can be used to characterize how that gene functions (Satoh-Nagasawa, Nagasawa, Malcomber, Sakai, & Jackson, 2006; Strable et al., 2017; Tanaka, Pautler, Jackson, & Hirano, 2013; Whipple et al.,
These experiments depend on the nature of the gene and can include molecular techniques such as \textit{in situ} hybridization and immunohistochemistry, transgenic approaches such as overexpression and fluorescence tagging, and biochemical techniques such as enzyme assays and immunoprecipitation (Bolduc & Hake, 2009; Eveland et al., 2014; Pautler et al., 2015; Strable et al., 2017; Zalabák et al., 2014). The functional characterization of a cloned maize mutant allows precise definition of the function for the cloned gene.

The first version of maize reference genome was released in 2009, and has been iteratively improved over time (Jiao et al., 2017; Schnable et al., 2009). The structural annotation (gene calling) of maize genome sequence has annotated gene structures for different types of genes including protein coding genes (Jiao et al., 2017). Around ~40,000 protein coding genes have been annotated in the maize genome, and precise experimental characterization of every predicted gene may not be feasible. The number of genome-wide association studies (GWAS) and transcriptomic studies have increased over the years and rely on functional annotations to interpret the results from the analyses. Computational methods to annotate gene function would utilize the information from functions defined by functional characterization of mutants in other species to predict function to the protein coding genes in maize. Three general categories of computational methods could be used for computational functional annotation, such as sequence-similarity, domain-presence, and mixed-methods (Radivojac et al., 2013).

Both defined and predicted functions are combined to generate functional annotations that are released for public use. While free text descriptions are useful to describe functions, standardizing the language across multiple organisms and even across multiple labs within the same species is challenging. Standardized vocabulary to describe function has been essential to make functional annotations comparable across analyses within the same species and across
different species. Several projects have defined functions using different aspects, Gene Ontology (GO), domain databases such as InterPro, Pfam, and Panther, and categories in MapMan. Gene ontology annotations have been widely used for downstream analysis of GWAS, gene expression datasets (Huntley et al., 2015; Li et al., 2016; Liu et al., 2018; Thakare, Zhang, Wing, Cotty, & Schmidt, 2017). Advantages of using GO for functional annotation include the availability of annotation methods, existing datasets for comparison, and analysis of large-scale datasets.

**Research Goals**

In my research I sought to improve bioinformatic and molecular approaches for narrowing the genomic regions potentially responsible for a phenotypic observation. I have approached this topic by improving the methods used to clone maize mutants, improving methods to detect transgene insertion locations to better characterize transgene functions, improved genome-wide maize GO annotation dataset to enable interpretation of large-scale experiments, and developing a streamlined pipeline to predict improved annotations for other plant species.

The process of characterizing maize mutants has immensely benefited from the release of maize genome sequence. The decreasing cost has enabled the use of HTS to speed up positional cloning of maize mutants. Both parts of the positional cloning process, rough-mapping and fine-mapping can be improved using HTS methods. One of the research goals was to utilize HTS to improve the positional cloning method used in the lab. As part of this goal, a molecular protocol was developed to use HTS for the rough-mapping of maize mutants. Furthermore, whole-genome-sequencing (WGS) approached was used to positionally clone maize mutants as well. Utility of transgenics to characterize maize mutants has been well documented, and in most cases, it is not essential to identify the genomic location of T-DNA insertion. There some specific transgenic experiments where the T-DNA insertion location will affect the transgene
expression. The goal was to develop a molecular protocol to use long-read sequencing to identify the T-DNA insertion locations. The above-mentioned goals help with faster characterization of the mutants or precise definition of gene function, but the methods used to define functions for genes are generally slow. The third goal was to use computational methods to improve the maize functional annotation dataset using gene ontology. The goal was to improve three aspects of GO annotations for maize, quantity, quality and reproducibility. The methods used for the third goal were used to develop a streamlined pipeline for the genome-scale functional annotation of plant genes, that could be used to predict improved annotations for other plant species as well.

**Dissertation Organization**

This dissertation includes a general introduction (Chapter 1), four journal papers (Chapters 2-5), and general conclusions (Chapter 6). The general introduction contains an overview about how a gene that is associated with a mutant phenotype is cloned, and how the biological processes regulated by the characterized gene is elucidated after the mutant is cloned. Chapter 2 details the application of high-throughput sequencing to improve the positional cloning of maize mutants. Chapter 3 describes the development of a molecular protocol to accurately identify the genomic location of transgenic DNA insertions in the maize genome using long-read sequencing technique. Chapter 4 details the use of computational methods to improve functional annotation of maize genes, and the evaluation of the predicted annotations. This manuscript has been published in Plant Direct journal, and the improved annotations have been released. Chapter 5 describes the GOMAP tool which is a streamlined-pipeline of the reproducible methods used in maize-GAMER. Chapter 6 describes the general conclusions and includes a summary of the thesis and future directions as well as a listing of additional research accomplished during my doctoral studies that are outside the scope of my dissertation research topic.
Contributions to projects

Chapter 2 was a collaborative project that was performed by different members in the lab. Dr Vollbrecht generated the mapping populations and phenotyped the mutants with Rebecca Weeks. Rebecca Weeks scored mutants and generated the bulked pools. Erica Unger-Wallace extracted high-molecular weight DNA and fine-mapped mutants. My contribution was the development of the bulked-segregant analysis using genotyping-by-sequencing bench protocol and the bioinformatics pipelines to analyze BSA-GBS, BSA-seq, and whole-genome sequencing data. Work described in Chapter 3 was performed with transgenic DNA constructs generated by Sarah Choudury from the Slotkin lab. Erica Unger-Wallace oversaw the transgenic event generation and experiments related to maize transformation. My contribution was the design of molecular protocol to enrich T-DNA flanks with Erica, and the bioinformatic analysis of the data generated from the bench protocol. Erica performed the genome specific PCR to confirm the locations identified from the T-DNA protocol. My contribution to Chapter 4 was the conceptualization of the project with Dr. Carolyn Lawrence-Dill. I performed the annotation and of the maize genes, and subsequent evaluation of the annotations using the gold-standard dataset obtained from MaizeGDB. My contribution to the work described in Chapter 5 was to containerize the methods used in maize-GAMER and test it on different HPC systems.
Figure 1.1: Overview of how functional annotations are assigned in maize
Cloning and characterization of maize mutants can lead to precise definition of function. The assembly of the genome sequence and subsequent structural annotation of genes could enable the computational prediction of functions. The functional annotation dataset will contain the precisely defined functions from experimental evidence, as well as predictions from computational tools.
References


CHAPTER 2. SEQUENCE-BASED METHODS TO FIND THE CAUSAL MUTATION OF MAIZE MUTANTS

A manuscript to be submitted

Wimalanathan Kokulapalan, Erica Unger-Wallace, Rebecca Weeks and Erik Vollbrecht

Abstract

Characterizing mutants has been invaluable in understanding how agricultural traits and basic biological processes are affected in maize. A common technique that is used to identify the causal gene to the mutant phenotype is called positional cloning. Identifying the causal gene using positional cloning has two major genetic mapping steps, rough-mapping that typically uses bulked-segregant analysis (BSA) to narrow down a causal locus, and a fine-mapping step to identify the causal mutation and the causal gene. We used high-throughput sequencing (HTS) methods to improve both rough mapping, and fine-mapping steps. A HTS protocol for BSA was developed by adapting the genotyping-by-sequencing (GBS) method (BSA-GBS), and was applied to mutants in the lab. A bioinformatics pipeline was developed to map mutants using BSA-GBS data. BSA-GBS was applied to the mutants in the lab and causal loci identified have been confirmed. We also compare the BSA-GBS method to BSA using a maize SNP array and show that the cost and performance of the methods are comparable. A whole-genome-sequencing (WGS) approach was also utilized to clone mutants in the lab. One mutant that was rough mapped with BSA-GBS was cloned by WGS, and the other unmapped mutant was cloned by BSA-seq. Bioinformatics analysis of the WGS and BSA-seq combined with fine-mapping efforts have successfully narrowed down the putative causal mutation down to single mutations.

Introduction

Maize is an important crop species, and the highest produced cereal crop in the world (FAOSTAT 2018). Maize has also been used as a model organism for understanding various
biological processes relating to crop species such as transposon function, inflorescence development, and carbohydrate metabolism (Bortiri et al., 2006; Lazarow, Doll, & Kunze, 2013; Sosso et al., 2015). Maize has benefited from forward genetics approaches both as a crop species and a model organism to study diverse biological processes ((Gallavotti & Whipple, 2015). Forward genetics approaches are used to identify and clone the gene responsible for a phenotype from a mutant that shows aberrant phenotype. Maize has a rich mutant collection that has been used to characterize various biological processes, and positional cloning has been used to characterize a number of these mutants (Neuffer, Coe, & Wessler, 1997). Positional cloning is the technique used to progressively narrow down the chromosomal location the causal mutation is located in and identify the causal mutation (Gallavotti & Whipple, 2015). The release of the maize genome in 2009 and subsequent iterative improvements of the maize reference genome have been invaluable in the progress of positional cloning techniques (Jiao et al., 2017; Schnable et al., 2009).

Positional cloning approaches can be used to characterize mutations from any source, but the effort and time to clone the gene varies based on the source of the mutation. Maize mutants have been generated by different mutagenesis techniques, such as natural variation, transposon, radiation and chemical mutagenesis (Neuffer et al., 1997). Ethyl methane sulfonate (EMS) mutagenesis has been a commonly used chemical mutagenesis method to generate maize mutants (Gallavotti & Whipple, 2015). The G/C to A/T (~99%) single-nucleotide polymorphism (SNP) is the most common mutation induced by EMS (Greene et al., 2003). After the EMS mutant is generated and the underlying genetics has been characterized, a mapping population is generated to positionally clone the gene. The manuscript will focus on methods to clone maize mutants generated with EMS mutations.
Mapping populations are generated by leveraging the genetic diversity among different maize inbred lines. Maize inbred lines show extensive genetic and phenotypic diversity (Schnable et al., 2009). A number of maize inbred lines have been used for positional cloning in the past, such as B73, Mo17, W22 and Oh43, and A619 (Gallavotti & Whipple, 2015). There are several different strategies to generate mapping populations such as F2 populations, doubled haploid lines, and back cross populations (Y. Xu, Li, Yang, & Xu, 2017). In the case of a single locus recessive mutant, a common strategy to generate a mapping population is to generate an F2 mapping population. The first step is to cross the recessive mutant in Mo17 onto a wild-type plant in another inbred line such as B73. The hybrid F1 generation from the cross, which will all be normal plants, is selfed to generate an F2 generation (Figure 2.1). The F2 population generated by this cross is called the mapping population. Due to crossover of parental chromosomes in meiosis, F1 gametes will have recombination events, and a genotype where each chromosome is a mosaic of B73 and Mo17. Individuals in the F2 generation will have different mosaic patterns of B73 and Mo17 genotypes across the pair of chromosomes the causal locus is on. On average, the mutant plants in the F2 will have homozygous mutant genotype at and nearby the causal locus, while normal plants could be heterozygous at and nearby the causal locus or homozygous for B73 at that location.

For a recessive mutant, the F2 segregation ratio of the phenotype of the mutant is 1:3, and the genotype segregates 1:2:1 for the mutants:heterozygotes:normals. Completely linked Mo17 markers will show perfect cosegregation with the mutant phenotype, and completely unlinked loci will not show any cosegregation (Figure 2.2). If individuals are pooled by phenotype, the Mo17 frequency in the normal pool for a linked locus is between 1/2 and 1/3, and complete linkage is 1/3. In the mutant pool, the Mo17 allele frequency for the linked locus is between 1/2
and 1. In both the normal and mutant pools, the Mo17 allele frequency of an unlinked locus is 1/2 (Figure 2.2). A simple ratio of those Mo17 allele frequencies, computed as a ratio of mutant:normal pool is between 1 to 3, where completely linked makers show the highest ratio (Figure 2.2). The chromosomal region enriched with non-equal mutant:normal ratio should identify the region the causal locus is located in.

This technique to use mutant and normal pools from a mapping population to identify the chromosomal interval of the causal mutation is called bulked-segregant analysis (BSA) (Gallavotti & Whipple, 2015). BSA utilizes the DNA or RNA extracted from pools bulked by phenotype from the mapping population. The extracted DNA or RNA is genotyped for markers across the genome. A number of different techniques have been used in the recent past to make genome-wide genotyping cheaper and faster, such as Sequenome, microarray, and high-throughput sequencing (Becker, Chao, Zhang, Salt, & Baxter, 2011; Elshire et al., 2011; Ganal et al., 2011; Hill et al., 2013; Liu et al., 2010; Liu, Yeh, Tang, Nettleton, & Schnable, 2012).

Sequenome uses the MASSARRAY platform to identify SNP markers for the mutant and normal pool DNA (Gabriel, Ziaugra, & Tabbaa, 2009). MASSARRAY platform uses a locus-specific PCR, followed by a single base extension of a tagged PCR primer at the site of interest, and mass spectrometry (Gabriel et al., 2009). The mass of the extended primer is used to identify each SNP allele. The maize Sequenome SNP-typing assay developed by Liu et al. was the first high-throughput genome-wide genotyping assay used for bulked-segregant analysis in maize (Liu et al., 2010). The assay genotyped ~1000 markers across the maize genome, derived from B73 and Mo17 inbred lines.

Different microarray-based SNP arrays have been designed for maize (Ganal et al., 2011; Illumina, 2018; C. Xu et al., 2017). These SNP arrays detect more markers than Sequenome. The
commonly used MaizeSNP50 is a commercially available kit for purchase from Illumina, and is also provided as a service by various vendors and core facilities (Illumina, 2018). The pooling process of BSA using SNP arrays is similar to Sequenome analysis, except that DNA from each pool is hybridized directly to a chip. Where the B genome is defined as the one mutagenized as described above (e.g., Mo17), the B-allele frequency, which is the normalized measure of the allelic intensity ratio, is calculated for each marker for each pool, and the $\log_2$ transformed $\frac{B_{\text{mutant}}}{B_{\text{normal}}}$ ratio is plotted, and the causal region is detected by identifying significant peaks in the plot (Peiffer et al., 2006).

Several different short read high-throughput-sequencing (HTS) methods also have been used for mapping mutants in maize and other species (Elshire et al., 2011; Etter, Bassham, Hohenlohe, Johnson, & Cresko, 2011; Hill et al., 2013; Jander et al., 2002). Maize has a large genome with ~85% repetitive sequences (Schnable et al., 2009). Initial HTS approaches for BSA in maize relied on reduced-representation-sequencing (RRS) to reduce the proportion of genome sequenced while increasing the depth of the sequenced portion (Elshire et al., 2011; Liu et al., 2012). Both DNA and RNA has been used for generating sequencing libraries for BSA. RNA-seq is an RRS method by design, because only a small proportion of the maize genome is expressed in any given tissue. Bulked-segregant RNA-seq (BSR-seq) developed by Liu et al uses RNA-seq of the bulked-pools for BSA and subsequent transcriptome analysis (Liu et al., 2012). The Genotyping-by-sequencing (GBS) method developed by Elshire et al is optimized for the RRS of DNA in maize (Elshire et al., 2011). The method was developed for population studies of different maize inbred lines and recombinant-mapping populations. GBS is optimized for barcoding a large number of individuals in a population. Although GBS has been successfully used for genetic mapping of mutations by barcoding and sequencing individuals in the mapping
population, the cost can be reduced by working with bulked DNA pools. We adapted the GBS method that is designed for population genetics studies and applied it to bulked pools for BSA, which we called Bulked-segregant-analysis by Genotyping-by-sequencing (BSA-GBS). We show the utility of BSA-GBS for rough mapping for positional cloning and show that it has some distinct advantages over other methods for rough-mapping.

The BSA approach using above mentioned methods defines a rough chromosomal interval, and subsequent fine-mapping efforts are necessary to refine the interval and clone the gene. Fine-mapping is the process of analyzing DNA from single recombinant individuals with successive genotyping assays to iteratively refine the interval. The process requires genotyping of existing or additional markers (e.g. indels, SSR, CAPS, dCAPS, KASP, Sanger sequencing) within the causal locus, usually for mutant individuals. The fine-mapping process is laborious and could take a long time. Often the number of mutant individuals generated in a mapping population is not enough to complete the fine-mapping and additional crosses have to be performed to generate new mutants with different recombination break points. Furthermore, the time taken for fine-mapping also depends on the size of the causal locus identified by BSA.

While HTS methods can improve the rough mapping step, identifying the exact mutation still requires a fine-mapping step. The time taken for fine-mapping could be substantially long, especially if a large chromosome interval was identified as the causal locus. Whole-genome sequencing (WGS) has been used in other species for BSA and subsequent cloning of the gene maize (Rowan, Patel, Weigel, & Schneeberger, 2015; Schneeberger, 2014; Takagi et al., 2013). WGS has recently been successfully used for mapping bulked-segregants in maize (Klein et al., 2018). With a WGS approach, both rough mapping and fine mapping steps of positional cloning technique can be combined into a single sequencing and ensuing analysis efforts. We applied the
WGS approach to map a bulked-segregant mutant using bulked-segregant-analysis sequencing (BSA-seq) (Klein et al., 2018), and designed an optimized bioinformatics pipeline to identify putative causal mutations. We also used a WGS of a mutant individual from a mutant that was rough mapped using BSA-GBS and identified a list of putative causal mutations. Bioinformatics analysis combined with the fine-mapping efforts have narrowed the putative causal mutation to a single candidate nucleotide mutation in both mutants. We describe the pipeline used to analyze the data, as well as compare the complexities of finding the causal mutation with and without rough mapping in maize.

**Methods**

**Mutants and generation of Bulked-Segregants**

Four of the mutants from Mo17 background and two mutants from W22 background were mapped using BSA-GBS, BSA-seq, WGS and SNP array (Table 2.1), and mapping populations were generated using the general schematic shown in Figure 2.1. The F2 plants were scored based on the phenotype as mutants or normals. The leaf tissue samples for the bulks were collected using a 6mm hole puncher. About 150-300 punches were collected for each bulk, and an equal number of punches were collected from each individual plant in the bulked sample. This varied between 6 and 13 punches per individual depending on the number of individuals in the pool. The leaf tissue from each pool was ground in liquid nitrogen and high molecular weight (HMW) genomic DNA was extracted using a Urea-based protocol as described in APPENDIX A.

**BSA-GBS Protocol**

A genotyping-by-sequencing (GBS) protocol was developed and optimized for maize by Elshire et al., (Elshire et al., 2011). The original GBS protocol was adapted and applied to the
DNA extracted from bulked pools. Our modified protocol, called BSA-GBS, is described below. Initially, the genomic DNAs were quantified using the Promega Quantifluor dsDNA system, which provides a more accurate measure of DNA concentration than NanoDrop based measurements. It is essential the genomic DNA is quantified accurately, because the BSA-GBS steps rely on accurate genomic DNA concentration. BSA-GBS was used to construct a barcoded sequence library for each bulked pool (Figure 2.3).

The BSA-GBS bench protocol consists of three main steps, namely restriction digestion, adapter ligation, and PCR amplification. First the genomic DNA was digested using ApeKI restriction enzyme (See Figure 2.3). The methylation sensitivity of ApeKI limits digestion in highly methylated regions potentially avoiding some repetitive genomic regions as they tend to be highly methylated (Elshire et al., 2011). Next, adapters were ligated to the digested genomic DNA fragments. Two types of adapters were used for ligation, barcoded adapter (BC adapter) and a common adapter. The barcode enables multiplexing multiple bulked pools in the same sequencing run. Multiple mutants were multiplexed in the same sequencing run to make BSA-GBS cost-effective. Barcoded libraries for 12 bulked pools from 6 mutants were constructed for each sequencing run (Figure 2.3). A subset of 12 barcodes were selected for BSA-GBS from the 96 barcodes designed for GBS (APPENDIX B. ). The fixed restriction site (CWG) leads to low sequence diversity within the first 10 bases of the read, and low sequence diversity at the start of the read reduces the base call quality (Elshire et al., 2011). The length of the barcodes selected (4-8-bp) and sequence composition were optimized to alleviate the effect of a fixed restriction site. The ligated DNA is cleaned up using SPRI beads (as described by the manufacturer, Aline Biosciences) before the proceeding to the PCR step.
The adapter ligated DNA fragments are amplified by PCR to construct the barcoded sequencing library. PCR primers used for amplification add the Illumina adapters necessary for sequencing (APPENDIX B). Adapter dimers that do not contain genomic fragments are produced by the GBS protocol, and this issue persists in BSA-GBS as well. The adapter dimers are amplified and sequenced more efficiently than the fragments with genomic insert because of the smaller size (~120-130-bp). To remove the dimers, we added a bead-based clean-up step to size select the fragments greater than 200-bp after the PCR (Figure 2.4).

The quality and concentration of the libraries were checked using the BioAnalyzer (Agilent) and Qubit Quantifluor dsDNA systems. The average peak size from the BioAnalyzer and the DNA concentration was used to normalize and combine the barcoded libraries for multiplex sequencing. The multiplexed sequencing library was sequenced using the Illumina sequencing platform. Two trials of BSA-GBS were performed, five mutants were mapped in the first trial (Table 2.1), and six mutants were mapped in the second trial (Table 2.2), some from collaborators. Only samples from Vollbrecht lab are included in the analysis. Additional samples from other groups are not included in the mapping analysis presented in the results. The first trial was sequenced using 150-bp single-end sequencing with Illumina NextSeq 500 (Cold Spring Harbor Lab, DNA facility). The second trial was sequenced using 100-bp single-end sequencing with Illumina HiSeq 2500 (Iowa State University, DNA facility).

BSA-GBS Analysis

The overview of the analysis is given in Figure 2.5. The sequences were initially de-multiplexed and separated into independent files for each bulked pool using fastq-mcf (Aronesty, 2013). The barcode was clipped, and low-quality bases are trimmed from both ends using fastq-mcf. Only reads with at least 50-bp length were retained. The reads were then aligned to the
maize reference genome (maize B73 RefGen_AGPv4) using minimap2 (Jiao et al., 2017; Li, 2018). Only uniquely aligned reads were retained after the alignment. MAAPPR method, which was used to analyze RNA-seq data from bulked pools, was used to analyze BSA-seq data (Hill et al., 2013). The MAAPPR method calculates the nucleotide composition of the reads aligned at each genomic position (i.e., counts the reads with A, C, G, T at each position). Euclidean distance (ED) between the normal and mutant pools is calculated using nucleotide composition. Then ED⁴ is calculated from ED to suppress the noise. The BSA-GBS data is prone to noise from the molecular protocol and this noise affects traditional sequencing analysis methods for allele frequency detection such as B allele-frequency (Elshire et al., 2011) and other whole genome sequencing based methods that use G-statistic and QTL-seq (Magwene, Willis, & Kelly, 2011; Takagi et al., 2013). The ED⁴ distances are plotted on a Manhattan plot using ggplot2 to visualize the differences between the pools (Wickham, 2016).

Raw ED⁴ plots were difficult to interpret and determine the putative causal locus. Two independent strategies were employed to identify the causal locus for BSA-GBS. One strategy was to plot a smoothed Manhattan plot to determine the peak location. Local regression was used to smooth the data. A local polynomial model was used to fit 10% nearest neighbors for the local regression. The smoothed data was then plotted on a Manhattan plot to visually identify the chromosome(s) where the causal locus is located. Then a detailed plot with the causal locus was plotted to determine the location of the putative causal locus.

The second strategy was to employ a genomic bin-based method to find the genomic bins where the difference between normal and mutant pools is enriched. The number of positions which had ED⁴ >1 and ED⁴ <= 1 were counted for each bin and compared to the number of positions with ED⁴ > 1 and ED⁴ <= 1 in the whole genome. Fisher’s exact test was performed
based on the numbers counted. The genomic bins were reduced to contiguous intervals. Fine-mapping experiments, performed independent of the genomic-bin based approach, were used to confirm the validity of this method.

**SNP-chip Analysis**

DNA samples from the bulked segregants were sent to GeneSeek (Neogen, Lincoln, NE) for a SNP-chip experiment. Each bulked-pool DNA sample was run individually. Each chip had measurement values for the reference SNP (B73) and the non-reference SNP (B-allele). The first analysis step was to calculate the B-allele frequency at each SNP position for each sample (B-allele frequency=$\log_2 \frac{B_{\text{mutant}}}{B_{\text{normal}}}$). This represents the non-reference allele frequency. Comparison of allele frequency between the mutant and normal bulked pools will give the causal locus (Figure 2.2). In all samples, homozygous SNPs were filtered out by removing markers with a B-allele frequency lower than 0.2 and B-allele frequency greater than 0.8. Removal of homozygous SNPs in either pool will allow for better visualization of the linkage between the causal mutation and the SNP marker. A value of 0.001 was added to B-allele frequencies of 0, so they did not result in infinite values when used in ratio calculations. The mutant pool B-allele frequency is divided by normal pool B-allele frequency and log2 transformed. These values for each remaining SNP are ordered based on chromosome and position, and the log value of the ratio is smoothed using a sliding window. The results are plotted on a Manhattan plot and the putative causal locus is identified.

**Informative Markers in SNP array and BSA-GBS**

The number of informative markers for SNP arrays were calculated based on the B-allele frequency. The markers for which the B-allele frequency was higher than 0.05 were defined as
informative markers for SNP arrays. The positions for which the $E_D^4 \geq 1$ value was defined as informative BSA-GBS makers.

**BSA-seq Analysis**

Two mutants ($knox^{*-11.0227}$ and $rel^{*-12.2995}$) were mapped using a Mo17 (mutant) x B73 mapping population, where high-quality de novo assembled Mo17 and B73 genomes were available at the time of analysis. The availability of both parent sequences allows the comparison of the SNPs to parental genotypes, and sequences generated from the mutant pool should be enough to detect the mutation by comparing the SNP or indel variation to both parental genomes. So only the mutant pool was sequenced for both BSA-seq ($knox^{*-11.0227}$) and WGS ($rel^{*-12.2995}$). For BSA-seq bulked DNA samples were sequenced by Novogene and approximately 50-Gb sequencing data was obtained for each sample (~20-25x average depth). Samples were sequenced on Illumina HiSeq X 150-bp paired-end platform. Sequences from both samples were run through the pipeline illustrated in Figure 2.6. Sequencing adapters were clipped from the reads, and low-quality bases were trimmed from the ends using fastq-mcf (Aronesty, 2013). Reads were filtered based on length, and only reads longer than 50-bp were retained. The filtered reads were aligned to the maize B73 (RefGen_v4) genome using minimap2, and only uniquely aligned reads were retained (Jiao et al., 2017; Li, 2018). SNPs against the B73 genome were called using samtools (Li, 2011). The maize Mo17 de novo genome assembly was downloaded and aligned to the maize B73 genome (Sun et al., 2018). Mo17 SNPs were identified using the genome alignment against B73. The SNP calls from the mutant alignments and Mo17 genome alignments were compared and identical SNPs were categorized as Mo17 SNPs. The putative causal locus was identified by counting the number of homozygous Mo17 SNPs per genomic-bin from the mutant WGS and plotting it across the genome. Effects of SNPs on the gene models were predicted using the Ensembl Variant Effect Predictor (VEP) tool (McLaren et al., 2016).
The effects of non-Mo17 SNPs were filtered to generate a list of candidate mutations. The mutants are currently being fine-mapped to confirm the causal mutation.

**Results**

**Mutants Mapped by the BSA-GBS, SNP Array and BSA-Seq**

Using mutants generated in the Vollbrecht lab, a total of eight mapping experiments were conducted using four different approaches. Three mutants were mapped using the BSA-GBS approach, namely \( rsl^*-12.2995 \), \( rsl^*-11.0243 \) and \( rel^*-11.0811 \). In addition, the \( knox^*-11.0227 \) mutant was mapped using BSA-seq, and \( rsl^*-12.2995 \) was mapped using WGS of a single mutant individual (Table 2.1). Moreover, a maize SNP array was used to map three mutants as well, namely \( knox^*-11.0227 \), \( lfy^*-16.0109 \) and a locus \( e1-mod \) that enhances the limited shoot phenotype of \( kn1-e1 \) (Vollbrecht, Reiser, & Hake, 2000). Two different mapping populations of the \( lfy^*-16.0109 \) mutant were mapped using a SNP array. The number of individuals in the bulked pools varied from 12 - 44, and number of individuals in the normal bulked pool varied from 25 - 106 (Table 2.1).

**BSA-GBS Sequence Analysis**

Two trials of BSA-GBS were performed, and each trial consisted of a single lane of Illumina sequencing. Multiple mutants were mapped using BSA-GBS in each trial (10-12 pools from 5-6 mutants). Only three mutants from Vollbrecht lab were mapped in both trials and the results from these three mutants have been presented here.

The first BSA-GBS trial consisted of a multiplexed library of 10 pools representing 5 mutants and was sequenced using the Illumina NextSeq 500. The sequencing run produced 115,325,746 single-end reads of 150-bp length. The library size (i.e., the number of sequencing reads for each pool) varied from ~7 million to ~14 million with an average ~10 million reads (Table 2.2). The proportion of reads retained after quality filtering ranged from ~81% to ~90%
with an average of ~86% (Table 2.2). The proportion of reads retained after alignment to the maize genome ranged from 48% to 71% with an average of ~57%. Genome coverage for the pools (i.e., proportion of the genome with a minimum of 10 reads aligned) varied among the pools, but all of them covered less than 1%. The genome coverage was also proportional to the library size. On the other hand, the proportion of aligned reads does not show such a correlation to the library size. Unbalanced libraries where either the mutant pool or normal pool had lower library size (e.g. Mutant-4 in Table 2.2) seems to lower the shared genome coverage. Interestingly, some of the mutants that had smaller but balanced library size have higher shared genome coverage (e.g., \textit{rsl*-12.2995} vs Mutant-8 in Table 2.3).

A second trial for BSA-GBS was used to map 6 mutants and was sequenced on an Illumina HiSeq 2500 machine. The sequencing run produced ~132 million 100-bp single-end reads (Table 2.3). The library size varied from ~4.6 million to 17 million with an average of ~10.8 million. The comparison of library size revealed that the mean library size was not significantly different between the two trials (Wilcoxon-test; P-value=0.8212). The proportion of reads passing quality filter ranged from ~84% to 94%, with an average of ~90%. Proportion of aligned reads of the library size ranged from ~51% to ~89% with ~71% on average. The proportion of the aligned reads is significantly higher for BSA-GBS trial 2 than trial 1 (Wilcoxon-test; P-value=0.0036), but this did not result in increased genome coverage for trial 2 (Wilcoxon-test; P-value=1).

**Loci mapped by BSA-GBS**

The MAPPR method was used to find putative, causal loci from BSA-GBS analysis (Hill et al., 2013). Mapping analysis from \textit{rsl*-12.2995}, \textit{rsl*-11.0243}, and \textit{rel*-11.0811} is presented here. MMAPPRL uses Euclidean distance (ED) between the pools of the nucleotide frequency at each position to identify the causal location. After $ED^4$ was calculated, uninformative markers
with low ED\(^4\) were filtered out. Manhattan plots of Raw ED\(^4\) for BSA-GBS trial 1 and trial 2 are shown in Figure 2.7. The \(rsl*-12.2995\) plot shows a signal on Chromosome 4, \(rsl*-11.0243\) shows a signal on chromosome 2, and \(rel*-11.0811\) shows two signals, one on chromosome 5 and one on chromosome 6. The signals for \(rsl*-12.2995\) and \(rsl*-11.0243\) are stronger than for \(rel*-11.0811\). Causal loci identified from the raw ED\(^4\) plots are large (e.g. Chromosome 4L for \(rsl*-11.2995\) and ~Chromosome 2 for \(rsl*-12.0243\)), so smoothed ED\(^4\) was plotted to identify the causal loci ((Figure 2.8). The causal loci windows were visually determined for \(rsl*-11.2995\) and \(rsl*-12.0243\) using the smoothed plot shows in Figure 2.9. The causal loci for both \(rsl*-11.2995\) and \(rsl*-12.0243\) were 50Mb wide. The causal locus for \(rsl*-12.0811\) was not as clear as for the other two mutants, and a 10 Mb region on chromosome 5 and 50 Mb region on chromosome 6 are potential causal loci.

A genomic-bin based method was used to narrow down the interval to decrease the time spent on subsequent fine-mapping. This simple method uses the ratio of markers that have ED\(^4\) >= 1 in each genomic region compared to the whole genome using Fisher's exact test. The p-values were adjusted for multiple testing, transformed and plotted in Figure 2.10.

The two BSA-GBS trials of \(rsl*-12.2995\) overlap the same position, and P-values were similar for both trials (Figure 2.11). The positions identified for \(rsl*-12.2995\) is fragmented, and spans 38Mb (155Mb - 193Mb) for Trial 1 and spans 32Mb (161-192Mb) for Trial 2 on Chr4 (Table 2.4). Signals on other chromosomes don’t show a peak signifying linkage (Chr2 and Chr8). The chromosomal location largely overlapped for both trials for \(rsl*-11.0243\), but the p-value was more significant in trial 1 (-10log10 (P-value) ~300) than trial 2 (-10log10 (P-value) ~80) (Figure 2.11). The region identified in Chr2 spans 36 Mb (20-56 Mb) in Trial 1 and 41 Mb (20-61 Mb) in Trial 2. There were less significant regions identified on Chr2 as well, but the
genetics suggested a single-locus recessive genotype. There were smaller regions of significance identified for chromosome 5 for \( \text{rel}^*\)-11.0811, but only in Trial 2 (27-30 Mb). A chromosome 6 signal for \( \text{rel}^*\)-11.0811 was significant for both trials but the locations did not overlap (Figure 2.11). The regions on Chr6 were a 2 Mb region (139-141 Mb) in Trial 1 and a 10 Mb region (124-134 Mb) in Trial 2 (Table 2.4). The P-values for the identified regions varied substantially between the two trials.

**Causal loci identified using SNP arrays and comparison with BSA-GBS**

Three mutants were mapped using SNP array, namely \( e1\)-mod*, \( knox^*\)-11.0227, and two mapping populations of \( lfy^*\)-16.0109. DNA from each pool was hybridized to single array, and analysis was performed by calculating the ratio of B-allele frequency between the mutant and the normal pools for each marker on the chip (Peiffer et al., 2006). Uninformative makers were filtered before the ratio was log transformed and plotted on Manhattan plots to visualize the causal locus. The results of the SNP array mapping are shown in Figure 2.12. The causal locus for \( e1\)-mod is on chromosome 6, the locus for \( knox^*\)-11.0227 was identified in chromosome 3, and for both \( lfy^*\)-16.0109 mapping populations the locus was in chromosome 3. The log ratio varied among the different mutants, and between the two different mapping populations of \( lfy^*\)-16.0109 mutant (Figure 2.12).

Comparison of the different approaches provides an opportunity to identify pros and cons of each technique for rough mapping maize mutants (Table 2.6). The SNP array uses microarray technology to genotype fixed markers across the genome, while BSA-GBS uses high-throughput sequencing to genotype random markers across the genome. SNP arrays require the less effort than BSA-GBS to generate the data, since a sequencing-based method, such as BSA-GBS, requires a library construction step. The maximum number of pools that can be mapped in a single trial varies between the methods, but for both mapping multiple mutants is necessary to
decrease the cost. The cost of both methods is comparable. The number of informative makers as defined in the methods were higher in SNP array-based analysis than BSA-GBS (Table 2.5), as SNP arrays had about 4450 markers on average which is higher than BSA-GBS which has about 2840 markers (Table 2.6).

**Causal locus mapped by BSA-seq**

The *knox*-11.0227 mutant was mapped using BSA-seq. The bulked-mutant pool for *knox*-11.0227 was sequenced using Illumina HiSeq to generate 50-Gb of 150-bp paired-end sequencing data. The reads were cleaned and aligned to B73 and Mo17 genomes. When mutant SNPs were called against B73, both Mo17 and non-Mo17 SNPS were identified. Similarly, mutant SNPs were called against the Mo17 genome and B73 and non-B73 SNPs were identified. The causal region was identified by plotting the number of homozygous Mo17 SNPs per genomic-bin (Figure 2.13). The causal locus for *knox*-11.0227 was identified on short arm of Chromosome 3 (Figure 2.13 and Figure 2.14), and fine-mapping efforts have narrowed it down to (Chr3 4,754,803 to 5,281,922 in B73). Close inspection of the *knox*-11.0227 causal locus revealed a SNP rich region (Figure 2.15). SNP effects on transcript models within the causal region were predicted. In B73 within the causal locus (527,120-bp, Chr3 4,754,803-5,281,922-bp) contained 53 transcript models that represent 23 gene models. Out of these gene models, only two gene models had SNP effects (Table 2.7). Genome alignments between B73 and Mo17 were used to identify the syntenic causal locus in Mo17 (1,762,001kb, Chr3 4337000 – 6099000-bp). The causal locus in B73 had to be extended to identify synteny, because the syntenic region in Mo17 contained genomic-rearrangements, and large indels. The SNP effects of non-B73 SNPs were predicted for transcript models within the expanded Mo17 causal region. The expanded syntenic region contained 78 transcript models representing 64 gene models. Out of the gene models within the region, only one gene model had a SNP effect (Table 2.7). The protein
sequence for the SNP-affected Mo17 gene model Zm00014a000575 was obtained and a BLASTP search was conducted against the SNP affected B73 gene models Zm00001d039453, Zm00001d039461, Zm00001d039461. The B73 ortholog was identified as Zm00001d039453, and mutation changes 62nd amino acid from glycine to asparagine in both B73 and Mo17 gene models.

Causal locus mapped by Whole Genome Sequencing of a mutant individual

The *rsl*-12.2995 mutant was mapped using WGS of a mutant individual. The DNA extracted from a single *rsl*-12.2995 mutant was sequenced by Novogene Inc., UC Davis, California using Illumina HiSeq to generate 50Gb of 150-bp paired-end sequencing data. The reads were cleaned and aligned to the B73 genome and the Mo17 genome. SNPs were called against the B73 genome and both Mo17 and non-Mo17 SNPs were identified. Homozygous Mo17 SNPs per genomic-bin was plotted to confirm that the regions identified by BSA-GBS and fine-mapping (Chr4 179,533,441 to 180,076,471-bp in B73) overlapped with WGS results. The Mo17 region of *rsl*-12.2995 covers most of Chromosome 4, and a small region in Chr 7 (Figure 2.13 and Figure 2.14). The Chr 7 region was ignored, because it is a fixed *ramosa1* location as expected by the underlying genetics. Upon close inspection the *rsl*-12.2995 causal locus lacked Mo17-B73 SNP diversity, and only smaller sub-regions showed diversity (Figure 2.15). Functional effects of the non-Mo17 mutant SNPs were predicted for B73 transcript models within the causal locus. The causal locus in B73 contained 71 transcript models that represent 9 gene models. Only one non-Mo17 SNP was found within the region, and thus only one gene model (Zm00001d052110) was affected (Table 2.7). The syntenic causal region in Mo17, identified using the genome alignments, was larger than B73 causal region (2,590,001-bp; Chr4 181,760,000 to 184,350,000), because of genome rearrangements and indel variations between B73 and Mo17. The interval contained 63 transcript models that represent 51 gene models. Of
these 51 gene models, only four gene models were predicted to be affected by SNPs (Table 2.7). The BLASTP search between the B73 gene model Zm00001d052110 and Mo17 gene models revealed that Zm00014a008561 was orthologous to Zm00001d052110. The amino acid change in Mo17 gene model Zm00014a008561 was 602th Ser to Leu, and both transcript models were affected by this. All 21 transcript models of B73 ortholog Zm00001d052110, were affected by SNPs, and 8 transcript models were affected by the same change 602th Ser to Leu seen in Mo17.

**Discussion**

A GBS method has been successfully adapted and optimized for bulked-segregant analysis. Maize inflorescence mutants have been successfully rough-mapped using the BSA-GBS method, and the causal loci have been confirmed by fine-mapping. BSA-GBS is comparable to the maize SNP array routinely used for BSA. In contrast, BSA-GBS does not use fixed markers like SNP arrays, but requires more effort than SNP arrays for generating the data. The causal loci identified by SNP array in this project have also been confirmed by fine-mapping. SNP arrays are more convenient than BSA-GBS to map mutants generated from the tested inbred lines. Mutants generated from other lines could benefit from the sequencing data generated by BSA-GBS. The sequencing data from BSA-GBS enables additional marker discovery to generate markers for fine-mapping. For example, BSA-GBS data from both rsl*-12.2995 and rsl*-11.0243 were used to generate CAPS markers for fine-mapping before the Mo17 genome was published. We also aligned the BSA-GBS reads to generate some INDEL markers. The Cost of BSA-GBS and SNP arrays were comparable at the time the BSA-GBS data was generated. The cost of SNP arrays has steadily decreased over time, as has the cost of sequencing. The BSA-GBS protocol could be optimized to increase the number of multiplexed mutants so that the cost of BSA-GBS would also decrease.
The MMAPPR data analysis method was chosen for BSA-GBS, because the MMAPPR method was tested on the noisy bulked-RNA-seq datasets (Hill et al., 2013). In case of RNA-seq the expression differences between the pools make it difficult to use traditional allele-based methods with confidence. The GBS protocol itself is known to introduce random noise due to the limitation of the molecular techniques used to generate the sequencing library (e.g. restriction digestion, adapter ligation). The three mutants \(rsl^*-12.2995\), \(rsl^*-11.0243\) and \(rel^*-11.0811\) analyzed in the study can be re-analyzed using a \(B\)-allele frequency now that both parental genomes (B73 and Mo17) are now published. This might improve the analysis, but this approach cannot be applied with confidence if the parental genomes are not available. The analysis to identify genomic-bins that were enriched for positions that had \(ED^d \geq 1\) using Fisher’s exact test did narrow the regions down, but it also identified several false-positive regions. The false positives seem to stem from the assumption that SNP density is random and uniform across the genome. This assumption is violated in some regions. The method was successful in \(rsl^*-11.0243\) where the causal region was SNP rich, and failed in \(rsl^*-12.2995\) for which the causal region is SNP poor (Figure 2.15). The method could be potentially improved by using nested enrichment testing that uses shrinking sliding window width. Irrespective of what statistical method is used with the rough-mapping datasets, considerable time will be required for fine-mapping and identifying the causal mutation without WGS approaches.

BSA-seq has already been used to map mutants in several species including maize, rice and \textit{Arabidopsis}. BSA-seq was used in this project to identify putative causative SNPs for EMS derived mutations. The bioinformatics analysis pipeline to analyze the BSA-seq data did not require complex analysis methods. Instead, the analysis focused on making effective use of the public data to reduce number of false-positive SNPs. The use of B73 and Mo17 genomes
reduced the number of false-positive SNPs considerably. Manual filtering of the SNPs would have been challenging without both genome sequences. Even in the absence of a de novo assembled genome, a comparable dataset is essential to filter out non-causal variants. A contrasting dataset could be generated from the normal pool or a high-depth sequencing of parental inbred lines. Cloning the gene by sequencing of a mutant individual for a rough-mapped mutant was also successful. There were a substantial number of SNP calls within the rough mapped location and prioritizing the number of SNPs with effects on transcripts without fine-mapping would have been time consuming. The BSA-seq takes less time than rough-mapping combined with WGS approach for positionally cloning the gene.

Forward genetics in maize has immensely benefited from lowering sequencing costs and the availability of genome sequences of various inbred lines. Even in the age of low-cost sequencing, the generation of mapping populations and bulked-segregants remains an efficient way to identify the causal mutation. Once a mapping population is generated, BSA-seq will get to the list of putative causal mutations faster than the less expensive SNP array or BSA-GBS approaches. There are cases when large numbers of mutant families are generated using genetic screens, such as enhancer-suppressor screens or EMS screens. Use of rough mapping methods would allow researchers to screen a large number of mutant families for low cost and prioritize which mutant families should be fine-mapped.

The decision of which strategy to use for BSA and cloning a gene will depend on the importance of the mutant, time constraint, budget and the nature of the specific mutant. BSA-seq is the fastest way to map if budget is not the limitation. Rough-mapping using SNP arrays or BSA-GBS can be used in conjunction with marker based fine-mapping efforts if the budget is limited. The fine-mapping process is smoother if the mutant is generated using one of the inbred
lines for which a high-quality genome assembly is available. The cost also depends on the pools that should be sequenced, and which pools need to be sequenced depends on the inbred lines used to generate the mutants. Irrespective of how the mutant was mapped and the gene was cloned, further studies are necessary for the confirmation and characterization of the mutant.

**Author Contributions**

EV generated the mapping populations and phenotyped the mutants with RW. RW scored mutants and generated the bulked pools. EUW extracted HMW DNA and fine-mapped mutants. KW developed the BSA-GBS bench protocol and the bioinformatics pipeline to analyze BSA-GBS, BSA-seq, and WGS sequencing data. EUW, RW and EV contributed to the experimental design and supervised the work. KW prepared the first draft of the manuscript, EUW and EV edited the manuscript. All authors read and approved the final manuscript.

**Acknowledgements**

We thank Longui Ren for extracting the DNA from bulked-segregant pools, and Hannes Claeys and David Jackson for helping with the sequencing of first BSA-GBS trial. Thanks to other Vollbrecht lab members who have helped with fine-mapping the mutants. This work was supported by funding from the National Science Foundation [PGRP, IOS # 1238202] to EV; and the PSI graduate fellowship to KW.
**Tables and Legends**

Table 2.1: Mutants Mapped using SNP-array, BSA-GBS and BSA-seq

<table>
<thead>
<tr>
<th>Mutant</th>
<th>Mutant Inbred</th>
<th># Mutant Plants</th>
<th># Normal Plants</th>
<th>Mapping Method(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>rel*-12.2995</td>
<td>B73</td>
<td>12</td>
<td>25</td>
<td>BSA-GBS, WGS</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>B73</td>
<td>25</td>
<td>25</td>
<td>BSA-GBS</td>
</tr>
<tr>
<td>rel*-11.0811</td>
<td>B73</td>
<td>19</td>
<td>25</td>
<td>BSA-GBS</td>
</tr>
<tr>
<td>knox*-11.0227</td>
<td>B73</td>
<td>21</td>
<td>62</td>
<td>SNP-Chip, BSA-seq</td>
</tr>
<tr>
<td>lfy*-16.0109 (P1)</td>
<td>W22</td>
<td>36</td>
<td>102</td>
<td>SNP-Chip</td>
</tr>
<tr>
<td>lfy*-16.0109 (P2)</td>
<td>W22</td>
<td>44</td>
<td>106</td>
<td>SNP-Chip</td>
</tr>
<tr>
<td>el-mod</td>
<td>W22</td>
<td>212</td>
<td>455</td>
<td>SNP-Chip</td>
</tr>
</tbody>
</table>
Table 2.2: Sequencing stats for mutants mapped with BSA-GBS Trial 1 with Illumina NextSeq 500

<table>
<thead>
<tr>
<th>Mutant</th>
<th>Normal Pool</th>
<th>Mutant Pool</th>
<th>Normal Pool</th>
<th>Mutant Pool</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># of Raw Reads</td>
<td>Reads passing QF (%)</td>
<td>Aligned Reads (%)</td>
<td>Genome coverage (%)</td>
</tr>
<tr>
<td>rsl*-12.2995</td>
<td>9,329,073</td>
<td>86.69</td>
<td>48.54</td>
<td>0.35</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>9,323,199</td>
<td>88.50</td>
<td>48.52</td>
<td>0.35</td>
</tr>
<tr>
<td>rel*-11.0811</td>
<td>12,573,072</td>
<td>87.11</td>
<td>55.69</td>
<td>0.58</td>
</tr>
<tr>
<td>Mutant-4</td>
<td>7,073,332</td>
<td>82.76</td>
<td>59.07</td>
<td>0.21</td>
</tr>
<tr>
<td>Mutant-5</td>
<td>10,641,235</td>
<td>81.20</td>
<td>61.28</td>
<td>0.47</td>
</tr>
</tbody>
</table>

Table 2.3: Sequencing stats for mutants mapped with BSA-GBS Trial 2 with Illumina HiSeq 2500

<table>
<thead>
<tr>
<th>Mutant</th>
<th>Normal Pool</th>
<th>Mutant Pool</th>
<th>Normal Pool</th>
<th>Mutant Pool</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># of Raw Reads</td>
<td>Reads passing QF (%)</td>
<td>Aligned Reads (%)</td>
<td>Genome coverage (%)</td>
</tr>
<tr>
<td>rsl*-12.2995</td>
<td>14,261,864</td>
<td>84.65</td>
<td>51.05</td>
<td>0.47</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>11,525,540</td>
<td>89.21</td>
<td>56.06</td>
<td>0.45</td>
</tr>
<tr>
<td>rel*-11.0811</td>
<td>17,514,330</td>
<td>85.16</td>
<td>60.93</td>
<td>0.67</td>
</tr>
<tr>
<td>Mutant-6</td>
<td>9,080,656</td>
<td>94.27</td>
<td>69.29</td>
<td>0.45</td>
</tr>
<tr>
<td>Mutant-7</td>
<td>6,176,445</td>
<td>93.84</td>
<td>89.30</td>
<td>0.33</td>
</tr>
<tr>
<td>Mutant-8</td>
<td>12,476,544</td>
<td>94.50</td>
<td>84.82</td>
<td>0.69</td>
</tr>
</tbody>
</table>
Table 2.4: Putative genomic regions identified to contain the causal locus by genome-bin based ED^4 Enrichment for BSA-GBS

<table>
<thead>
<tr>
<th>Mutant</th>
<th>Trial</th>
<th>Chromosome</th>
<th>Start (Mb)</th>
<th>End (Mb)</th>
<th>Width (Mb)</th>
<th>Count of ED^4&gt;1</th>
</tr>
</thead>
<tbody>
<tr>
<td>rsl*-12.2995</td>
<td>trial1</td>
<td>Chr4</td>
<td>155</td>
<td>157</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>rsl*-12.2995</td>
<td>trial1</td>
<td>Chr4</td>
<td>161</td>
<td>164</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>rsl*-12.2995</td>
<td>trial1</td>
<td>Chr4</td>
<td>171</td>
<td>174</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>rsl*-12.2995</td>
<td>trial1</td>
<td>Chr4</td>
<td>175</td>
<td>178</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>rsl*-12.2995</td>
<td>trial1</td>
<td>Chr4</td>
<td>183</td>
<td>186</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>rsl*-12.2995</td>
<td>trial1</td>
<td>Chr4</td>
<td>187</td>
<td>190</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>rsl*-12.2995</td>
<td>trial1</td>
<td>Chr4</td>
<td>191</td>
<td>193</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial1</td>
<td>Chr2</td>
<td>20</td>
<td>25</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial1</td>
<td>Chr2</td>
<td>27</td>
<td>32</td>
<td>5</td>
<td>45</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial1</td>
<td>Chr2</td>
<td>33</td>
<td>44</td>
<td>11</td>
<td>74</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial1</td>
<td>Chr2</td>
<td>53</td>
<td>56</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial1</td>
<td>Chr2</td>
<td>120</td>
<td>125</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial1</td>
<td>Chr2</td>
<td>129</td>
<td>134</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial1</td>
<td>Chr2</td>
<td>182</td>
<td>184</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>rel*-11.0811</td>
<td>trial1</td>
<td>Chr6</td>
<td>139</td>
<td>141</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>rsl*-12.2995</td>
<td>trial2</td>
<td>Chr2</td>
<td>69</td>
<td>72</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>rsl*-12.2995</td>
<td>trial2</td>
<td>Chr4</td>
<td>161</td>
<td>164</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>rsl*-12.2995</td>
<td>trial2</td>
<td>Chr4</td>
<td>181</td>
<td>184</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>rsl*-12.2995</td>
<td>trial2</td>
<td>Chr4</td>
<td>187</td>
<td>193</td>
<td>6</td>
<td>28</td>
</tr>
<tr>
<td>rsl*-12.2995</td>
<td>trial2</td>
<td>Chr8</td>
<td>31</td>
<td>34</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial2</td>
<td>Chr2</td>
<td>20</td>
<td>25</td>
<td>5</td>
<td>23</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial2</td>
<td>Chr2</td>
<td>27</td>
<td>37</td>
<td>10</td>
<td>44</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial2</td>
<td>Chr2</td>
<td>38</td>
<td>45</td>
<td>7</td>
<td>28</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial2</td>
<td>Chr2</td>
<td>49</td>
<td>51</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial2</td>
<td>Chr2</td>
<td>52</td>
<td>54</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial2</td>
<td>Chr2</td>
<td>58</td>
<td>61</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial2</td>
<td>Chr2</td>
<td>108</td>
<td>110</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>rsl*-11.0243</td>
<td>trial2</td>
<td>Chr2</td>
<td>155</td>
<td>158</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>rel*-11.0811</td>
<td>trial2</td>
<td>Chr5</td>
<td>27</td>
<td>30</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>rel*-11.0811</td>
<td>trial2</td>
<td>Chr6</td>
<td>124</td>
<td>126</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>rel*-11.0811</td>
<td>trial2</td>
<td>Chr6</td>
<td>131</td>
<td>134</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>
Table 2.5 Number of informative makers that was used for analysis in BSA-GBS and SNP arrays

<table>
<thead>
<tr>
<th>Mutant</th>
<th>Trial</th>
<th>BSA Method</th>
<th># of Informative Markers</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>rsl</em>-12.2995</td>
<td>Trial1</td>
<td>BSA-GBS</td>
<td>2,897</td>
</tr>
<tr>
<td><em>rsl</em>-11.0243</td>
<td>Trial1</td>
<td>BSA-GBS</td>
<td>2,391</td>
</tr>
<tr>
<td><em>rel</em>-11.0811</td>
<td>Trial1</td>
<td>BSA-GBS</td>
<td>3,838</td>
</tr>
<tr>
<td><em>rsl</em>-12.2995</td>
<td>Trial2</td>
<td>BSA-GBS</td>
<td>2,589</td>
</tr>
<tr>
<td><em>rsl</em>-11.0243</td>
<td>Trial2</td>
<td>BSA-GBS</td>
<td>1,987</td>
</tr>
<tr>
<td><em>rel</em>-11.0811</td>
<td>Trial2</td>
<td>BSA-GBS</td>
<td>3,349</td>
</tr>
<tr>
<td><em>lfy</em>-16.0109 (P1)</td>
<td>N/A</td>
<td>SNP Array</td>
<td>5,748</td>
</tr>
<tr>
<td><em>lfy</em>-16.0109 (P2)</td>
<td>N/A</td>
<td>SNP Array</td>
<td>4,334</td>
</tr>
<tr>
<td><em>knox</em>-11.0227</td>
<td>N/A</td>
<td>SNP Array</td>
<td>5,377</td>
</tr>
<tr>
<td><em>e1-mod</em></td>
<td>N/A</td>
<td>SNP Array</td>
<td>2,367</td>
</tr>
</tbody>
</table>

Table 2.6: Comparison between SNP array and BSA-GBS

<table>
<thead>
<tr>
<th>Protocol</th>
<th>SNP Array</th>
<th>BSA-GBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology Used</td>
<td>Microarray</td>
<td>HTS</td>
</tr>
<tr>
<td>Fixed Markers</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Library Prep</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Prep Time</td>
<td>1 day</td>
<td>2 days</td>
</tr>
<tr>
<td>Number of Pools</td>
<td>20</td>
<td>12-16</td>
</tr>
<tr>
<td>Avg # of Informative Markers</td>
<td>4456</td>
<td>2842</td>
</tr>
<tr>
<td>Cost per Mutant ($USD)</td>
<td>250</td>
<td>~250</td>
</tr>
</tbody>
</table>
Table 2.7: Genes with predicted SNP effects within the regions identified by fine-mapping

<table>
<thead>
<tr>
<th>Mutant</th>
<th>Chr</th>
<th>Pos</th>
<th>Gene</th>
<th>Chr</th>
<th>Pos</th>
<th>Gene</th>
</tr>
</thead>
<tbody>
<tr>
<td>rsl*-12.2995</td>
<td>Chr4</td>
<td>179,918,584</td>
<td>Zm00001d052110</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>182,763,799</td>
<td>Zm00014a008571</td>
</tr>
<tr>
<td></td>
<td></td>
<td>183,390,136</td>
<td></td>
<td></td>
<td>183,177,895</td>
<td>Zm00014a008561</td>
</tr>
<tr>
<td></td>
<td></td>
<td>183,390,425</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>183,547,440</td>
<td></td>
<td></td>
<td>183,547,442</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>183,547,457</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>183,547,651</td>
<td></td>
<td></td>
<td>183,548,265</td>
<td></td>
</tr>
<tr>
<td>knox*-11.0227</td>
<td>Chr3</td>
<td>4,751,870</td>
<td>Zm00001d039453</td>
<td></td>
<td>5,831,030</td>
<td>Zm00014a000575</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4,956,503</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4,956,647</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4,956,657</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4,956,811</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4,956,830</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4,956,849</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4,956,923</td>
<td>Zm00001d039461</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 2.1: Steps to generate an F2 mapping population is shown here. The mutant that has been generated in a Mo17 inbred line, is crossed onto a normal (wild-type) plant from a B73 inbred line. Only the chromosome that has the causal mutation has been shown in the illustration. Mo17 is shown in tan color, B73 in blue color and causal locus in pink color. The F1 hybrid is selfed to generate F2 population. 2 phenotypes (mutant:normal 1:3) and 3 genotypes (mutant:heterozygous:wild-type, 1:2:1) are seen at the causal locus.
Figure 2.2: The inheritance patterns of the markers for the F2 populations associated with the causal locus (L/l), for a recessive mutant (l). Second location contains a co-dominant marker which can be used to identify whether the location is B73 (B) or Mo17 (M).

<table>
<thead>
<tr>
<th></th>
<th>Complete Linkage</th>
<th>Unlinked Marker</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Normal Phenotype</strong></td>
<td>L-B, L-M</td>
<td>L, B</td>
</tr>
<tr>
<td></td>
<td>L-B, M</td>
<td>L, M</td>
</tr>
<tr>
<td>B:M=2:1</td>
<td></td>
<td>B:M=1:1</td>
</tr>
</tbody>
</table>

| **Mutant Phenotype** | L-M, L-M         | 1, B, B, M, M   |
|                     | 1, B, B, 1, B, B, M |
| B:M=0:2             |                  | B:M=1:1         |
Figure 2.3: Bulked-segregant analysis using genotyping-by-sequencing protocol and trial setup
(a) The left side gives the main steps for the BSA-GBS protocol as described in (Elshire et al. 2011), and the right side illustrates fragmentation of the DNA, ligation of the GBS adapters (partial Illumina adapters) to the restriction digested DNA fragment, and the addition of the rest of the Illumina Adapter by PCR. A color legend is given in bottom of the panel. (b) Typical BSA-GBS trial setup. Each pool is subjected to the BSA-GBS protocol leading to a barcoded library, which are then pooled to make a sequencing library, which is sequenced and analyzed.
Figure 2.4: BioAnalyzer Traces from non-clean library vs bead-cleaned library
The BioAnalyzer 2100 high-sensitivity DNA chip results from high two barcoded library samples. (a) A barcoded library which was not cleaned using SPRI beads. Note the small peak at 127-bp. (b) A barcoded library which was cleaned using SPRI beads. Note the absence of the peak at 127-bp.

Figure 2.5: BSA-GBS Analysis Pipeline
The steps of the BSA-GBS analysis pipeline for two sample barcodes of mutant and normal bulked pools for an example mutant. Analysis starts with demultiplexing of the reads into the correct pools and follows through cleaning and quality control of the reads. Next, the alignment of the anchors reads to the genome. The distance is calculated between pools and the results are plotted on Manhattan plots.
Figure 2.6 BSA-seq Analysis Pipeline

The pipeline used for the analysis of BSA-seq sequencing data. The steps outlined here are for the analysis using the B73 genome. Similar steps were performed using the Mo17 genome for alignment and SNP calling as well, and non-B73 putative candidate SNPs were identified in the syntenic region in Mo17.
Figure 2.7: Raw ED^4 plots for mutants mapped using BSA-GBS
ED^4 values were calculated for each position that had at least 10 reads aligned in both normal and mutant bulked-pools. Only positions with ED^4 > 0.05 have been potted. The 10 chromosomes are ordered and every other chromosome is alternatively colored. Three mutants mapped with BSA-GBS are plotted, 12.2995=rsli*-12.2995, 11.0243=rsli*-11.2043, and 11.0811=rel*-11.0811. Results from Trial 1 and Trial 2 are plotted next to each other for each mutant in the vertical direction.
Figure 2.8: Smoothed ED⁴ plots for mutants mapped using BSA-GBS
Only positions with ED4 > 0.05 were plotted here after smoothing by LOESS method with a span of 0.2. The 10 chromosomes are ordered and every other chromosome is alternatively colored. Three mutants mapped with BSA-GBS are plotted, 12.2995=rsl*-12.2995, 11.0243=rsl*-11.2043, and 11.0811=rel*-11.0811. Results from Trial 1 and Trial 2 are plotted next to each other for each mutant in the vertical direction.
Figure 2.9: Smoothed ED4 plots for the causal loci of mutants mapped using BSA-GBS
The chromosomes the putative causal locus detected are plotted for the three mutants mapped with BSA-GBS. Chr4 for 12.2995=rsl*-12.2995, Chr2 for 11.0243=rsl*-11.2043, and Chr5 and Chr6 for 11.0811=rel*-11.0811. Results from Trial 1 and Trial 2 are plotted next to each other for each mutant in the vertical direction.
Figure 2.10: Genomic-bin based ED^4 count enrichment for the mutants mapped using BSA-GBS
Number of positions with ED4 > 1 for each 2 Mb genomic-bin, sliding window every 1 Mb. Fisher’s exact test was performed on the ratio of number of positions with ED4=>1 vs <1 within the genomic-bin and the whole genome. The p-value for the test was transformed and plotted on a Manhattan plot. The 10 chromosomes are ordered, and every other chromosome is alternatively colored. Three mutants mapped with BSA-GBS are plotted, 12.2995=rsl*-12.2995, 11.0243=rsl*-11.2043, and 11.0811=rel*-11.0811. Results from Trial 1 and Trial 2 are plotted next to each other for each mutant in the vertical direction.
Figure 2.11: Genomic-bin based $ED^4$ enrichment for causal in the causal loci of the mutants mapped using BSA-GBS. Number of positions with $ED^4 > 1$ for each 2 Mb genomic-bin, sliding window every 1 Mb. Fisher’s exact test was performed on the ratio of number of positions with $ED^4 >= 1$ vs <1 within the genomic-bin and the whole genome. The p-value for the test was transformed and plotted on a Manhattan plot. The chromosomes where the causal locus was detected are plotted for the three mutants mapped with BSA-GBS. Chr4 for 12.2995=rsl*-12.2995, Chr2 for 11.0243=rsl*-11.2043, and Chr5 and Chr6 for 11.0811=rel*-11.0811. Results from Trial 1 and Trial 2 are plotted next to each other for each mutant in the vertical direction.
Figure 2.12: Manhattan plots for mutants mapped with the SNP array

The absolute value of B-allele frequency ratio of mutant:normal is log2 transformed and plotted on a Manhattan Plot. $e_1 = e1-mod$, $knox = knox*-11.0227$, $lfy1 = lfy*-16.0109$ mapping population 1, and $lfy2 = lfy*-16.0109$ mapping population 2. The 10 chromosomes are ordered and every other chromosome is alternatively colored.
Figure 2.13: Number of homozygous Mo17 SNPs per 2Mb region for \textit{knox}*-11.0227 and \textit{rsl}*-12.2995
The 10 chromosomes are ordered and every other chromosome is alternatively colored.
Figure 2.14: Number of Mo17 SNPs per 2Mb region on the chromosome with the putative causal locus for \textit{knox*-11.0227} and \textit{rsl*-12.2995}
Figure 2.15: Number of Mo17 SNPs per 250 kb region surrounding putative causal locus for $knox^{*-11.0227}$ (blue) and $rl^{*-12.2995}$ (red).
References


CHAPTER 3. IDENTIFICATION OF GENOMIC T-DNA INSERTION LOCATIONS USING FLANKING GENOMIC SEQUENCES IN TRANSGENIC MAIZE

Wimalanathan Kokulapalan *, Erica Unger-Wallace* and Erik Vollbrecht

Abstract

Transgenic experiments in maize have been invaluable not only for the improvement of agricultural traits, but also for careful investigation to determine the mechanisms of gene function. During a typical transgenic maize experiment, transfer-DNA (T-DNA) is transformed using Agrobacterium into maize, and the T-DNA integrates in the maize genome in a non-specific integration location. Identifying the precise T-DNA location is essential for biological studies using the transgenic plants, such as investigating expression patterns of promoters, or large-scale T-DNA based mutagenesis projects. Traditional molecular methods that use inverse-PCR and tail-PCR methods are time consuming to identify multiple insertion locations at the same time. More recent methods use short-read sequencing to sequence multiplexed pools of enriched genomic T-DNA flanks in other plant species. The use of short-read based methods limits the accuracy of placing insertions in maize, with its large repetitive genome. We developed an optimized protocol to enrich T-DNA flanks in transgenic maize and used long read sequencing to sequence the flanks. We identified T-DNA insertion locations using flanking sequences, and successfully identified insertions for 16 of 20 transgenic events using the optimized protocol and bioinformatics analysis. We confirmed a majority of the insertions using PCR and using simulated reads evaluated whether placing the insertions benefited from the longer reads.
Introduction

Transgenic technology has been invaluable in maize for the understanding of gene function and crop improvement. Transgenics in maize have been routinely used to understand the precise function of maize genes from characterized maize mutants. Transgenes have been used to study various aspect of gene function, using techniques such as florescent tagging, gene silencing, immunohistostaining, ChIP-seq, and more recently gene editing using TALENS and CRISPR (Char et al., 2015; Lappe et al., 2018; Mei & Whitham, 2018; Pautler et al., 2015; Q. Wu, Regan, Furukawa, & Jackson, 2018).

A breakthrough in transformation of maize that allowed the highly-efficient transformation of maize embryos by Agrobacterium, facilitated wide-spread adoption of transgenics in maize (Ishida et al., 1996). Agrobacterium tumefaciens based transformation begins with the construction of a plasmid vector that contains one or more T-DNA fragments (Singh & Prasad, 2016). Each T-DNA may contain one or more genes, including the promoter and terminator sequences for each gene in the construct. One of the genes in the T-DNA fragments is the gene of interest (GOI), and generally one more marker gene is co-transformed to select successful transformation events. An example of a T-DNA construct is given in Figure 3.1, where the RFP is the GOI and BAR is the selectable marker gene. The BAR gene confers resistance to the herbicide bialaphos, which is used to select successful transformants (Thompson et al., 1987). The T-DNA is flanked by nearly identical 25-bp repeat Left border (LB) and right border (RB) sequences (Gelvin, 2017). During the transformation process LB and RB will guide the integration of T-DNA into the host genome (Yadav, Vanderleyden, Bennett, Barnes, & Chilton, 1982). The maize embryos are transformed by treatment with the Agrobacterium cells containing the T-DNA containing vector.
During the transformation process, the T-DNA travels from the bacterial cell to cells in the maize embryo and is inserted into the maize genome. The process of T-DNA integration into the host genome is imprecise. The copy number of integrations can vary, and frequently the plasmid vector backbone is also integrated in one or more locations across the genome (Krizkova & Hrouda, 1998; Meza et al., 2002; Oltmanns et al., 2010). Often, the T-DNA integration is incomplete and doesn’t contain the entire T-DNA fragment.

In many transgenic studies, the goal is to obtain stable expressing transformants, and location of T-DNA integration is not determined. However, several studies have indicated that integration location affects the expression of the transgene (Day et al., 2000; Gelvin, 2017; Meyer, 1995; Stam, Viterbo, Mol, & Kooter, 1998). Moreover, identification of T-DNA integration location is imperative for large scale T-DNA based mutagenesis, such as those done in Arabidopsis thaliana, and Brachypodium distachyon (Hsia et al., 2017; O’Malley, Alonso, Kim, Leisse, & Ecker, 2007). For more complex studies that involve, for example, dissection of promoter functions using transgenes, events where the T-DNA integrated in a genomic location unaffected by silencing factors such as methylation or inverted repeats would be essential (Fultz & Slotkin, 2017).

Several different molecular approaches have been developed to identify the genomic integration location of the T-DNAs in plants. Tail-PCR and inverse-PCR methods have been used to amplify the genomic DNA fragment flanking the T-DNA insertion (commonly known as T-DNA flanks), and the amplified DNAs have been sequenced using Sanger sequencing for smaller projects (Stefano, Patrizia, Matteo, & Massimo, 2016; L. Wu et al., 2015). For large-scale mutagenesis projects in plants optimized tail-PCR based methods were used to generate high-throughput sequencing libraries to identify genomic insertion location of T-DNAs. Methods
developed by O’Malley et al have been used to characterize the *A. thaliana* T-DNA mutagenesis collection, and more recently the *B. distachyon* T-DNA collection (Hsia et al., 2017; O’Malley et al., 2007). This method uses the Illumina platform to sequence the T-DNA flanks and then the integration location is identified by placing the flanks on the genome. The use of Illumina for the sequencing of T-DNA flanks in smaller genomes such as *A. thaliana* (~250-Mb) and *B. distachyon* (270-Mb) has been successful (Tello-Ruiz et al., 2017). Placing the T-DNA flanking sequences in a larger and more repetitive maize genome could be limited by the short reads generated by Illumina sequencing. Development of long read sequencing technologies such as PacBio and Oxford Nanopore gives the opportunity to generate longer reads and thereby give a better chance to place the T-DNA flanks in larger, repetitive genomes such as maize and wheat (International Wheat Genome Sequencing Consortium (IWGSC) et al., 2018; Jiao et al., 2017).

The main goals of the project were to develop a bench protocol to enrich T-DNA flanks, generate long-read sequencing libraries from the enriched flanks, and design a bioinformatics pipeline to analyze the sequencing data to accurately identify the T-DNA insertion location. Here we present a protocol developed to enrich and sequence T-DNA flanks and its application to 20 transgenic maize events, and the bioinformatic analysis of the sequenced flanks to accurately identify single-bp genomic insertion locations for these events.

**Methods**

**Transgenic Events and Pooling Setup**

Three experiments, namely LB1, LB2 and RB2, were used to map twenty transgenic events from the transformation of a red fluorescent protein (RFP)- containing T-DNA construct (Table 3.1). Transgenic events were first arrayed into in a 2D grid format of pools shown in Table 3.1. The 20 transgenic events were pooled using 4 columns (C1-C4) and 5 rows (R1-R5) for the first experiment (LB1). The 707.26.1 event was treated as an independent pool in LB2
and RB2 experiments. Each row contained leaf tissue punches from 4 events, and each column contained punches from 5 events. Each pool of tissue contained a total of 300 leaf punches collected equally from among 4 to 5 transgenic events (APPENDIX C). High-molecular weight (HMW) genomic DNA was prepared from the frozen leaf punches for each pool using a Urea-based protocol described in APPENDIX A, which was a modification of a previously described procedure. The extracted DNA was quantified and used for the T-DNA flank enrichment protocol. The flank enrichment protocol was performed on each DNA pool. The T-DNA flanks from each pool were barcoded during the flank enrichment protocol (Figure 3.2) and a PacBio sequencing library was constructed after the enriched flanks from the pools were normalized and combined. The sequencing was performed using PacBio RS II system at the Iowa State University DNA facility.

**Flank Enrichment Protocol**

The flank enrichment protocol starts with the shearing of the genomic DNA extracted from the pools. The steps of the flank enrichment protocol are illustrated in Figure 3.3, and the detailed steps are given in APPENDIX C. The sheared DNA fragments are end-repaired and dA-tailed in a single step. The reaction is cleaned up using SPRI beads that are DNA binding paramagnetic beads. Next, the generic adapter is ligated to the repaired DNA fragments (APPENDIX C). The next step is PCR designed to enrich for the flanks from the genomic DNA. Different enrichment strategies were used for LB1, LB2 and RB2 (Table 3.2). In LB1, linear amplification of the T-DNA flanks was performed by including only the biotinylated left border (bLB) primer in PCR. In Trial 2, bLB was used in conjunction with a non-biotinylated adapter primer (AP) for low-cycle (15 cycles) polymerase chain reaction (PCR). The same step was performed with the biotinylated right border (bRB) primer and AP for the RB2 experiment. The PCR reactions were cleaned using streptavidin beads to enrich for biotinylated amplicons.
Barcodes were added to the enriched T-DNA flanks by nested PCR with barcoded primers. The barcoded LB (bc-LB) primer was used with the barcoded AP (bc-AP) in LB1. The bc-LB and bc-AP were used to add barcodes and amplify bLB enriched flanks for LB2 experiment, and bc-RB and bc-AP primers were used to barcode and amplify the bRB enriched flanks for RB2 experiment. The sequences for the primers and the barcodes used are presented in APPENDIX C.

The enriched flanks from the pools were quantified and the fragment size was characterized using a BioAnalyzer. Barcoded flanks from different pools were normalized using the concentration and size and multiplexed. A single PacBio sequencing library was constructed for each experiment using the combined barcoded fragments following SMRTBell™ template preparation instructions. The constructed library was quantified, and the size distribution was characterized. From each experiment multiplexed libraries were sequenced on a single SMRT cell on the PacBio RS II system (Iowa State University, DNA Facility).

**Sequence Analysis**

The sequences from the PacBio RS II system are obtained as raw h5 files (Figure 3.4). The updated analysis tools such as ccs and lima do not work with the older h5 format, so h5 files were converted to the newer PacBio bam format. The PacBio bam format contains the subreads from each PacBio zero-mode waveguide (ZMW), and the subreads are multiple sequencing runs of each sequencing template. High base call error rate of the subreads decreases demultiplexing accuracy, so circular consensus sequences (CCS) were constructed using PacBio ccs tool (Weirather et al., 2017). The barcode for each read was identified using the PacBio lima tool ($BC_{score} \geq 40$), but the reads were not demultiplexed.

The CCS reads were converted to fastq and aligned to three different reference sequences using minimap2 (H. Li, 2018). minimap2 has automated clipping, so clipping was not necessary.
The reads were aligned to the maize RefGen_v4 genome, barcoded adapter sequences (bc-AP), and barcoded T-DNA specific border sequences (bc-LB or bc-RB) (Figure 3.4). The minimap2 “sr” preset was used for the alignment of the reads to the barcoded adapter sequences, and barcoded border sequences. The minimap2 asm5 preset was used to align the CCS reads to the maize genome.

The alignments were subjected to quality filters that were specific to the aligned reference sequence. The filtering criteria for different reference sequences are listed in Table 3.3. The alignment from the adapter, border and the genome were combined, and only intact fragments with unique hits to the reference sequences (genome, adapter and border) were retained. The barcode identified by lima was used to group the intact T-DNA flanks into the correct pools. Alignment to the border (bc-LB and bc-RB) and adapter (bc-AP) sequences were used as an additional QC step to verify the barcode assigned by lima. The genomic alignments of the intact T-DNA flanks were analyzed to identify putative the putative T-DNA insertion locations for each pool.

**Identification of putative insertion locations**

Genomic alignment of the intact T-DNA flanks from each pool was analyzed independently to identify putative T-DNA locations. The steps to identify the insertion locations for each pool is as follows. Contiguous genomic regions with a minimum read depth of 5 were identified for each pool. The number of reads aligned within each contiguous genomic region was counted ($RC_{region}$). The read depth (DEP) at each nucleotide position (POS) within a genomic region was calculated, and Pearson correlation coefficient ($r$) was calculated for variables DEP and POS ($r_{DEP,POS}$). The correlation coefficient was used to identify the exact single-bp genomic insertion location and whether the T-DNA border was upstream or
downstream of the genomic position. The number of reads for each location was used to filter for background noise. The putative insertion locations with a $\rho_{DEP,POS} \geq 0.8$ and $RC_{region} \geq 15$ were retained for further analysis. Putative T-DNA locations from different pools were combined to identify insertion locations shared by intersecting pools in the two-dimensional pooling scheme. The insertion locations shared by multiple pools were manually inspected using IGV and confirmed using PCR experiments. The confirmed locations were assigned to individual transgenic events.

**Analysis of simulated read lengths at the T-DNA insertion location**

The 20 insertion locations (3-22) that were identified from the three experiments were selected for the simulation of reads. A total of 4 read lengths ($Len_R$) were simulated at each location. The two longest sizes of Illumina (500-bp and 600-bp), and two other longer-read read sizes (1200-bp, and 2000-bp) were simulated for comparison. Each location was simulated as a LB insertion. The average LB length ($Len_{LB}$) in the intact flanks from the experiment is ~200-bp, and the adapter length ($Len_A$) is 84-bp. The length of the LB and Adapter were subtracted from the simulated read length (RL) to account for border and adapter sequences, to determine the size of T-DNA flank ($Len_T$). The potential minimum length of the Illumina reads was set at 50-bp, and of the long-reads at 200-bp, to simulate real world alignment conditions. The min and max sizes of the simulated T-DNA flanks were set to 50 and ($Len_R - 284$) for Illumina, and 200 to($Len_R - 284$) for long-reads. T-DNA flank lengths were randomly chosen between the $min(Len_F)$ and $max(Len_F)$, the genomic sequence was obtained from the genome sequence using the coordinates calculated based on the insertion position and $Len_F$. This random sampling was performed 1500 times for each $Len_R$ and the 20 insertion locations (Peak ID 2 to 22) from previous results. Each extracted DNA sequence was randomly mutated at a mutation rate of 2%
to add noise to the sequence, that would mimic the error rate seen in the ccs reads constructed from LB2 and RB2. The simulated T-DNA flanks from each $Len_R$ were aligned back to the genome using minimap2 using “sr” preset, and precise alignment locations were determined. The proportion of uniquely mapped reads, proportion of reads uniquely mapping to insertion, and number reads that uniquely map to each insertion was calculated.

**Results**

**Transgenic DNA construct and the transgenic events**

Transgenic plants were generated at the Plant Transformation facility at Iowa State University, by transforming T-DNA constructs for RFP. In total, 20 transgenic events were generated by transformation. A simplified illustration of the T-DNA construct used is shown in Figure 3.1. The T-DNA construct has two genes, namely RFP and BAR. Each gene has a promoter sequence, and a terminator sequence. Left border (LB) and right border (RB) sequences mark the ends of the construct. Primers in the LB and RB sequences were used as T-DNA specific primers to enrich for T-DNA flanks. Specific differences between the three experiments are listed in Table 3.2. One additional difference among the experiments was, how the 707.26.1 event was handled. In LB1 the 707.26.1 was part of the R4 and C4 pools, but 707.26.1 was separately barcoded in both LB2 and RB2.

**Design of the bioinformatics pipeline**

The T-DNA enrichment protocol is shown in Figure 3.3, and an example T-DNA fragment is illustrated in the diagram. The bioinformatics analysis pipeline was specifically designed to use the features seen in the sequenced fragments. An intact T-DNA flank should contain three DNA fragments in the following order, namely barcoded adapter (bc-AP), a barcoded border (bc-LB or bc-RB), and a genomic fragment. A read that contains unique adapter, border and genomic sequences will be representative of an intact T-DNA flank. The
bioinformatics pipeline will focus on identifying intact T-DNA flanks by filtering reads that have these three sequence fragments, and then locating the T-DNA insertion location using the genomic placements of the T-DNA flanks

**Sequence analysis of T-DNA flanks**

Reads from the three experiments, namely LB1, LB2 and RB2 were analyzed to identify the T-DNA insertion locations. Initially the subreads from the PacBio RS II were used to construct circular consensus sequence (CCS) reads. CCS reads are substantially more accurate than subreads, especially for the identification of the barcodes. The number of subreads produced for each experiment, the proportion of subreads used for CCS construction, and the number of final CCS reads are given in Table 3.4. LB2 produced the highest number of subreads, but only ~39% of the subreads were used to construct CCS reads. The LB1 had the lowest number of subreads, but higher proportion ~79% of the subreads were used to construct CCS reads. Although RB2 produced more subreads than LB1, it had similar number of CCS reads as LB1.

The read lengths between the three experiments were compared to see if there were any length differences. Both LB2 and RB2 had higher median read lengths than LB1 (Table 3.4). The minimum read length varied between each experiment and was higher in LB2 and RB2. Median read length for the LB1 was higher than both LB2 and RB2 (Figure 3.5). The differences in read lengths were statistically significant when LB1 was compared to LB2 and RB2 (Wilcoxon rank sum test:LB1-LB2:P-value=0, Wilcoxon rank sum test:LB1-RB2:P-value=0) but was not different between LB2 and RB2 (Wilcoxon rank sum test:LB2-RB2:P-value=1). The next step was to identify the barcodes and assign the reads to pools. The number of reads assigned to the pools was counted for the three experiments. The individual read counts are given in Table 3.5. The mean number of reads assigned to each pool is higher for LB2, compared to LB1 and RB2, and this was expected based on the library size (number of CCS reads) of the LB2. The
proportion of reads that were assigned to a pool ranged from 85-90% for the experiments (Table 3.5). The number of CCS assigned to a pool in an experiment is called the library size (LS) of the pool. A visual comparison of the proportions of LS are given in Figure 3.6. Visual comparison of the proportion of LS in each library indicates that the libraries were better balanced in LB2 and RB2 compared to LB1. The standard deviation of the proportions also confirmed this observation ($\sigma_{LS(LB1)} = 3.39\% ; \sigma_{LS(LB2)} = 1.72\% ; \sigma_{LS(RB2)} = 1.90\% $). Unbalanced libraries could lead to limitations in identifying insertions from the low-represented pools.

Next, the CCS reads were aligned to the adapter, border and maize genome sequences. This step was performed on all CCS reads to determine order and orientation of the border, adapter, and genomic fragments in each read. Some reads were filtered out because the fragments were not unique (LB1~4.0%, LB2~6.2%, RB2~9.8%). The number of reads that contain different sequence fragment combinations were determined for the pools (Figure 3.7) from all three experiments. Intact T-DNA flanks and other artifacts were identified from these alignments. Proportion of reads that represent intact T-DNA flanks was less than 50% in all three experiments. The T-DNA flank proportion varied between the three experiments. Both LB2 (~37%) and RB2 (~22%) had significantly lower number of intact flanks than LB1 (~47%). Although LB1 and RB2 have similar library sizes, the number of intact flanks is significantly lower for RB2. The number of intact flanks assigned to each pool is presented in Table 3.6. In RB2 experiment two pools had almost no intact flanks, namely 707 and R1. In general, the RB2 experiment had lower number of intact flanks compared to two LB based experiments. Visual comparison of the proportion of reads from intact flanks revealed that LB1 had a more consistent proportion of intact flanks across all the barcodes when compared to both LB2 and RB2 (Figure 3.8). The standard deviation of proportion of intact flanks assigned to the
pools \( (\sigma_{prop(UF)}) \), also confirm that LB1 \((\sigma_{prop(UF)} = 0.03)\) is more balanced than LB1 \((\sigma_{prop(UF)} = 0.10)\) and RB2 \((\sigma_{prop(UF)} = 0.14)\). Of the other read types, most of the variation from the LB2 and RB2 library seems to be from the reads that contain only border and adapter fragments.

**Identification of putative T-DNA insertion locations**

The intact flanks from each pool were used to determine the putative T-DNA insertion locations for the transgenic events in that pool, as each event is represented by a unique row and column pool intersect. The peaks can be identified using read depth across the genome. Simply using read depth identifies a few false-positive regions, due to the noise that was not cleaned by identifying only intact fragments. We can use specific of read depth characteristic observed around the T-DNA insertion (Figure 3.9). The specific non-random pattern is due to the variation of the length of genomic flanks within reads. The correlation coefficient \( (r) \) between the read depth and genomic position was calculated. The \( r \) was used to filter out false-positive locations. The \( r \) was used to determine a single-base genomic T-DNA insertion location, and whether the insertion was upstream or downstream of the genomic position identified. Depending on which border primer was used the orientation of the T-DNA insertion was also identified. A total of 198 peaks with adequate read depth were identified from the genomic alignments of the uniquely aligned intact flanks from all the pools of the three experiments, and only 127 of these locations passed the filtering criteria. The peaks were summarized to identify the pools that shared T-DNA insertion locations (Table 3.7). Both LB1 and LB2 identified 13 insertion locations, compared to only 7 from RB2. Of the 13 insertion locations from LB1 and LB2, 10 were shared between the both, and 4 were unique to each. The two locations from LB1 that were not shared with LB2 were placed on Chromosome 1 and these were present in all the pools. Also, the location from
707.26.1 is properly identified in LB2, but the same location is contained in all the pools in LB1. Out of the 13 insertions identified by LB1, ~85% of the reads are within five locations that were assigned to 8 or more pools. Neither LB2 nor RB2 show the same issue.

The pools from the experiments were deconvoluted to identify individual transgenic events (Table 3.8). This also gives a comparison of the pools that were present in each experiment. Once the locations from all experiments are summarized, the number of unique genomic locations reduced to 22. In total 16 of the 20 transgenic events have been mapped to a specific location in the genome by the three experiments, and out of the 16 mapped insertions 14 have been confirmed by PCR experiments.

Simulation of the impact of the read length on precise alignment of the flanks

To see whether read length would impact the identification of the insertion sites, a simulation of flanks was performed around the T-DNA insertions identified in this project. The simulation was performed using the longest two sequencing configurations available for Illumina MiSeq (500-bp and 600-bp), and two read lengths for long-reads (1200-bp and 2000-bp). The details of how the flanks were simulated based on read length is described in methods. The average lengths of the simulated reads were \(\overline{Len_F(500)}=142.79; \overline{Len_F(600)}=190.9; \overline{Len_F(1200)} = 568.71; \overline{Len_F(2000)} = 963.41\). Two metrics were calculated, namely proportion of uniquely aligned reads, and proportion of unique reads mapped to the correct location (Table 3.9). The mean fragment lengths for all the read lengths were substantially smaller, but the flank size was uniformly distributed across the possible lengths for a given read length. The smaller length flanks have lower unique alignments to the genome, but majority of the uniquely aligned flanks map to the insertion site for all read lengths. Interestingly, the 2000-bp flanks have lower uniquely aligned flanks than 1200-bp. This could potentially be overcome by optimizing the read
alignment parameters for minimap2. To visually determine which insertion locations were mappable by different length of reads, a heat map was generated for the number of flanks that were uniquely mapped to each the 20 selected insertion locations (Figure 3.10). For most of the insertion locations the longer reads improve the flank counts at the insertion sites. Insertion 8 only has flanks mapped for the 2000-bp read length. Insertion 6 is not mapped in the smallest read length, and the read numbers are low in other read-lengths, indicating a repeat region. Apart from insertion 3, all the other insertions benefit from longer read lengths.

**Discussion**

In this project, we have developed a protocol to enrich genomic flanks surrounding the T-DNA insertions and sequenced them using a PacBio platform. The bioinformatics analysis of the sequences placed the flanks on the genome to identify the genomic T-DNA insertion locations. We mapped 16 of the 20 the transgenic events to a genomic location using the sequenced T-DNA flanks. Of the 16 events mapped 14 were confirmed by PCR with genome specific primers.

Among the three experimental approaches used for enriching T-DNA flanks, the enrichment approach used in LB2 identified the most T-DNA insertions accurately. LB2 flank enrichment approach identified 13 of 14 confirmed events, and in this approach the low-cycle PCR with LB specific primer was used for the flank enrichment. In comparison, the RB2 experiment that used RB primers was only was only able to identify 9 of the 14 confirmed events. The difference in the number of identified events could mainly be due to the imprecise nature of T-DNA integration into the plant genome (Gelvin, 2017; Krizkova & Hrouda, 1998). There is evidence for concatenated T-DNA insertions in some transgenic events (707.26.11 and pool R1 in LB2), and there could be more such events that make the reads from RB extend into the vector backbone or a second LB. The general observation from other studies is that the LB
has weaker consensus cleavage site than RB, but this doesn’t seem to impact the detection of
insertion sites using LB in this project (Gelvin, 2017).

Combining the 707.26.1 transgenic event into the R4 and C4 pools, and linear
amplification used in LB1 also identified the same insertions as LB2. There were unexpectedly
few insertions that were overrepresented in terms of number of reads aligned. The same
overrepresented locations contained reads from all nine pools and could not be reliably assigned
to specific transgenic events. The barcode analysis steps were rechecked to confirm that it was
not an issue with the barcode assignment step. This suggests an issue with the molecular protocol
or pooling stage or intermixing the DNA from the pools. Two of the locations from LB1 on
chromosome 1 were short repeat regions that had large number of short alignments and are most
likely false positive alignment of repeat sequences. LB1 also had lower mean read length
compared to LB2 and RB2, and this could be a consequence of linear amplification technique,
where lower numbers of size-enriched flanks were produced as compared with low cycle
exponentially amplified LB2 and RB2 libraries. The data suggest linear amplification is
comparable to low-cycle PCR in terms of enrichment potential, but since there were a large
number of reads appearing is several pools, the LB1 experiment should be repeated to make a
meaningful comparison between those approaches.

The bioinformatics analysis was designed for the accurate identification of the T-DNA
flanks on the genome and to reduce the false-positive T-DNA insertions. This works especially
well for insertions in the unique and non-repetitive regions and has been successful in placing
majority of the insertions. The analysis pipeline is fast and the time from sequence to putative
insertions is around one to two days. The read depth and correlation coefficient analysis
identifies a precise single base pair T-DNA insertion location in the maize genome and identifies
the orientation of the T-DNA insertion. The simple filtering criteria used for the analysis seem to perform well if the constructed library is balanced, and a good sequencing dataset is produced. The pipeline could be further tweaked to identify more meaningful filtering criteria, to improve the identification of barcodes, vector and genomic alignments. The statistical analysis of the peaks could be improved by borrowing the normalization techniques used for other sequence analyses. It might be more prudent to identify insertion locations using the aligned reads from all the pools, and use read counts and normalization techniques to get more comparable numbers for each T-DNA flank region for the pools. This would eliminate the need for hard-defined thresholds for filtering criteria such as minimum number of reads, and correlation coefficient.

Currently the pipeline doesn’t handle reads multi-mapped to the genome, or other sequences. Filtering for unique alignments limits the flank alignment to low copy regions. The noise introduced by chimeric artifacts from the molecular protocol could be an issue when working with multi-mapped reads. The simulation and manual inspection of the alignment indicate, that even some of the insertions identified by uniquely aligned reads could be in repeat rich regions. These could be false-positive insertion locations for which the real insertion is located on a paralogous region elsewhere in the genome. More careful optimization of the alignment parameters and filtering criteria is essential to eliminate such false-positives insertions. The analysis could be optimized to detect and isolate intact flanks from reads with non-unique fragments and trim the concatenated fragments on the read. The number of additional reads could be inconsequential for insertions with higher signal but would benefit insertions that are hampered by library imbalance. The reads can be used to understand the complex insertion pattern shown by T-DNA insertions by aligning the reads to the vector sequence. Analyzing the order and orientation of the vector fragments in each pool, could give a more accurate picture of
which insertions are structurally complex. Identifying complex insertions would be useful in determining the cause of imbalanced read counts among pools such as the one shown by R1 in RB2 experiment. The complex insertion patterns could impact the expression patterns of the construct, and potentially lead to misleading conclusions in downstream biological studies.

The utility of using long-read sequencing is showcased by the simple simulation performed around the T-DNA insertion location using the maize genome sequence. The simulations point to the fact most of the insertion locations show a positive correlation between the read length and the number of reads. Several locations indicate that the shorter reads cannot be used to place those insertions reliably on the maize genome. It is possible that the same issue exists in other crop-species with repetitive genomes. The results from the simulation should be taken with care. The simulation has several caveats in terms of the parameters chosen—such as the example mutation rates, mutation types, and read lengths that could potentially be addressed by fitting better parameters derived from real datasets. An improved simulation is also a method to assess the utility of longer-read sequencing of enriched-flanks for other types of genomic insertions as well. Specifically, it would be of value to examine, through simulation, the utility of longer reads for the accurate placement of transposon insertions generated in large-scale mutagenesis projects in maize. Two examples in maize are the Mutator collection, and the Activator/Dissociation collection (Y. Li, Segal, Wang, & Dooner, 2013; Settles et al., 2007; Vollbrecht et al., 2010). The Ac/Ds project has traditionally used iPCR for flank placement and is currently developing an Illumina-based enrichment protocol for high-throughput placement of transposon flanks. The Mutator collections have used an Illumina-based method (Mu-seq) to identify Mu insertion locations (McCarty et al., 2013). Both these collections have insertions that have not been mapped, and longer reads could potentially help improve the placement of these
The existing multi-mapped insertions can be used to simulate the read-length necessary to map those insertions more accurately and the flank enrichment protocol could be tested and optimized to generate reads of a certain size predicted as necessary to accurately map those insertions. One of the next steps for the project is to work on simulating other types on insertions in maize and determine the read sizes required to map those insertions.

In summary, the T-DNA flank enrichment protocol coupled to long-reads sequencing has successfully used to identify the genomic locations of T-DNA insertions in transgenic maize. Simulated reads show that longer reads perform better than shorter reads when identifying T-DNA insertions using in the repetitive maize genome.

**Author Contributions**

EUW generated the T-DNA constructs and oversaw the transgenic event generation. KW and EUW designed the experiments. EUW performed the bench experiments to enrich the flanks and construct the sequencing library. KW performed the bioinformatics analysis. EV contributed to the experimental design and supervised the work. KW prepared the first draft of the manuscript, EUW and EV edited the manuscript. All authors read and approved the final manuscript.
Tables and Legends

Table 3.1: The transgenic events in column and row pools

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>668.11.1</td>
<td>668.2.1</td>
<td>668.4.1</td>
<td>668.8.5</td>
</tr>
<tr>
<td>R2</td>
<td>668.13.4</td>
<td>668.15.3</td>
<td>668.5.2</td>
<td>668.9.5</td>
</tr>
<tr>
<td>R3</td>
<td>668.14.2</td>
<td>668.5.4</td>
<td>668.2.4</td>
<td>668.7.1†</td>
</tr>
<tr>
<td>R4</td>
<td>668.15.1</td>
<td>668.3.1</td>
<td>668.8.1</td>
<td>707.26.1</td>
</tr>
<tr>
<td>R5</td>
<td>668.3.4</td>
<td>668.8.2</td>
<td>668.14.2</td>
<td>668.15.11</td>
</tr>
</tbody>
</table>

†In LB1, The DNA extracted from the 707.26.1 event was pooled in C4 and R4 pools. In LB2 and RB1 the 707.26.1 event was prepared independently using different barcode (0054_Rev).

Table 3.2: Differences between the T-DNA flank experiments

<table>
<thead>
<tr>
<th>Step</th>
<th>Experiment 1 (LB1)</th>
<th>Experiment 2 (LB2)</th>
<th>Experiment 3 (RB2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trial #</td>
<td>trial 1</td>
<td>trial 2</td>
<td>trial 2</td>
</tr>
<tr>
<td>Pooling</td>
<td>707 spiked in 668 pools</td>
<td>707 Barcoded separate</td>
<td>707 Barcoded separate</td>
</tr>
<tr>
<td>T-DNA specific primer</td>
<td>Left Border (LB)</td>
<td>Left Border (LB)</td>
<td>Right Border (RB)</td>
</tr>
<tr>
<td>Enrichment</td>
<td>Linear Amplification (bLB)</td>
<td>Low-cycle PCR (bLB)</td>
<td>Low-cycle PCR (bRB)</td>
</tr>
<tr>
<td>Amplification</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: minimap2 Alignment and filtering criteria for the CCS reads

<table>
<thead>
<tr>
<th>Reference Sequence</th>
<th>Preset</th>
<th>MAPQ</th>
<th>Alignment Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barcoded Adapter (bc-AP)</td>
<td>sr</td>
<td>&gt;=55</td>
<td>82 – 86</td>
</tr>
<tr>
<td>Barcoded Border (bc-LB or bc-RB)</td>
<td>sr</td>
<td>&gt;=15</td>
<td>47 – 51</td>
</tr>
<tr>
<td>Maize RefGen_v4</td>
<td>asm5</td>
<td>&gt;= 55</td>
<td>47 – 51</td>
</tr>
</tbody>
</table>

Table 3.4: Summary of sequencing reads for each experiment

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Read Numbers</th>
<th>Read Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Subreads</td>
<td>Used for CCS</td>
</tr>
<tr>
<td>LB1</td>
<td>956,512</td>
<td>687,875</td>
</tr>
<tr>
<td>LB2</td>
<td>1,938,948</td>
<td>754,671</td>
</tr>
<tr>
<td>RB2</td>
<td>1,515,514</td>
<td>344,136</td>
</tr>
</tbody>
</table>
Table 3.5: Library sizes ($LS$) of the pools

<table>
<thead>
<tr>
<th>Pool</th>
<th>LB1</th>
<th>LB2</th>
<th>RB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>707</td>
<td>NA</td>
<td>4,898</td>
<td>2,588</td>
</tr>
<tr>
<td>C1</td>
<td>2,541</td>
<td>3,907</td>
<td>3,346‡</td>
</tr>
<tr>
<td>C2</td>
<td>2,780</td>
<td>4,080</td>
<td>1,910</td>
</tr>
<tr>
<td>C3</td>
<td>3,804</td>
<td>3,985</td>
<td>2,284</td>
</tr>
<tr>
<td>C4</td>
<td>2,982</td>
<td>3,249</td>
<td>1,914</td>
</tr>
<tr>
<td>R1</td>
<td>3,154</td>
<td>3,086‡</td>
<td>1,751‡</td>
</tr>
<tr>
<td>R2</td>
<td>1,256‡</td>
<td>4,930</td>
<td>2,779</td>
</tr>
<tr>
<td>R3</td>
<td>4,078‡</td>
<td>5,336</td>
<td>2,792</td>
</tr>
<tr>
<td>R4</td>
<td>1,621</td>
<td>5,504‡</td>
<td>1,867</td>
</tr>
<tr>
<td>R5</td>
<td>2331</td>
<td>4,964</td>
<td>2,371</td>
</tr>
<tr>
<td>Unassigned</td>
<td>2,655</td>
<td>5,383</td>
<td>3,648</td>
</tr>
</tbody>
</table>

‡ indicates $min(LS)$ of the experiment ‡ indicates $max(LS)$ of the experiment

Table 3.6: Number of intact T-DNA flanks in each pool

<table>
<thead>
<tr>
<th>Pool</th>
<th>LB1</th>
<th>LB2</th>
<th>RB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>707</td>
<td>NA</td>
<td>1,354</td>
<td>16</td>
</tr>
<tr>
<td>C1</td>
<td>1,180</td>
<td>2,016</td>
<td>415</td>
</tr>
<tr>
<td>C2</td>
<td>1,312</td>
<td>1,734</td>
<td>290</td>
</tr>
<tr>
<td>C3</td>
<td>1,635</td>
<td>1,261</td>
<td>750</td>
</tr>
<tr>
<td>C4</td>
<td>1,304</td>
<td>1,164</td>
<td>422</td>
</tr>
<tr>
<td>R1</td>
<td>1,622</td>
<td>1,075</td>
<td>8‡</td>
</tr>
<tr>
<td>R2</td>
<td>584‡</td>
<td>2,253</td>
<td>1,014</td>
</tr>
<tr>
<td>R3</td>
<td>1,861‡</td>
<td>1,026</td>
<td>1,020‡</td>
</tr>
<tr>
<td>R4</td>
<td>890</td>
<td>2,495‡</td>
<td>656</td>
</tr>
<tr>
<td>R5</td>
<td>1,169</td>
<td>1,842</td>
<td>574</td>
</tr>
</tbody>
</table>

‡ indicates minimum number of the column, and ‡ indicates of maximum number in the column
Table 3.7: Summary of deconvoluted pools from the intact T-DNA flank placements from pools

<table>
<thead>
<tr>
<th>Chr</th>
<th>Position</th>
<th>Experiment</th>
<th>T-DNA Position</th>
<th>Pools</th>
<th>Min Cor</th>
<th>Min Reads</th>
<th>Max Reads</th>
<th>Total Reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chr1</td>
<td>124933623</td>
<td>LB1</td>
<td>Downstream</td>
<td>C1,C2,C3,C4,R1,R2,R3,R4,R5</td>
<td>0.92</td>
<td>92</td>
<td>453</td>
<td>2124</td>
</tr>
<tr>
<td>Chr1</td>
<td>124933722</td>
<td>LB1</td>
<td>Upstream</td>
<td>C1,C2,C3,C4,R1,R2,R3,R4,R5</td>
<td>-0.97</td>
<td>190</td>
<td>716</td>
<td>4067</td>
</tr>
<tr>
<td>Chr1</td>
<td>149278276</td>
<td>RB2</td>
<td>Downstream</td>
<td>C1,C3,R3,R5</td>
<td>0.94</td>
<td>448</td>
<td>567</td>
<td>2059</td>
</tr>
<tr>
<td>Chr1</td>
<td>149289578</td>
<td>LB1</td>
<td>Upstream</td>
<td>C1,C3,R3,R5</td>
<td>-0.96</td>
<td>89</td>
<td>157</td>
<td>503</td>
</tr>
<tr>
<td>Chr1</td>
<td>149289578</td>
<td>LB2</td>
<td>Upstream</td>
<td>C1,C3,R2,R3,R5</td>
<td>-0.95</td>
<td>32</td>
<td>804</td>
<td>2505</td>
</tr>
<tr>
<td>Chr2</td>
<td>41538805</td>
<td>LB2</td>
<td>Upstream</td>
<td>C3,R1</td>
<td>-0.98</td>
<td>179</td>
<td>209</td>
<td>388</td>
</tr>
<tr>
<td>Chr2</td>
<td>41544452</td>
<td>RB2</td>
<td>Downstream</td>
<td>C3</td>
<td>0.95</td>
<td>344</td>
<td>344</td>
<td>344</td>
</tr>
<tr>
<td>Chr2</td>
<td>213932560</td>
<td>LB1</td>
<td>Downstream</td>
<td>C3,R2,R3</td>
<td>0.87</td>
<td>94</td>
<td>242</td>
<td>508</td>
</tr>
<tr>
<td>Chr2</td>
<td>213932560</td>
<td>LB2</td>
<td>Downstream</td>
<td>C2,C3,R2,R3</td>
<td>0.95</td>
<td>333</td>
<td>783</td>
<td>2252</td>
</tr>
<tr>
<td>Chr2</td>
<td>214159798</td>
<td>RB2</td>
<td>Upstream</td>
<td>C2,C3,R2,R3</td>
<td>-0.97</td>
<td>155</td>
<td>1144</td>
<td>2383</td>
</tr>
<tr>
<td>Chr2</td>
<td>218947217</td>
<td>RB2</td>
<td>Upstream</td>
<td>C4,R3</td>
<td>-0.97</td>
<td>356</td>
<td>484</td>
<td>840</td>
</tr>
<tr>
<td>Chr5</td>
<td>216879111</td>
<td>RB2</td>
<td>Downstream</td>
<td>C3,R3</td>
<td>0.96</td>
<td>58</td>
<td>92</td>
<td>150</td>
</tr>
<tr>
<td>Chr5</td>
<td>216879117</td>
<td>LB1</td>
<td>Upstream</td>
<td>C3,R3</td>
<td>-0.97</td>
<td>95</td>
<td>102</td>
<td>197</td>
</tr>
<tr>
<td>Chr5</td>
<td>216879117</td>
<td>LB2</td>
<td>Upstream</td>
<td>C3,R3</td>
<td>-0.96</td>
<td>393</td>
<td>572</td>
<td>965</td>
</tr>
<tr>
<td>Chr5</td>
<td>218913242</td>
<td>LB1</td>
<td>Downstream</td>
<td>C2,R1</td>
<td>0.94</td>
<td>42</td>
<td>99</td>
<td>141</td>
</tr>
<tr>
<td>Chr5</td>
<td>218913242</td>
<td>LB2</td>
<td>Downstream</td>
<td>C2,R1</td>
<td>0.95</td>
<td>435</td>
<td>577</td>
<td>1012</td>
</tr>
<tr>
<td>Chr6</td>
<td>157650848</td>
<td>LB1</td>
<td>Downstream</td>
<td>C1,C2,R3,R4,R5</td>
<td>0.93</td>
<td>49</td>
<td>324</td>
<td>741</td>
</tr>
<tr>
<td>Chr6</td>
<td>157650848</td>
<td>LB2</td>
<td>Downstream</td>
<td>C1,C2,R3,R4,R5</td>
<td>0.94</td>
<td>220</td>
<td>1912</td>
<td>4770</td>
</tr>
<tr>
<td>Chr6</td>
<td>157650948</td>
<td>RB2</td>
<td>Upstream</td>
<td>C1,C2,R3,R4,R5</td>
<td>-0.99</td>
<td>60</td>
<td>493</td>
<td>941</td>
</tr>
<tr>
<td>Chr6</td>
<td>168446443</td>
<td>LB1</td>
<td>Downstream</td>
<td>C1,C2,C3,C4,R1,R2,R3,R4,R5</td>
<td>0.95</td>
<td>127</td>
<td>489</td>
<td>3051</td>
</tr>
<tr>
<td>Chr6</td>
<td>168446443</td>
<td>LB2</td>
<td>Downstream</td>
<td>707</td>
<td>0.95</td>
<td>1723</td>
<td>1723</td>
<td>1723</td>
</tr>
<tr>
<td>Chr7</td>
<td>94931756</td>
<td>LB1</td>
<td>Downstream</td>
<td>C1,C2,C3,C4,R1,R2,R3,R5</td>
<td>0.9</td>
<td>217</td>
<td>811</td>
<td>3885</td>
</tr>
</tbody>
</table>
Table 3.7 continued: Summary of deconvoluted pools from the intact T-DNA flank placements from pools

<table>
<thead>
<tr>
<th>Chr</th>
<th>Position</th>
<th>Experiment</th>
<th>T-DNA Position</th>
<th>Pools</th>
<th>Min Cor</th>
<th>Min Reads</th>
<th>Max Reads</th>
<th>Total Reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chr7</td>
<td>94931756</td>
<td>LB2</td>
<td>Downstream</td>
<td>C4,R2</td>
<td>0.95</td>
<td>1270</td>
<td>1556</td>
<td>2826</td>
</tr>
<tr>
<td>Chr7</td>
<td>102903973</td>
<td>LB1</td>
<td>Downstream</td>
<td>C1,R1</td>
<td>0.97</td>
<td>127</td>
<td>162</td>
<td>289</td>
</tr>
<tr>
<td>Chr7</td>
<td>102903973</td>
<td>LB2</td>
<td>Downstream</td>
<td>C1,R1</td>
<td>0.96</td>
<td>567</td>
<td>597</td>
<td>1164</td>
</tr>
<tr>
<td>Chr7</td>
<td>102904031</td>
<td>LB1</td>
<td>Upstream</td>
<td>C1,R1</td>
<td>-0.97</td>
<td>30</td>
<td>51</td>
<td>81</td>
</tr>
<tr>
<td>Chr7</td>
<td>102904031</td>
<td>LB2</td>
<td>Upstream</td>
<td>C1,R1</td>
<td>-0.97</td>
<td>290</td>
<td>294</td>
<td>584</td>
</tr>
<tr>
<td>Chr10</td>
<td>4194146</td>
<td>LB1</td>
<td>Downstream</td>
<td>C1,R2</td>
<td>0.94</td>
<td>51</td>
<td>78</td>
<td>129</td>
</tr>
<tr>
<td>Chr10</td>
<td>4194146</td>
<td>LB2</td>
<td>Downstream</td>
<td>C1,R2</td>
<td>0.94</td>
<td>291</td>
<td>407</td>
<td>698</td>
</tr>
<tr>
<td>Chr10</td>
<td>4208995</td>
<td>LB2</td>
<td>Upstream</td>
<td>C1,R2</td>
<td>-0.84</td>
<td>218</td>
<td>236</td>
<td>454</td>
</tr>
<tr>
<td>Chr10</td>
<td>96587927</td>
<td>RB2</td>
<td>Downstream</td>
<td>C1,C2,C4,R2,R4,R5</td>
<td>0.86</td>
<td>81</td>
<td>393</td>
<td>1245</td>
</tr>
<tr>
<td>Chr10</td>
<td>96588084</td>
<td>LB1</td>
<td>Upstream</td>
<td>C1,C2,C3,C4,R1,R2,R3,R4,R5</td>
<td>-0.98</td>
<td>88</td>
<td>266</td>
<td>1390</td>
</tr>
<tr>
<td>Chr10</td>
<td>96588084</td>
<td>LB2</td>
<td>Upstream</td>
<td>C1,C2,C4,R2,R4,R5</td>
<td>-0.99</td>
<td>280</td>
<td>1499</td>
<td>4268</td>
</tr>
</tbody>
</table>
Table 3.8: Summary of putative T-DNA flank placements from all experiments

<table>
<thead>
<tr>
<th>Insertion</th>
<th>Chr</th>
<th>Pos</th>
<th>T-DNA Pos</th>
<th>LB1</th>
<th>LB2</th>
<th>RB2</th>
<th>Confirmed</th>
<th>TG Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chr1</td>
<td>124933623</td>
<td>Downstream</td>
<td>C1,C2,C3,C4, R1,R2,R3,R4,R5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Chr1</td>
<td>124933722</td>
<td>Upstream</td>
<td>C1,C2,C3,C4, R1,R2,R3,R4,R5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Chr1</td>
<td>149278276</td>
<td>Downstream</td>
<td>C1,C3,R3,R5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Chr1</td>
<td>149289578</td>
<td>Upstream</td>
<td>C1,C3,R3,R5</td>
<td>C1,C3,R2,R3,R5</td>
<td></td>
<td></td>
<td>668.14.2</td>
</tr>
<tr>
<td>5</td>
<td>Chr1</td>
<td>14538805</td>
<td>Upstream</td>
<td>C3,R1</td>
<td></td>
<td></td>
<td></td>
<td>668.4.1</td>
</tr>
<tr>
<td>6</td>
<td>Chr1</td>
<td>4154452</td>
<td>Downstream</td>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Chr1</td>
<td>213932560</td>
<td>Downstream</td>
<td>C3,R2,R3</td>
<td>C2,C3,R2,R3</td>
<td></td>
<td></td>
<td>668.5.2,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>668.5.4</td>
</tr>
<tr>
<td>8</td>
<td>Chr2</td>
<td>214159798</td>
<td>Upstream</td>
<td></td>
<td>C2,C3,R2,R3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Chr2</td>
<td>218947217</td>
<td>Upstream</td>
<td>C4,R3</td>
<td></td>
<td></td>
<td></td>
<td>668.7.1</td>
</tr>
<tr>
<td>10</td>
<td>Chr2</td>
<td>216879111</td>
<td>Downstream</td>
<td></td>
<td>C3,R3</td>
<td>C3,R3</td>
<td></td>
<td>668.2.4</td>
</tr>
<tr>
<td>11</td>
<td>Chr2</td>
<td>218913242</td>
<td>Upstream</td>
<td>C2,R1</td>
<td>C2,R1</td>
<td></td>
<td></td>
<td>668.2.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>same</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Chr5</td>
<td>157650848</td>
<td>Downstream</td>
<td>C1,C2,R3,R4,R5</td>
<td>C1,C2,R3,R4,R5</td>
<td></td>
<td></td>
<td>668.3.1,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>668.3.4</td>
</tr>
<tr>
<td>13</td>
<td>Chr6</td>
<td>157650948</td>
<td>Upstream</td>
<td></td>
<td>C1,C2,R3,R4,R5</td>
<td>Same</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Chr6</td>
<td>168446443</td>
<td>Downstream</td>
<td>C1,C2,C3,C4, R1,R2,R3,R4,R5</td>
<td>707</td>
<td>707</td>
<td></td>
<td>707.26.1</td>
</tr>
<tr>
<td>15</td>
<td>Chr6</td>
<td>94931756</td>
<td>Downstream</td>
<td>C1,C2,C3,C4, R1,R2,R3,R4,R5</td>
<td>C4,R2</td>
<td></td>
<td></td>
<td>668.9.5?</td>
</tr>
<tr>
<td>16</td>
<td>Chr6</td>
<td>102903971</td>
<td>Downstream</td>
<td>C1,R1</td>
<td>C1,R1</td>
<td></td>
<td></td>
<td>668.11.1</td>
</tr>
<tr>
<td>17</td>
<td>Chr6</td>
<td>102904301</td>
<td>Upstream</td>
<td>C1,R1</td>
<td>C1,R1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>same location</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Chr7</td>
<td>4194146</td>
<td>Downstream</td>
<td>C1,R2</td>
<td>C1,R2</td>
<td></td>
<td></td>
<td>668.13.4</td>
</tr>
<tr>
<td>19</td>
<td>Chr7</td>
<td>4208995</td>
<td>Upstream</td>
<td>C1,R2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Chr7</td>
<td>96587927</td>
<td>Downstream</td>
<td>C1,C2,C4, R1,R2,R3,R4,R5</td>
<td>C1,C2,C4, R2,R4,R5</td>
<td></td>
<td></td>
<td>668.15.1</td>
</tr>
<tr>
<td>21</td>
<td>Chr8</td>
<td>96588084</td>
<td>Upstream</td>
<td>C1,C2,C3,C4, R1,R2,R3,R4,R5</td>
<td>C1,C2,C4, R2,R4,R5</td>
<td></td>
<td></td>
<td>668.15.1</td>
</tr>
<tr>
<td>22</td>
<td>Chr8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>668.15.3</td>
</tr>
</tbody>
</table>
Table 3.9: Results of the alignment from simulated flanks

<table>
<thead>
<tr>
<th>Read Length</th>
<th>Flanks Simulated</th>
<th>( Len_F )</th>
<th>Unique Alignments</th>
<th>Uniquely mapped to Insertion</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>30,000</td>
<td>141.56</td>
<td>64.22%</td>
<td>64.19%</td>
</tr>
<tr>
<td>600</td>
<td>30,000</td>
<td>191.34</td>
<td>71.91%</td>
<td>71.74%</td>
</tr>
<tr>
<td>1,200</td>
<td>30,000</td>
<td>565.28</td>
<td>87.45%</td>
<td>85.52%</td>
</tr>
<tr>
<td>2,000</td>
<td>30,000</td>
<td>963.41</td>
<td>84.14%</td>
<td>81.42%</td>
</tr>
</tbody>
</table>

Figures and Legends

Figure 3.1: The transgenic construct from the events

The figure illustrates the generic construct that was transformed during the transgenic events. The construct has two genes, namely RFP and BAR on opposing strands. Each gene has promoters to drive genes expression, and terminators to stop transcription. LB and RB indicate left and right borders.
Figure 3.2: Overview of an experiment

The overview of the main steps of a single experiment performed to identify the genomic T-DNA insertion locations. Details of the bioinformatics analysis is given in figure 3.3.
Figure 3.3: T-DNA flank enrichment Protocol

The main steps performed during the T-DNA flank enrichment protocol. The diagram presents the main steps, and conceptually illustrates the different sequence fragments that are added during each step. LB=Left border, RB=Right border, bc-RB=barcoded right border primer, bc-AP=barcoded left adapter primer. Dotted lines indicate regions of the fragment that is newly generated during the amplification.
Figure 3.4: Bioinformatics Analysis Overview

The steps used for the bioinformatic analysis of the sequencing data produced for each experiment.
Figure 3.5: Read length distribution of the CCS read

Violin plot of the CCS read length distribution of LB1, LB2 and RB2. The three experiments are on the x-axis, and colored as follows: LB1 is green, LB2 is lavender, and RB2 tan. Area of the plots represents the number of CCS reads produced for each experiment. The y-axis is the read length. The width of the shape at a given point in the y-axis is represents the number of reads that are of that specific length in the sample. The three lines drawn are the first, second (median), and third quartiles.
Figure 3.6: Proportion of reads that are assigned to each pool

The figure illustrates a stacked proportion plot of the reads that were assigned to specific pools in LB1, LB2, and RB2. The unassigned reads from each pool are given in yellow color. The experiments are laid out on the x-axis, and the proportion of reads are shown in y-axis. The y-axis has been normalized to 1 for each experiment.
Figure 3.7: Number of reads that contain sequence fragments from adapter, border and genome composition. Adapter fragments are given in lavender, genome is given in green and border is given in orange. (a) Reads composition counts for LB1, (b) Read composition count for LB2, and (c) Read composition for RB2.
Figure 3.8: Proportion of reads that contain sequence fragments from adapter, border and genome for the pools

Stacked proportion plots of fragment composition for individual pools in each experiment. The pools are represented on the x-axis, and the proportion of reads are represented on the y-axis. The stacked proportion plot for each experiment is arranged vertically top to bottom in the following order, LB1, LB2 and RB2. The abbreviations are as follows, bor=border, ada=adapter, gen=genome. Tan color represents the intact flanks (bor-ada-gen).
Figure 3.9: Read depth patterns around different T-DNA insertion directions

Read depth in three different genomic locations. The x-axis represents the genomic position, and y-axis represents read depth. The three panels from left to right represent downstream, undetermined, and upstream T-DNA insertion directions. The chromosome the insertions are from are colored with green for Chr 6 and lavender for Chr 10.
Figure 3.10: Heat map of simulated reads correctly mapped to the insertion
The number of simulated T-DNA flanks that were correctly uniquely aligned to the maize genome and accurately mapped to the T-DNA insertion. Y-axis - Insertion, and X-axis is the Read length. The gradient color scale indicates the read count correctly mapped to the insertion. Red=0 and Blue=1500. For each insertion, 1500 T-DNA flanks were produced by simulation.


posttranscriptional silencing of homologous host genes in plants. *Molecular and Cellular Biology, 18*(11), 6165–6177.


CHAPTER 4. MAIZE GO ANNOTATION- METHODS, EVALUATION, AND REVIEW (maize-GAMER)

Wimalanathan Kokulapalan, Iddo Friedberg, Carson M. Andorf, and Carolyn J. Lawrence-Dill

Abstract

We created a new high-coverage, robust, and reproducible functional annotation of maize protein coding genes based on Gene Ontology (GO) term assignments. Whereas the existing Phytozome and Gramene maize GO annotation sets only cover 41% and 56% of maize protein coding genes, respectively, this study provides annotations for 100% of the genes. We also compared the quality of our newly-derived annotations with the existing Gramene and Phytozome functional annotation sets by comparing all three to a manually annotated gold standard set of 1,619 genes where annotations were primarily inferred from direct assay or mutant phenotype. Evaluations based on the gold standard indicate that our new annotation set is measurably more accurate than those from Phytozome and Gramene. To derive this new high-coverage, high-confidence annotation set we used sequence-similarity and protein-domain-presence methods as well as mixed-method pipelines that developed for the Critical Assessment of Function Annotation (CAFA) challenge. Our project to improve maize annotations is called maize-GAMER (GO Annotation Method, Evaluation, and Review) and the newly-derived annotations are accessible via MaizeGDB (http://download.maizegdb.org/maize-GAMER) and CyVerse (B73 RefGen_v3 5b+ at doi.org/10.7946/P2S62P and B73 RefGen_v4 Zm00001d.2 at doi.org/10.7946/P2M925).

Introduction

Maize is an agriculturally important crop species and model organism for genetics and genomics research (Lawrence, Dong, Polacco, Seigfried, & Brendel, 2004). Not only is maize historically important for genetics research, along with other model species, significant efforts
have been made to transition existing datasets into a more sequence-centric paradigm (Sen et al., 2009), thus enabling genomics approaches to be brought to bear on both basic research problems and applied breeding (Lawrence et al., 2008). In 2009 the maize genome’s reference sequence was made available to the research community (Schnable et al., 2009). Since then, much work has gone into improving the utility of the genome sequence to scientists with a focus on sequence annotation.

In practice, making a genome sequence useful involves three basic steps: assembling the genome sequence, assigning gene structures, and assigning functions to genes. The quality of data generated at each step influences downstream inferences, with high-quality sequence, assembly, and gene structure assignments generally resulting in better functional annotations overall. Functional predictions serve as the basis for formulating hypotheses that are subsequently tested in the lab. As such, experimentalists have a great interest in high-quality functional annotation sets that cover all or most of the genes in their species of interest.

The Gene Ontology (GO) is a controlled vocabulary of hierarchically related terms that describe gene product function. It consists three categories: Biological Process (BP), Cellular Component (CC), and Molecular Function (MF). In the context of GO, functional annotation of a gene consists of the assignment of one or more GO terms from one or more of the GO categories to a given gene or gene model (here we will refer to genes and gene models simply as ’genes’ for simplicity).

For individual GO term associations to genes, Evidence Codes (ECs) are assigned to assert how the association of term to gene was made (Harris et al., 2004). GO evidence codes are aggregated into five general categories: Experimental, Computational Analysis, Curator Statement, Author Statement, and Automatically Assigned. See Table 4.1 and

The use of experimental ECs asserts that the assignment results from a physical characterization of the protein’s function as described in a publication. Computational approaches are based on in silico analyses. One of the simplest and most commonly conducted computational approaches involves matching similar genes between an existing, well-annotated genome and an unannotated genome. Once the matches are assigned, annotations are inferred to genes in the unannotated genome. Such assignments receive the ISS (Inferred from Sequence or Structural Similarity) EC. The ISS EC is also assigned if an uncharacterized sequence contains a characterized domain. In such instances, the presence of the domain itself can be used to predict function for the uncharacterized sequence. For Curator and Author Statements, included EC types are based on judgment by curators and scientists in their expert opinion. As such, they are considered to be reviewed annotation types, though these do include two ECs based on little data: NAS (Non-traceable Author Statement) and ND (No biological Data available). The Automatically Assigned EC type contains only one EC: Inferred from Electronic Annotation (IEA). IEA is unique in that no reviewed analysis of the assignment is required. Put another way, no curatorial judgment is applied, making it the least supported EC of the group.

Sequence-based approaches to automated functional annotation generally fall into three basic categories: sequence-similarity, domain-based methods, and mixed-methods. Sequence-similarity based gene matching most often relies on BLAST (e.g., BLAST2GO) followed by limiting the number of accepted matches based on e-value or a reciprocal-best-hit (RBH) strategy (Altschul, Gish, Miller, Myers, & Lipman, 1990; Conesa & Götz, 2008; Moreno-Hagelsieb & Latimer, 2008). Domain-based methods score sequences for the presence of well-
described protein domain such as those included in Pfam, PANTHER, and ProSite (Finn et al., 2017). InterProScan is a commonly used domain-based GO annotation pipeline (Jones et al., 2014). Mixed-methods combine sequence-similarity, domain-based approaches, and other evidence such as inferred orthology through phylogenetics to assign GO terms systematically (Clark & Radivojac, 2011; Falda et al., 2012; Koskinen, Törönen, Nokso-Koivisto, & Holm, 2015). For more of the latest methods, see (Jiang et al., 2016).

For maize, two genome-scale GO annotation sets exist for the B73 reference assembly and gene set (i.e., B73 RefGen_v3 and 5b+, respectively). These functional annotations are generated by and accessible from the Gramene (www.gramene.org; (Tello-Ruiz et al., 2016)) and Phytozome (phytozome.jgi.doe.gov; (Goodstein et al., 2012)) projects and websites, respectively. Gramene annotations are based on the Ensembl annotation pipeline (http://ensemblgenomes.org/info/data/cross_references), which is a mixed-method approach. The primary sources of the Ensembl annotations are from UniProtKB, community-based annotations from MaizeGDB (C. M. Andorf et al., 2016), InterPro2GO, and projections from orthologs inferred from phylogenetic analyses. Phytozome has a two-step process for GO annotation. First, Pfam domains are assigned to proteins. Second, GO annotations are determined based on the Pfam2GO mapping (Hunter et al., 2009).

Given the wealth of functional descriptions derived from mutational analyses, many researchers rely on the available maize GO-based functional annotations from large-scale, high-profile community resources like Gramene and Phytozome for formulating experimental hypotheses, and also as input datasets to transitively annotate predicted functions to newly sequenced grass species and crop genomes (e.g., (Hirsch et al., 2016)). However, if we compare the EC types for GO assignments between the model species Arabidopsis thaliana and the
Gramene and Phytozome functional annotations of the maize reference line B73, it is clear that the evidence supporting GO term assignments for these maize datasets is comparatively lacking (see Figure 4.1). Both the Gramene and Phytozome maize annotations have few annotations beyond those Inferred from Electronic Annotation (IEA). This situation is not intuitive to researchers given that maize has a wealth of functional descriptions in the literature.

Exacerbating this problem, transfers of predicted function often are based on sequence similarity alone with no restriction of input data to associations based on well-documented EC types. Furthermore, although mixed-method pipelines like the Ensemble COMPARA pipeline used by Gramene and the Phytozome Pfam2GO (Goodstein et al., 2012; Herrero et al., 2016) mappings may seem reproducible in principle given that they are based on the use of specific systems and software, details including input files and parameters often are unavailable or incomplete, making it impossible for research groups outside the group that generated those annotation resources to reproduce the annotation sets. In addition, because many computational pipelines inherit functional annotations that were also purely computationally derived, a single errant annotation can be propagated to many genomes (C. Andorf, Dobbs, & Honavar, 2007), making it mistakenly appear that many genomes agree on the errant function. For these reasons, existing computational functional annotations of maize (and many other plant genomes) should be approached with skepticism.

Given these issues with the maize functional annotation, we endeavored to create an improved annotation set. This task requires both application of robust and reproducible methods and a gold standard set of maize GO annotations to compare generated result sets to each other as well as to the Gramene and Phytozome maize functional annotations. One small dataset of well-curated GO-based functional annotations does exist for maize. It was initially created by
curators at MaizeGDB for the purpose of enriching the MaizeCyc metabolic pathway database (Monaco et al., 2013) and expanded through manual literature curation. This dataset constitutes 1,621 genes and 2,002 GO terms.

We annotated the maize B73 RefGen_v3 annotation set 5b+ using only experimentally-based annotations by filtering out GO assignments with IEA, NAS, and ND ECs from the input data and assigned GO terms using multiple input datasets then compared the performance of sequence-similarity, domain-presence, and mixed-methods based on how well the methods predicted function for genes included in the MaizeGDB gold standard dataset. For mixed-methods, we used pipelines developed for the Critical Assessment of Functional Annotation (CAFA) challenge, a competition designed to evaluate the latest computational functional annotation methods and to promote improvement of methods for functional annotation (Jiang et al., 2016; Radivojac et al., 2013). Groups competing in the CAFA challenge create tools that are applied to a set of specified target sequences. GO assignments are subsequently evaluated based on accumulation of functional data in the literature for the target sequence set. Some CAFA tools use pre-processing steps combined with a number of different computational and statistical approaches to reduce the number of false positive and false negative annotations (Clark & Radivojac, 2011; Falda et al., 2012; Koskinen et al., 2015). Some mixed-method pipelines performed better on average than other methods in the first iteration of the CAFA competition (Radivojac et al., 2013), indicating that the use of mixed-method pipelines for large scale GO annotations could potentially improve the overall quality of the annotation sets.

The project to evaluate and improve maize GO annotations is called GAMER: GO Annotation Method, Evaluation, and Review. We compared GAMER annotations to annotations based on sequence-similarity, domain, and three CAFA mixed-methods. Next we combined
GAMER outputs to generate an aggregate maize-GAMER GO annotation set and compared it to the existing Phytozome and Gramene GO annotations based on the $hF1$ score. The GAMER annotations had three major advantages compared to the Gramene and Phytozome annotations: (1) an increased number of maize genes annotated with GO terms; (2) more than twice the number of annotations (GO terms assigned) for maize protein coding genes; (3) similar or better quality scores relative to existing annotations sets based on $hF1$ score. The B73 RefGen_v3 5b+ maize-GAMER functional annotation dataset described here is accessible via MaizeGDB (http://download.maizegdb.org/maize-GAMER) and CyVerse (doi.org/10.7946/P2S62P). Scripts used to generate the annotation are available via GitHub at https://github.com/Dill-PICL/maize-GAMER.

Materials and Methods

Functional Annotation of Maize Genes

Three sequence-based approaches were used to annotate function to genes in the maize reference genome: sequence-similarity, domain-based, and mixed-method pipelines (see Figure 4.2; also described in the sections Sequence-similarity based annotation, Domain Presence, Mixed-Method Pipelines). The scripts (bash, R and Python) used to generate the annotations for maize B73 RefGen_v3 are available at https://github.com/Dill-PICL/maize-GAMER. These scripts run free and open-source tools on different inputs required for these tools to generate annotation datasets. Please refer to the reproducibility supplemental file for details on versions of software, version of input datasets used and commands and parameters used to run these tools.

The B73 genome and protein sequences for gene models included in the Filtered Gene Set (FGS) were downloaded from Gramene Release 42 (Tello-Ruiz et al., 2016). The downloaded protein FASTA file contained sequences for all FGS transcripts (e.g., the gene model X has transcript models X_T01, X_T02, and X_T03). For each gene model only the
longest translated protein sequence derived from the transcripts was analyzed. The gold standard annotations used for evaluations were obtained from MaizeGDB, and they encompass GO annotations for 1,619 gene models from RefGen_v3. The number of annotations for cellular component (CC), molecular function (MF), and biological process (BP) were 1,584, 88, 323 respectively.

**Sequence-similarity based annotation**

The sequence-similarity based annotation method has three main steps: 1) calculation of sequence-similarity, 2) valid hit detection, and 3) inheritance of high-confidence GO annotations. BLASTP was used (Altschul et al., 1990) with default parameters to calculate sequence-similarity between maize protein sequences and two other datasets: the "Arabidopsis" dataset from TAIR, The Arabidopsis Information Resource (Berardini et al., 2015) and the “Plant” dataset from UniProt (UniProt Consortium, 2015). Valid hits were detected using the RBH method from BLASTP results. GO terms with non-reviewed ECs (i.e., IEA, NAS, and ND described in the introduction) were removed from input datasets. All others were inherited between the RBH pairs of maize and the other plant.

Arabidopsis has the largest number of reviewed (human curated) EC GO annotations among plant model organisms (see Table 4.5). A FASTA file of Arabidopsis protein sequences along with the cognate GO Annotation File (GAF) were downloaded from TAIR v.10 (Berardini et al., 2015). The TAIR protein file contained predicted protein sequences from all transcripts. This file was filtered to retain only the protein sequence derived from longest transcript for each gene. Retained protein sequences from TAIR were used to create the TAIR BLAST database, and maize protein sequences were used to create a maize BLAST database. Maize protein sequences were used to query the TAIR BLAST database. Likewise, TAIR sequences were used to query the maize BLAST database. Results from both searches were used to detect RBH pairs
between Arabidopsis and maize. All non-reviewed EC GO annotations were removed, and remaining GO associations to Arabidopsis genes were inherited to maize genes for each RBH pair. This maize/Arabidopsis RBH ortholog dataset is called “maize-TAIR GO annotations”.

All reviewed EC GO annotations and protein sequences for all plants from the UniProt-GOA database were downloaded using the QuickGO tool hosted at EBI (Binns et al., 2009). Protein sequences and reviewed EC GO annotations were downloaded separately. The UniProt plant GO annotation dataset containing 304,426 annotations from 75,537 unique protein sequences. The protein sequences downloaded spanned 292 taxa. Only ten species had more than 1,000 annotations (see Table 4.5). Annotations from the top 10 species (in terms of number of reviewed GO annotations) were retained for our analyses. The process to annotate maize genes using UniProt plant data was similar to that for Arabidopsis. Maize protein sequences were matched against protein sequences from each species separately using BLASTP. Putative orthologs were determined using RBH for each maize-plant pair. Terms annotated to the other plant protein were inherited to the maize protein sequence for each putative ortholog pair. GO annotations inherited from each plant species were concatenated together. The derived dataset is called the “maize-UniProt GO annotations”.

**Domain Presence**

InterProScan5 (IPRS) version 5.16-55.0 was used to create domain based GO annotation of maize protein coding genes (Jones et al., 2014). IPRS was used to annotate GO terms to maize genes to produce the “maize-IPRS GO annotations”.

**Mixed-Method Pipelines**

At the beginning of this project, the first iteration of the CAFA challenge (CAFA1; described in the Introduction) had been completed. The results from the challenge indicated that CAFA1 mixed-method pipelines performed as well or better than standard methods (Radivojac
et al., 2013). To determine their predictive power for functional annotation in plants, the top-performing mixed-method pipelines from CAFA1 were reviewed to identify a group that could be implemented based upon availability of code and sufficient documentation. Three tools were selected: Argot2, FANN-GO, and PANNZER (Clark & Radivojac, 2011; Falda et al., 2012; Koskinen et al., 2015).

**Argot2**

Argot2 has a batch processing tool that can annotate up to 5,000 pre-processed input sequences. There are two different pre-processing steps for Argot2: 1) querying the UniProt database for sequence-similarity matches to the input sequences, and 2) querying the Pfam database for putative domains present in the input sequences. The maize sequences were split into multiple FASTA files containing a maximum of 5,000 sequences. The eight FASTA files resulting from the previous step were used to query the UniProt database using BLASTP for matches and the output was saved. HMMER was used to search a local Pfam database for potential hits for all the input protein sequences (Finn, Clements, & Eddy, 2011). Pre-processing each input FASTA resulted in a pair of input files for Argot2: BLAST and HMMER files. Each pair of pre-processed files was compressed and submitted to Argot2 batch processing tool. Results from each pair of pre-processed data were downloaded and concatenated to create the "maize-Argot2 GO annotations."

**FANN-GO**

The file containing maize protein sequences was imported into MATLAB using a built-in function (MATLAB:2017). The MAIN function from FANN-GO was used to pre-process and annotate maize protein sequences. FANN-GO uses BLASTP to query FANN-GO training sequence dataset (derived from UniProt) for potential matches for the input sequences and converts the results to input feature vectors. The FANN-GO predictor built from the training
dataset is then used to process the input feature vectors and calculate the probability that a particular protein is associated to a particular GO term. These probabilities are represented in a matrix where rows represent sequences and columns represent GO terms. The matrix was converted to a GAF (GO Annotation File Format) file to be used for subsequent evaluations. This dataset is referred to as the “maize-FANN-GO annotations”.

**PANNZER**

Maize protein sequences were pre-processed using BLASTP to query a local UniProt protein BLAST database, and the output was saved in XML format as required by PANNZER. PANNZER was run on the xml file output from the previous step, and the output was converted into a GAF file. This dataset from PANNZER is referred to as the "maize-PANNZER GO annotations."

**Metrics used in maize-GAMER**

A number of metrics defined and described by the AIGO (Analysis and the Inter-comparison of GO functional annotations) library were used to select high-confidence annotations, clean, and evaluate the maize-GAMER derived annotation sets (Defoin-Platel et al., 2011). AIGO has defined two type of metrics: analysis metrics and comparison metrics (see Table 4.2).

**Analysis Metrics**

Analysis metrics defined by AIGO measure features of a given annotation set. Four AIGO analysis metrics were used for maize-GAMER: Duplication, Redundancy, Coverage, and Specificity (see Table 4.2). With respect to calculating AIGO metrics, an annotation is defined as a single gene-GO term pair (Defoin-Platel et al., 2011). Duplication is the proportion of annotations that are not unique to a given annotation set. Duplication is calculated for each annotation set as described in Table 4.2. The collection of more general GO terms that can be
inherited from a specific GO term are called ‘ancestral’. Redundancy occurs when a GO term and one or more of its ancestral terms are annotated to the same gene in a dataset. Redundancy metric was calculated by obtaining the mean of the proportion of ancestral terms annotated for each gene in a dataset. Coverage is the proportion of genes that have at least one GO term assigned. Specificity is measured by counting the number of ancestral terms that exist for a given annotation, then averaging those counts across all annotations in the set. An ideally cleaned annotation dataset would have no duplication, no redundancy, high coverage, and high specificity.

**Comparison Metrics**

Comparison metrics defined by AIGO measure how well a given set of annotations match with another set of annotations. The AIGO comparison metrics hierarchical Precision ($hPr$) and hierarchical Recall ($hRc$) were used to evaluate annotation sets against gold standard annotations from MaizeGDB (see Table 4.2 & APPENDIX D). Different metrics have been defined for the evaluation of GO annotations against a gold standard (Clark & Radivojac, 2013; Defoin-Platel et al., 2011; Jiang et al., 2016; Radivojac et al., 2013). AIGO provided a set of well described evaluation metrics which were adapted by maize-GAMER and was utilized for large number of annotations produced by mixed-method pipelines. Both $hPr$ and $hRc$ evaluations start with propagating the GO terms in the annotations to the root. $hPr$ is the proportion of the GO terms (directly annotated and inferred by propagation) in an annotation set which is shared with the GO terms (directly annotated and inferred by propagation) in the gold standard. $hRc$ is the proportion of the GO terms in the gold standard which are found in the annotation set. $hPr$ and $hRc$ were calculated for the genes in the gold standard dataset and were calculated independently for each GO category. If a gene was annotated in the gold standard but was not annotated in the annotations set then both $hPr$ and $hRc$ were set to 0. See supplementary materials for precise
steps used to calculate $hPr$ and $hRc$. Harmonic mean ($hF_1$) of $hPr$ and $hRc$ was calculated for each annotation set for each GO category to use a single number to compare different annotation methods.

**Cleaning and Combining Component Datasets**

**Score threshold selection for mixed-methods**

Mixed-method pipelines used in the maize-GAMER project provide a confidence score for each GO annotation. The confidence score ranges from 0.0-1.0, where a higher score indicates more confidence for a given annotation. A score threshold which maximizes $hF_1$ ($hF_{max}$) will select the optimal set of annotations which reduces the total number of false-positives and false-negatives (see Metrics used in maize-GAMER section for the description of the metrics). The range of annotation scores from mixed-method pipelines did not span the whole 0.0-1.0 range, so the scores were normalized to fall between 0.0-1.0 independently for each annotation set. A set of thresholds (every 0.05 from 0.0 to 1.0; i.e. 0.00, 0.05, 0.1, 0.1, ...., 0.95, 1.00) were selected. $hF_1$ score for each GO category and each threshold was calculated by selecting the annotations with a normalized score that was ≥ to the threshold and evaluating against the gold standard annotations. $hF_{max}$ for each GO category was determined by getting the highest $hF_1$ obtained from the previous step. The score thresholds which resulted in $hF_{max}$ were used to select a subset of annotations from each mixed-method pipeline (see Table 4.6). The maize-Argot2, maize-FANN-GO, maize-PANNZER GO annotations described in subsequent sections refer to the subset of annotations selected via this selection step.

**Removing Redundancy and Duplication**

Duplication is the presence of two or more instances of the same gene-GO term pair in a single annotation set (Defoin-Platel et al., 2011). Redundancy is the presence of an ancestral GO
term in the annotations of a gene which also contains a specific annotation from which the ancestral GO term can be inferred by propagation (Defoin-Platel et al., 2011). Component annotation sets from all methods described above were cleaned by removing redundancy and duplication for each annotation set across all three GO categories. Duplication was cleaned by replacing multiple instances of a gene-GO term pair with a single instance for a given annotation set. Duplicate annotations from all six raw annotation sets were removed and files with non-duplicate annotations were created for each annotation set. Redundancy was cleaned by removing annotations containing GO terms that could be inferred from other terms based on the GO hierarchy, and only retaining the annotations with GO terms that cannot be inferred.

The maize-GAMER Aggregate Dataset

Clean (non-redundant and non-duplicated) annotation sets from all component methods were merged to generate the maize-GAMER aggregate annotation set. Redundancy and duplication introduced by concatenating multiple datasets were removed.

A new genome assembly (B73 RefGen_v4) and annotation set (Zm00001.2) for maize inbred line B73 was recently released (Jiao et al., 2017). Because this dataset has not been available for long, only few published analyses are available and the research community is only now in the process of transitioning to general use of RefGen_v4 for large-scale analyses. As such, analyses and results described here derive from the well-annotated v3 assembly and annotation set. To extend outcomes of the work described here for future v4 efforts, maize-GAMER aggregate annotations have also been created for the maize B73 RefGen_v4, which can be accessed at MaizeGDB (http://download.maizegdb.org/maize-GAMER) and via CyVerse (doi.org/10.7946/P2M925).
Evaluation of GAMER-derived Annotation Sets

Component and aggregate annotation sets were compared at two levels; a general comparison, and a GO category-specific comparison.

Comparison metrics mentioned in Metrics used in maize-GAMER Section were calculated for the general comparison (see Table 4.2). All metrics were calculated independently for each annotation set, and compared among component annotation sets as well as the aggregate annotation set. Coverage and the number of annotations were calculated directly for each annotation set. Specificity was calculated for each annotation and the mean across all annotations is reported.

The annotations from component annotation sets and the aggregate annotation set were divided into specific GO categories and category-specific annotations were evaluated separately. Three different metrics were used for GO category-specific evaluations: coverage, number of annotations, and $hF_1$ (see section Metrics used in maize-GAMER for more details). Coverage and the number of annotations were calculated individually for each GO category for each dataset. $hF_1$ score was calculated for each annotation set for each GO category.

Comparisons among the maize-GAMER Aggregate, Gramene, and Phytozome Annotation Sets

The existing Gramene, Phytozome, and maize-GAMER annotations were compared to each other. Redundancy and duplication were removed from the Gramene and Phytozome annotation sets before evaluations were performed. Evaluation and comparisons were identical to the analyses performed in the previous section. General evaluations for the maize-GAMER, Gramene, and Phytozome annotation sets were based on coverage, number of annotations, and specificity. These metrics were calculated as described in the previous section. The Gramene, Phytozome, and maize-GAMER annotation sets were also compared in a GO category-specific
manner to account for biases in performance among different categories (i.e., CC, BP, and MF). Comparisons were made based on coverage, number of annotations, and mean $hF_1$ score.

**Case Study of the Gene *nana plant1 (na1)***

The gene *na1* (GRMZM2G449033) had the most terms (7 GO terms) associated with it in the gold standard dataset, and all the terms were from BP GO category. Annotations for *na1* from the three maize annotation sets were obtained. The ancestral nodes were inferred from the leaf nodes for each annotation set, and a subgraph for the BP ontology was generated (see Figure 4.5). The nodes in the subgraph were compared to gold standard and nodes shared between a given annotation set and the gold standard dataset were identified. Nodes exclusively found only in a given annotation set or the gold standard were also identified. Illustrations of the subgraphs without node labels were drawn to compare among the three different GO annotation sets.

**Results**

**Evaluation of maize-GAMER Derived Component Annotation Sets**

The maize-GAMER derived component annotation sets (i.e., the TAIR, UniProt, IPRS, Argot2, FANN-GO, and PANNZER) and the maize-GAMER aggregate annotation set were evaluated across GO categories as well as within each GO category using metrics described in Table 4.2.

**General Evaluation of maize-GAMER Component Annotation Sets**

Initial evaluations and comparisons of datasets created by the maize-GAMER pipeline were assessed based on coverage, number of annotations, and specificity among all clean component annotation sets as well as the aggregate annotation set (see Table 4.2 & Table 4.3). The specificity and redundancy have been described in methods and Table 4.2. The TAIR and UniProt annotation sets had the lowest coverage and number of annotations among all maize-GAMER component annotation sets (Table 4.3). The Argot2, FANN-GO, and PANNZER
annotation sets had the highest number of annotations compared to other annotation sets, as well as higher coverage compared to other annotation sets. Notably, FANN-GO had the highest coverage at 100% of genes, and Argot2 had annotations for more than 90% of the genes. The IPRS annotation set had a lower number of annotations compared to the CAFA mixed-method pipelines but covered more genes than sequence-similarity methods. Although sequence-similarity methods and IPRS covered a lower number of genes, they had higher specificity compared to mixed-method pipelines in general. Of the three mixed-method pipelines, only PANNZER had comparable specificity to the methods, but had lower coverage than both Argot2 and FANN-GO. Both Argot2 and FANN-GO had lower average specificity but had higher coverage than other methods. The maize-GAMER aggregate annotation (made up of all component annotation sets) covered all maize genes with at least one annotation (as expected given that the FANN-GO component annotation set also covers 100% of genes). In addition, the aggregate annotation set contains more than double the number of annotations that occur in any component annotation set. This indicates that different component methods assign different GO terms to genes. Therefore, combining annotations from different methods results in increased diversity of GO term assignments. Moreover, the aggregate annotation set has higher specificity than the mixed-method pipelines which have higher coverage but has lower specificity than all other component annotation sets.

Genes that are annotated with at least one GO term from each component annotation set were compared among the three different method types (i.e., sequence-similarity, domain-based, and mixed-methods; see Figure 4.3a). This comparison revealed that less than a quarter of genes had been annotated by all three methods, but more than half were annotated by two different methods. The remainder were only annotated by mixed-method pipelines. Sequence-similarity
and domain-based methods resulted in annotations to genes that were also annotated by mixed-method pipelines. The number of genes annotated by domain-based methods and mixed-method pipelines, and are not annotated by sequence-similarity based methods are higher than genes annotated by all three methods. In contrast, sequence-similarity methods shared more genes with both other annotation sets than only with mixed-method pipelines. Moreover, only mixed-method pipelines annotate at least one GO term to all genes in the maize FGS.

Although the mixed-method pipelines annotated all genes, they did not capture all GO terms annotated to genes by the other methods. GO term assignments were compared to evaluate the diversity of the GO terms present in the three types of GO annotation methods (see Figure 4.3b). GO terms annotated directly and ancestral terms inferred from the direct terms annotated to genes were compared among the three GO annotation methods used in maize-GAMER. The number of GO terms annotated by sequence-similarity, domain-based, and mixed-method pipelines were 3,794, 8,145, and 14,225, respectively. The number of GO terms annotated by the mixed-method pipelines are significantly higher than both other methods, however there are a small number of GO terms that are only annotated by sequence-similarity (721) and domain-presence (16) methods. Only a small proportion (23.05%) of the total (15,028) GO terms are annotated by all three methods.

**GO Category-specific Evaluations of maize-GAMER Component Annotation Sets**

CAFA1 indicated that annotations for some GO categories are easier to predict than others (Radivojac et al., 2013). This indicated that the GO category specific evaluations could provide a more accurate comparison between component methods. This would also allow unbiased comparison of tools which do not predict certain categories (e.g., FANN-GO doesn’t predict the CC category). Therefore, maize-GAMER derived annotation sets were divided into
specific GO categories (i.e., CC, BP, and MF) and each category was evaluated separately based on coverage, number of annotations, and \( hF1 \).

Mixed-method pipelines had higher coverage across all three GO categories. Argot2 covered more than 80% genes across all three categories (see Figure 4.3c). FANN-GO does not annotate GO terms for CC category, but had 100% coverage in BP category, and covered about 50% genes in MF category. PANNZER had the lowest coverage compared to the other mixed-method pipelines, and covered only 30-50% of genes across different categories, and had highest coverage in BP. Sequence-similarity methods consistently had lowest coverage compared to other methods in BP and MF, but IPRS had the lowest coverage in CC. IPRS covered higher number of genes than sequence-similarity methods in BP and MF, but had lower coverage than mixed-method pipelines. When comparing IPRS coverage across three GO categories, the coverage was highest in MF. Aggregate annotation set covered slightly more genes than the component annotation sets with highest coverage in each category, and covered more than 88% of the FGS genes in all categories. In the BP category, the aggregate annotation set annotated all genes from maize FGS with at least one annotation.

Mixed-method pipelines produce a higher number of annotations than other methods in all three GO categories. Moreover, the number of annotations from mixed-method pipelines loosely correlate with coverage in different GO categories. The only exception was PANNZER, which annotated more GO terms per gene in BP category (data not shown), than any other component annotation set. The number of annotations from sequence-similarity methods and IPRS were consistently lower than mixed-method pipelines. The variation in the number of annotations was proportional to the number of genes annotated in sequence-similarity and IPRS methods. The lowest number of annotations was seen in the CC category from IPRS, and
sequence-similarity methods in other GO categories. As the union of all component method annotations, the aggregate annotation set had a higher number of annotations in all three GO categories. The highest number of annotations for the aggregate annotation set was from the BP GO category, followed by CC and MF.

We used $hF_1$ scores as a representation of the quality of annotations and annotation sets. As described in materials and methods, the $hF_1$ was calculated individually for all genes in the gold standard dataset and then averaged across all genes within an annotation set. There are clear differences in $hF_1$ across different GO categories. The highest performance was seen in the MF category, and the lowest performance is seen in the BP category. This fits the observation from CAFA1 (Radivojac et al., 2013). Mixed-method pipelines outperformed other methods in all three GO categories. PANNZER produced the highest $hF_1$ within the MF category, but Argot2 had the highest $hF_1$ scores in CC and BP. IPRS outperformed sequence-similarity methods in both MF and BP categories, but was the lowest performing method in the CC category. Comparison between two sequence-similarity methods indicated that maize-UniProt method performs better than the maize-TAIR method in MF and BP categories. On the other hand, maize-TAIR method performs better than maize-UniProt method in the CC category. Aggregating the component annotations from maize-GAMER increased the performance in the CC category. In contrast, aggregating the component annotation sets did not increase the performance compared to the top performing tool in other categories.

**Evaluation of Existing Maize GO Annotation Sets and Comparison to the maize-GAMER Aggregate Annotation Set**

Two existing maize GO annotation sets, Gramene and Phytozome, were downloaded, evaluated, cleaned (i.e., redundancies and duplicates were removed), and compared with maize-GAMER aggregate annotations (referred to as the “maize-GAMER annotation set”). The same
metrics used for the evaluation of maize-GAMER derived annotation sets were used for the comparison among the existing maize GO annotation sets and maize-GAMER aggregate annotation set.

**General Evaluation of Public Maize GO Annotation Sets**

The maize-GAMER aggregate annotations covered all genes in the maize FGS with at least one GO term, but Gramene and Phytozome covered only about half the genes (see Table 4.4). Phytozome covered the fewest genes (less than half of the genes), and Gramene covered slightly more than half of the genes (see Table 4.4). The maize-GAMER annotation set had more annotations than both Gramene and Phytozome. Gramene had two-fold more annotations than Phytozome, and the maize-GAMER had several fold more annotations than Gramene. While the maize-GAMER has higher coverage and a higher number of annotations, it has lower average specificity than Gramene and Phytozome. Gramene has the highest average specificity of all three annotation sets.

Genes with annotations from each set were compared to see the distribution of annotated genes among different annotations (see Figure 4.4a). Genes from Gramene and Phytozome annotations were a subset of the maize-GAMER annotations. Less than half of the genes were annotated in all three sets, and slightly more than half of the genes were annotated in at least two sets. Comparison of Gramene and Phytozome annotations show that most of the genes that were annotated were shared. Both Gramene and Phytozome had genes that were annotated in only one of the two (i.e., Gramene or Phytozome but not both; See Figure 4.4a).

GO terms annotated directly to genes by different methods and ancestral GO terms propagated from these annotations were compared among the three annotation sets. The number of GO terms annotated in each set varied greatly. The least diverse set in terms of number of GO terms annotated was Phytozome, which was annotated with only 3,234 GO terms (approximately
7% of total GO terms). Gramene has annotated 7,215 GO terms (approximately 16%), and was more diverse than Phytozome, but had lower diversity than maize-GAMER. maize-GAMER had the highest diversity and contained 15,028 GO terms (approximately 33%). A small number of GO terms were used by all three annotation sets, and most of the terms from Phytozome were shared across all three annotation sets. Only a single GO term was exclusive to the Phytozome annotations, and small number of terms were found to be exclusive to Gramene annotations. Approximately 50% of the GO terms from maize-GAMER were unique. The maize-GAMER aggregate annotations shared a higher number of GO terms with Gramene than Phytozome.

**GO Category-specific Evaluations of maize-GAMER and Existing Maize GO Annotation Sets**

Annotations from the three maize GO annotation sets were analyzed in a GO category-specific manner to identify differences in performance among the different categories (see Figure 4.4). As was true for the component annotation sets, three different metrics were used for evaluation and comparison: coverage, number of annotations, and $hF_1$ score.

Comparison of coverage across GO categories indicated that all annotation sets had lower coverage in CC category, compared to other categories. Both Gramene and Phytozome had lower coverage in BP than MF, but maize-GAMER had higher coverage in BP than MF. Lowest coverage for all annotation sets and categories was seen in the CC category for the Phytozome annotation set, and the highest coverage was seen in the maize-GAMER aggregate annotation set in the BP category. Comparison among the three maize annotation sets indicates that the maize-GAMER annotation set had the highest coverage in all three categories by a large margin. Coverage from maize-GAMER was almost twofold that of Gramene, which had the next highest coverage in all GO categories. Gramene had higher coverage than Phytozome in all three categories.
When the number of annotations were compared across different GO categories the lowest number of annotations for Gramene and Phytozome annotation sets were seen in the CC category. In contrast, maize-GAMER had the lowest number of annotations in the MF category. Moreover, both Gramene and Phytozome both had a higher number of annotations in the MF category whereas maize-GAMER had the highest number of annotations in the BP category. Comparison among the annotation sets illustrated that the maize-GAMER annotation set has the highest number of annotations in all three categories. Phytozome had the fewest annotations in all three GO categories. Number of annotations loosely correlated with coverage in different GO categories for both Gramene and Phytozome. Furthermore, maize-GAMER had the highest number of annotations in the BP category. The number of annotations from the maize-GAMER annotations for the BP was severalfold higher than other annotation sets (approximately 9x that of Gramene and approximately 28x that of Phytozome).

The $hF_1$ score reflects the overall quality of annotations produced by different pipelines used by the three maize annotation projects. Comparing performance of different pipelines across the three GO categories revealed a similar trend that was seen in the previous section. All pipelines had higher $hF_1$ scores in the MF category and had lower $hF_1$ scores in the BP category. The only pipeline that did not fit this trend was Phytozome, which had lowest performance in the CC category. maize-GAMER had a higher $hF_1$ score than other pipelines in the CC category. maize-GAMER also had higher performance than Phytozome in other categories but performed slightly lower than Gramene in those categories. Gramene performed better than other pipelines in the MF and BP categories. Phytozome consistently had lower performance than other pipelines across all three GO categories. Phytozome’s performance was
especially low in the CC category, which was the lowest $hF_1$ score seen for any annotation set in any GO category.

To visualize how GO annotation methods perform comparatively, the distribution of metrics $hPr$, $hRc$, and $hF_1$ can be calculated across all genes included in the gold standard dataset. In Figure 4.6, lower coverage by sequence-similarity and domain-presence methods is illustrated by the high number of genes with a value of zero. In general, mixed-method pipelines are shown to perform better than other methods we used. They cover more genes, and two of the three mixed-method pipelines have higher $hPr$ and $hRc$ than all other methods. The FANN-GO distribution is different from both other mixed-methods: in general, it has lower performance than other methods. This can be attributed to the fact FANN-GO annotations have lower specificity than other methods, and lower specificity results in lower values for $hPr$ and $hRc$. In addition, the maize-GAMER has fewer genes with a value of 0 than both Phytozome and Gramene.

**Example Annotations from *nana plant1* (na1)**

The gene *nana plant1* (na1; GRMZM2G449033) has more annotations than any other gene in the gold standard dataset. A classical maize mutant with a dwarf phenotype (Hartwig et al., 2011), the *na1* recessive mutant results from a loss-of-function mutation in the gene that affects the brassinosteroid (BR) biosynthetic pathway where BR is a plant hormone that is required for normal plant growth (Hartwig et al., 2011). In the gold standard dataset, *na1* had 7 biological process GO terms annotated. Annotations for *na1* from different maize annotation sets were compared to the gold standard, and a subgraph for each annotation set and gold standard dataset was plotted (see Figure 4.5). Phytozome did not annotate any GO terms to *na1* (see Figure 4.5a), but both Gramene (see Figure 4.5b) and maize-GAMER (see Figure 4.5c) have
annotated BP GO terms for *na1*. Gramene annotates 3 GO terms *na1* while maize-GAMER has annotated 13 GO terms to *na1*. Two GO terms from the gold standard are known to be related to *na1* dwarf phenotype from previous studies, "brassinosteroid biosynthetic process" (GO:0016132) and "unidimensional cell growth" (GO:0009826). While both of these were annotated correctly by maize-GAMER (see Figure 4.5c), only one of them was correctly annotated by Gramene (see Figure 4.5b). Comparison of overlapping nodes indicates that the maize-GAMER aggregate annotation set also contains a number of less specific non-leaf terms which overlap with nodes inferred from gold standard dataset. Overall, the maize-GAMER has larger proportion of overlapping nodes with the gold standard than the Gramene for the BP GO category.

The different approaches taken by the pipelines from Gramene and maize-GAMER result in different annotations for the example case study of the maize *na1* gene. Gramene has a lower number of GO terms annotated to *na1* than maize-GAMER. The average specificity of GO terms annotated in the BP category for *na1* (see Figure 4.5) is not significantly different between GAMER (mean=12.154) and Gramene (mean=12.667) pipelines (2-sided 2-group Wilcoxon rank-sum test; p = 0.89). This example from *na1* indicates that the specificity of the annotations are not significantly different for specific instances, but are different when compared overall. We further compared the GAMER component and aggregate dataset creation methodologies annotate gene function to that of Gramene and Phytozome by comparing the hPr, hRc, and hF₁ metrics for the *na1* gene (See Supplementary Figure 4.7). Phytozome’s method stands out because it has no annotations for *na1*, thus all the metrics have a value of 0. The GAMER aggregate dataset has higher hF₁ score compared to Gramene, and has higher hPr and hRc as well. Among the component datasets, Pannzer has the highest hF₁ score and highest hRc.
InterProScan has the highest $hPr$, but its lower $hRc$ reduces the $hF_1$. FANN-GO and Argot2 have lower metrics due to lower specificity of annotations compared to other methods.

**Discussion**

In keeping with our goal, through the maize-GAMER project we were able to improve the GO annotation dataset for maize and to document inputs, methods, and results at a level that enables both reproducibility and reuse of the pipeline for future genome versions.

To determine how best to create an improved maize annotation dataset, we tried out multiple different methods and compared the resulting datasets based on a gold standard set of gene functions. This also enabled us to better understand differences in term assignments among the methods we used. Using the same gold standard, we also were able to compare resulting datasets to those produced by and available from Gramene and Phytozone.

We used the $hF_{max}$ metric to select high-confidence annotations from mixed-method pipelines and to evaluate annotation sets resulting from all methods under evaluation. We found that mixed-method pipelines developed for the CAFA1 challenge outperformed RBH and domain-presence methods for GO annotation (Radivojac et al., 2013). They covered more genes with annotations, produced higher number of annotations, and had higher $hF_1$ score than both sequence similarity and domain-based methods. The higher performance from mixed-method pipelines are the outcome of advanced statistical (Falda et al., 2012; Koskinen et al., 2015) and machine learning approaches (Clark & Radivojac, 2011) used to reduce the false positive and false negative annotations. Mixed-method pipelines do have a limitation: they have higher coverage but annotations are less specific in general when compared with datasets produced using other approaches. This could be due to the dearth of training dataset for the more specific GO terms, which is required for training machine learning methods.
When we aggregated the predictions from RBH, domain-based methods, and three tools from CAFA1, we produced the maize-GAMER aggregate dataset, which covers more gene space than the datasets produced by Gramene and Phytozome, and with similar or better accuracy. While the higher coverage could cause concern, evaluating the annotations using the gold standard has shown that the performance is similar or better than existing datasets. This also indicates that maize-GAMER annotations are no less reliable than Gramene and are in fact are better than those for Phytozome. Removing less specific GO terms annotated by some methods in cases where there were more specific GO terms annotated to the same genes was important for aggregating different datasets. In certain cases, more than one annotations with lower specificity were replaced by a single annotation with higher specificity. As with any computational approach to annotate GO terms, the current maize-GAMER dataset should be considered as an initial step in improving the GO annotations in maize. As future iterations of the CAFA competition evaluate new tools and methods for GO annotations, we anticipate that the quality of computational maize GO annotations could be iteratively improved in a reproducible manner by continuing to apply the newest, best performing methods.

To enable better reproducibility, we have generated a supplementary document with exact parameters and commands used to generate the maize dataset. We are currently in the process of formalizing the code used to generate the maize GO annotation set into a reusable pipeline called GO-MAP. Once completed, the GO-MAP pipeline can be used for GO annotation of newly sequenced plant genomes as well as existing plant genomes. The pipeline will be made freely available and will utilize the same methods and datasets used for maize.

The set of manually reviewed gene function annotations for maize that we call the gold standard is both incomplete and sparse. This situation does not reflect the amount of published
literature describing gene function for maize. Instead, this situation is due to limited curation of gene function into GO terms. While tools exist at MaizeGDB that enable researchers to assign GO terms to genes directly, these tools remain poorly utilized. In an effort to improve community engagement and to upgrade the evidence codes for GO assignments, our next step for maize-GAMER will be to develop and deploy a tool to enable experts in the maize community to review existing GO annotations. By enabling GO annotation review through expert crowdsourcing, term assignments produced by computational pipelines including GAMER can be upgraded from IEA (inferred from electronic annotation) to RCA (reviewed computational analysis). In this way, we will enable the transfer of collective knowledge members of the maize community have generated over the years to produce higher-quality functional annotation datasets for maize with clear extension of this practice for other species.

**Author Contributions**

KW and CJLD conceptualized the methodology, and KW performed formal analysis. CMA provided feedback about the methodology and provided resources for evaluation of the results. CJLD supervised the work and obtained funding for the project. First draft was prepared by KW, and edited by CMA, IF and CJLD. The final manuscript was reviewed and approved by all authors.

**Acknowledgements**

We thank Darwin Campbell and Ramona Walls for efforts to create the DOIs for datasets available via CyVerse and John Portwood and Ethalinda Cannon for their efforts to populate resulting datasets for MaizeGDB. Thanks to members of the Dill Plant Informatics and Computation Lab (dill-picl.org) for critical review and helpful suggestions, with particular thanks to Darwin Campbell for help with figures. This work was supported by funding from the Iowa State University Plant Sciences Institute Faculty Scholars Program to C.J.L.D.; the
National Science Foundation [IOS #1027527] to C.J.L.D.; and the United States Department of Agriculture - Agricultural Research Services to C.M.A. IF was supported, in part, by National Science Foundation award ABI-1458359.
Table 4.1: Evidence Codes used in gene ontology annotations

<table>
<thead>
<tr>
<th>Type</th>
<th>Evidence Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental</td>
<td>EXP</td>
<td>Inferred from Experiment</td>
</tr>
<tr>
<td></td>
<td>IDA</td>
<td>Inferred from Direct Assay</td>
</tr>
<tr>
<td></td>
<td>IPI</td>
<td>Inferred from Physical Interaction</td>
</tr>
<tr>
<td></td>
<td>IMP</td>
<td>Inferred from Mutant Phenotype</td>
</tr>
<tr>
<td></td>
<td>IGI</td>
<td>Inferred from Genetic Interaction</td>
</tr>
<tr>
<td></td>
<td>IEP</td>
<td>Inferred from Expression Pattern</td>
</tr>
<tr>
<td>Computational Analysis</td>
<td>ISS</td>
<td>Inferred from Sequence or structural Similarity</td>
</tr>
<tr>
<td></td>
<td>ISO</td>
<td>Inferred from Sequence Orthology</td>
</tr>
<tr>
<td></td>
<td>ISA</td>
<td>Inferred from Sequence Alignment</td>
</tr>
<tr>
<td></td>
<td>ISM</td>
<td>Inferred from Sequence Model</td>
</tr>
<tr>
<td></td>
<td>IGC</td>
<td>Inferred from Genomic Context</td>
</tr>
<tr>
<td></td>
<td>IBA</td>
<td>Inferred from Biological aspect of Ancestor</td>
</tr>
<tr>
<td></td>
<td>IBD</td>
<td>Inferred from Biological aspect of Descendant</td>
</tr>
<tr>
<td></td>
<td>IKR</td>
<td>Inferred from Key Residues</td>
</tr>
<tr>
<td></td>
<td>IRD</td>
<td>Inferred from Rapid Divergence</td>
</tr>
<tr>
<td></td>
<td>RCA</td>
<td>Inferred from Reviewed Computational Analysis</td>
</tr>
<tr>
<td>Author statement</td>
<td>TAS</td>
<td>Traceable Author Statement</td>
</tr>
<tr>
<td>Curatorial Statement</td>
<td>NAS*</td>
<td>Non-traceable Author Statement</td>
</tr>
<tr>
<td></td>
<td>IC</td>
<td>Inferred by Curator</td>
</tr>
<tr>
<td>Automatically Assigned</td>
<td>IEA*</td>
<td>Inferred from Electronic Annotation</td>
</tr>
</tbody>
</table>

* Indicates evidence codes without either curation or biological data supporting them.
Table 4.2: Metrics used to analyze and evaluate the annotation datasets

<table>
<thead>
<tr>
<th>Analysis Metrics*</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Metric</strong></td>
</tr>
<tr>
<td>Duplication (%)</td>
</tr>
<tr>
<td>Redundancy (%)</td>
</tr>
<tr>
<td>Coverage (%)</td>
</tr>
<tr>
<td>Specificity</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Metric</strong></td>
</tr>
<tr>
<td>( hPr )</td>
</tr>
<tr>
<td>( hRc )</td>
</tr>
<tr>
<td>( hF1 )</td>
</tr>
</tbody>
</table>

*Analysis and Comparison metrics used here have been described in detail in methods (Defoin-Platel et al., 2011).
†See APPENDIX D. for precise steps to calculate \( hPr \) and \( hRc \).
Notations used as follows, \( N \): Total # of Annotations, \( M \): Total # of Genes, \( AS \): Annotation Set, \( GS \): Gold Standard.
Table 4.3: Results from the general evaluation of maize-GAMER derived annotation sets

<table>
<thead>
<tr>
<th>Method Type</th>
<th>Sequence-Similarity</th>
<th>Domain</th>
<th>CAFA Mixed-Methods</th>
<th>Union</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotation Set</td>
<td>TAIR-RBH</td>
<td>Plant-RBH</td>
<td>IPRS</td>
<td>Argot2</td>
</tr>
<tr>
<td>Raw Annotations</td>
<td>61,528</td>
<td>106,053</td>
<td>200,324</td>
<td>450,013</td>
</tr>
<tr>
<td>Duplication (%)</td>
<td>9.11</td>
<td>65.89</td>
<td>72.18</td>
<td>0.33</td>
</tr>
<tr>
<td>Redundancy (%)</td>
<td>14.18</td>
<td>9.17</td>
<td>10.91</td>
<td>48.25</td>
</tr>
<tr>
<td>Clean Annotations</td>
<td>35,791</td>
<td>32,085</td>
<td>46,599</td>
<td>224,827</td>
</tr>
<tr>
<td>Coverage (%)</td>
<td>23.70</td>
<td>20.10</td>
<td>48.48</td>
<td>92.64</td>
</tr>
<tr>
<td>Specificity</td>
<td>11.54</td>
<td>12.20</td>
<td>10.45</td>
<td>8.54</td>
</tr>
</tbody>
</table>

Table 4.4: Overall results from maize-GAMER and other existing maize datasets

<table>
<thead>
<tr>
<th>Analysis Metric</th>
<th>Existing Gramene</th>
<th>Existing Phytozone</th>
<th>maize-GAMER aggregate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Annotations</td>
<td>111,203</td>
<td>66,709</td>
<td></td>
</tr>
<tr>
<td>Duplication (%)</td>
<td>0.00</td>
<td>37.90</td>
<td></td>
</tr>
<tr>
<td>Redundancy (%)</td>
<td>23.25</td>
<td>7.24</td>
<td></td>
</tr>
<tr>
<td>Clean Annotations</td>
<td>81,315</td>
<td>36,987</td>
<td>515,059</td>
</tr>
<tr>
<td>Coverage (%)</td>
<td>55.55</td>
<td>40.87</td>
<td>100.00</td>
</tr>
<tr>
<td>Specificity</td>
<td>10.90</td>
<td>10.41</td>
<td>9.56</td>
</tr>
</tbody>
</table>
Table 4.5: Top 10 plants by number of high-confidence GO annotations in UniProt-GOA

<table>
<thead>
<tr>
<th>Rank</th>
<th>Species</th>
<th>Proteins</th>
<th>Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><em>Arabidopsis thaliana</em></td>
<td>3,702</td>
<td>72,089</td>
</tr>
<tr>
<td>2</td>
<td><em>Glycine max</em></td>
<td>3,847</td>
<td>43,202</td>
</tr>
<tr>
<td>3</td>
<td><em>Oryza sativa (Japonica)</em></td>
<td>39,947</td>
<td>32,750</td>
</tr>
<tr>
<td>4</td>
<td><em>Populus trichocarpa</em></td>
<td>3,694</td>
<td>31,851</td>
</tr>
<tr>
<td>5</td>
<td><em>Solanum lycopersicum</em></td>
<td>4,081</td>
<td>24,250</td>
</tr>
<tr>
<td>6</td>
<td><em>Sorghum bicolor</em></td>
<td>4,558</td>
<td>23,470</td>
</tr>
<tr>
<td>7</td>
<td><em>Vitis vinifera</em></td>
<td>29,760</td>
<td>23,350</td>
</tr>
<tr>
<td>8</td>
<td><em>Brachypodium distachyon</em></td>
<td>15,368</td>
<td>22,454</td>
</tr>
<tr>
<td>9</td>
<td><em>Physcomitrella patens</em></td>
<td>3,218</td>
<td>18,348</td>
</tr>
<tr>
<td>10</td>
<td><em>Chlamydomonas reinhardtii</em></td>
<td>3,055</td>
<td>9,826</td>
</tr>
</tbody>
</table>

Table 4.6: $hF_{max}$ and score thresholds for mixed-method pipelines by GO categories

<table>
<thead>
<tr>
<th>Cellular Component</th>
<th>Pipeline</th>
<th>$hF_{max}$</th>
<th>Pipeline Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Argot2</td>
<td>0.572</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>FANN-GO</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>PANNZER</td>
<td>0.460</td>
<td>0.20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Molecular Function</th>
<th>Pipeline</th>
<th>$hF_{max}$</th>
<th>Pipeline Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Argot2</td>
<td>0.584</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>FANN-GO</td>
<td>0.582</td>
<td>0.65</td>
</tr>
<tr>
<td></td>
<td>PANNZER</td>
<td>0.607</td>
<td>0.55</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Biological Process</th>
<th>Pipeline</th>
<th>$hF_{max}$</th>
<th>Pipeline Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Argot2</td>
<td>0.300</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>FANN-GO</td>
<td>0.272</td>
<td>0.30</td>
</tr>
<tr>
<td></td>
<td>PANNZER</td>
<td>0.241</td>
<td>0.40</td>
</tr>
</tbody>
</table>

Mixed-method pipeline scores that result in the max $hF_1$ calculated for each pipeline for each GO category. The threshold values shown here were used to select high confidence GO annotations from each pipeline and only annotations with a score $\geq$ threshold were selected. *It is important to note that absolute values from the pipelines were normalized between 0-1 before $hF_1$ scores were calculated and $hF_{max}$ was determined.* NA indicates that the pipeline did not annotate terms in the GO category.
Figures and Legends

Figure 4.1: Numbers of annotations by EC category
Arabidopsis (TAIR10) shown in magenta, maize in green and orange for Gramene and Phytozome, respectively. Annotation counts on the y-axis are shown in thousands. Each bar in the histogram is labeled with the actual count to show where counts are so small that no bar is visible.
Figure 4.2: Overview of steps to produce the maize-GAMER datasets.

Three types of methods are used: sequence-similarity (yellow), domain presence, (blue) and CAFA mixed-method pipelines (green). Within sequence-similarity, two input datasets were subjected to reciprocal-best-hit against maize: TAIR10 (Arabidopsis) and UniProt (the ten most well-annotated plant species). For domain presence, InterPro signatures were applied to maize using InterProScan (IPRS). From the CAFA mixed-method pipelines, Argot2, FANN-GO, and PANNZER were applied to maize. For each individual output, duplications and redundancies were removed, then the datasets were combined. A second round of duplication and redundancy removal was carried out to produce the maize-GAMER Aggregate dataset.
Figure 4.3: GO assignment metrics for each method type

Sequence-similarity in yellow, domain presence in blue, and mixed-method pipeline in green. (a) Number of genes with at least one GO term annotated. (b) Number of GO terms with at least one gene annotated. (c) Percent coverage, number of annotations, and average $hF_1$ score for each annotation set across the three GO graphs (i.e., Cellular Component, Molecular Function, and Biological Process). Color codes as used in (a) and (b), with the aggregate dataset shown in orange.
Figure 4.4: GO assignment metrics for Gramene, Phytozome, and maize-GAMER

Gramene in green, Phytozome in rust, and maize-GAMER in tan. (a) Number of genes with at least one GO term annotated. (b) Number of GO terms with at least one gene annotated. (c) Percent coverage, number of annotations, and $hF_1$ score for each annotation dataset across the three GO graphs (i.e., Cellular Component, Molecular Function, and Biological Process).
Leaf terms are toward the bottom, root terms are toward the top. Terms covered only by the gold standard are shown in orange (labeled G), those in the dataset but absent from the gold standard are shown in blue (labeled D), and those that appear in both are shown in green (labeled DG). Leaf terms in each subgraph have an * next to them. Phytozome graph is shown at the top (5a) Gramene graph is shown in the middle (5b), and maize-GAMER aggregate graph is shown at the bottom (5c).
Figure 4.6: Distribution of evaluation metrics for Gold Standard Genes

Distribution of metrics $hRc$, $hPr$, and $hF_1$ for each annotation method based on genes in the gold standard dataset. If an annotation method did not have an annotation for a gene in the gold standard, it was assigned a value of 0. The bin width used to calculate the distribution was 0.05.
Figure 4.7: Evaluation metrics calculated for na1 gene for each dataset
Metrics hPr, hRc, and hF1 for existing (Gramene and Phytozome) datasets as well as the maize-GAMER aggregate and component datasets. hPr, hRc, and hF1 are shown in green, orange, and periwinkle, respectively.
References


CHAPTER 5. GENE ONTOLOGY META ANNOTATOR FOR PLANTS (GOMAP)

A manuscript to be revised and submitted as an Application Note to the journal *Bioinformatics*

Wimalanathan Kokulapalan, Carson M. Andorf, Iddo Friedberg, and Carolyn J. Lawrence-Dill

**Abstract**

**Summary**

Annotating gene structures and functions are crucial steps to making a newly assembled genome useful. The public gene ontology (GO) annotations generated and released for plants (and other species) are valuable for many research applications, including interpreting large-scale expression profiling studies, prioritizing candidate gene for functional analysis, etc. We have developed a high-throughput and reproducible pipeline for genome-scale GO annotation of plant genes called GOMAP for Gene Ontology Meta Annotator for Plants. The GOMAP pipeline is based on methods we used to improve functional annotations for the B73 maize reference genome. These methods were generalized for application across any sequenced plant genome and the pipeline was containerized to increase portability for the system and reproducibility for generating an annotation product. The GOMAP pipeline is optimized for HPC environments, and a beta-version has been released and demonstrated by annotating gene function for three additional maize lines. The pipeline is now being deployed for annotating maize, rice and cotton.

**Availability and implementation**

The beta-version of GOMAP is available as a Singularity version for reproducible annotation of plant genes, and as an open source development version for custom work. Instructions to obtain the Singularity container are accessible online at https://gomap-
singularity.readthedocs.io/en/latest/ and instructions for accessing the source code is available through https://github.com/Dill-PICL/GOMAP.

**Introduction**

The availability and accessibility of long-read sequencing technology has enabled the widespread adoption of this technology for plant genome assembly (Jiao et al., 2017; Rhoads & Au, 2015). The number of whole-genome sequencing (WGS) datasets generated by long-read sequencing has increased steadily over the past few years. Prior to 2015, only 19 plant long-read WGS datasets were published in NCBI Short Read Archive (SRA) database, whereas over 2,500 such datasets are currently available (Table 5.1). The number of species sequenced using long-read sequencing technology for the same period has increased from 14 to 124. This means that the number of high-quality de novo assembled plant genomes has also increased during the same time period. Just as better computational tools for genome assembly were required to reach this level of high-throughput, high-quality genome assembly, better computational tools for gene structure prediction and functional prediction are necessary to improve the usability of these high-quality assemblies.

Computational tools for functional annotation of genes leverage the knowledge derived from experimental characterization of gene functions and predict function for uncharacterized genes. From its inception in 1998, the Gene Ontology (GO) consortium has provided a common vocabulary that describes gene function (Ashburner et al., 2000). GO terms are organized as a directed, acyclic graph composed of nodes that are well-defined terms and edges that assert relationships between the terms (Ashburner et al., 2000). GO consists three different ontologies, namely biological process, cellular component, and molecular function. Each of these provide terms to describe different aspects of gene function. GO is widely used as a controlled functional annotation vocabulary and serves as an interpreter for many types of experimental data. Many
tools have been developed to assign GO terms to genes based to various types of gene features, and numerous statistical methods have been developed to perform GO term enrichment analysis to find shared functions among genes that exhibit similar expression patterns (Grossmann, Bauer, Robinson, & Vingron, 2007; Jiang et al., 2016; McLean et al., 2010; Radivojac et al., 2013; Young, Wakefield, Smyth, & Oshlack, 2010).

The maize-GAMER project was our effort to explore various methods and tools available for GO annotation and to evaluate the annotations and compare them to existing public annotation sets using a manually curated high-confidence dataset (Wimalanathan, Friedberg, Andorf, & Lawrence-Dill, 2018). The maize-GAMER project produced a high-coverage, reproducible GO annotation dataset for maize. Various GO annotation methods were evaluated for performance and accuracy during maize-GAMER project and only reproducible and state of the art methods were used for the annotation of GO terms. To enable better reproducibility and portability, we have generalized and streamlined the methods used in maize-GAMER and implemented a pipeline to annotate GO terms to other plant genomes. We call this pipeline Gene Ontology Meta Annotator for Plants (GOMAP).

Methods

GOMAP uses sequence-similarity, domain-presence and mixed-method pipelines to annotate GO terms to the plant protein sequences given by the user. The user is expected to annotate gene models from a whole genome assembly and filter the longest translated sequence for each gene model. GOMAP annotates the input sequences using the three types of methods mentioned above and produces a single unique and non-redundant GOMAP aggregate dataset (Figure 5.1). The datasets and tool version used in GOMAP have been listed in Table 5.2 and Table 5.3.
GO Annotation methods used in GOMAP

GOMAP uses sequence-similarity, domain-presence and mixed-method pipelines to annotate GO terms to the plant protein sequences given by the user. The user is expected to annotate gene models from a whole genome assembly and filter the longest translated sequence for each gene model.

The sequence similarity searches are performed against two plant datasets, *Arabidopsis* and UniProt. The Arabidopsis dataset consisted of translated transcript sequences obtained from TAIR (Table 5.2). A first set of annotations are generated by using BLAST search to obtain reciprocal-best-hits between input and Arabidopsis sequences, and inheriting curated GO terms from Arabidopsis to the input sequence (Figure 5.2). A second set of annotations are obtained by utilizing a similar approach, but instead of Arabidopsis the search is performed against the protein sequences for the top 10 annotated plants species in UniProt database. The species were ranked by the number of protein sequences with curated GO annotations in UniProt.

The InterProScan5 pipeline is used to detect the valid domains present on the input sequences, and assign GO terms to input sequences. InterProScan uses 14 types of protein signatures to detect putative domains in the input sequences, and assign GO terms (Jones et al., 2014). The InterProScan only reports the valid domains and GO annotations so the annotations were not filtered for this step.

Two mixed-method pipelines are used to annotate GO terms to the input sequences, namely Argot2.5 and PANNZER (Falda et al., 2012; Koskinen, Törönen, Nokso-Koivisto, & Holm, 2015). Each of these tools require preprocessed input sequences before they can be used to annotate GO terms (Figure 5.3). Argot2 requires the BLAST hits of the input sequences to the UniProt database and Pfam hits identified by HMMER search against Pfam domain database.
Design of the GOMAP-Singularity Container

The GOMAP pipeline was designed and developed with the purpose of building a containerized version. Singularity works well in high performance computing (HPC) environments and has been used to containerize complicated pipelines and tools for better portability and ease of use (Kurtzer, Sochat, & Bauer, 2017). Several challenges were encountered during the construction of the Singularity container. The first challenge was the size of the pipeline. The second challenge was the runtime the pipeline took to complete.

Minimizing GOMAP-singularity size

GOMAP local installation uses around 110 GB of disk space for local installation. Most of the size is due to the inclusion of tools and associated data for the tools (Table 5.2 and Table 5.3). The singularity container size increased to 60-70 GB when all the tools and data were included in the container. This made developing, optimizing and testing the pipeline a challenge. The tools and associated data necessary for GOMAP was separated from the container and stored in CyVerse data commons (Merchant et al., 2016). GOMAP pipeline has a specific step to download and setup the tools and data necessary to a location specified by the user within the container.

Utilizing MPI to parallelize tasks

The runtime for the GOMAP pipeline on a single machine or a single node in HPC for ~40000 protein sequences ranges from 10-14 days. The time depends on several parameters, such as input sequence lengths and node configuration. The step that takes most time is the BLAST search against UniProt sequence database, which takes 8-10 days. The time limit for a single job on the HPC environments vary between 2-5 days, and most of them have a limit of 2
days. After the time limit jobs are killed. UniProt BLAST step of the GOMAP pipeline is designed to recover from crashes, so it can be restarted until completion. This would still take the 2 weeks for GOMAP to complete. The singularity container is compatible with several message passing interface (MPI) libraries, which are used to parallelize the tasks and run the job on multiple nodes in HPC. GOMAP pipeline was optimized using MPI libraries to enable the use of multiple nodes on clusters. Using MPI and 10 nodes the runtime of the UniProt BLAST step was reduced to 14-15 hours.

**Basic steps to run the GOMAP-singularity Pipeline**

GOMAP-singularity has 7 discrete steps, and these steps are as listed in Table 5.4. After the initial setup, GOMAP can run the first three steps concurrently, to complete sequence-similarity, domain-presence, and UniProt BLAST steps. This reduces the overall time for GOMAP to complete. Next steps mixed-method preprocessing, mixed-method and aggregation steps require output from previous steps and cannot be run concurrently. Users can choose to use bash scripts that are distributed with the container to run the container if they do not want to configure the GOMAP installation themselves. The detailed instructions for running the container are available at https://gomap-singularity.readthedocs.io/en/latest/.

**Results**

The GOMAP-Singularity container was tested by annotating GO terms to three maize genomes from different inbred lines. The Pittsburgh Supercomputing (PSC) - Bridges HPC cluster was used to run GOMAP. Each compute node on PSC Bridges is equipped with 28-core processors and 128GB RAM. The analysis was performed on single nodes for the un-parallelized steps and was run on 10 nodes for the parallelized steps. The time taken for each step is given in Table 5.5. The number of protein coding genes is as follows Mo17 had 38,620, W22 had 40,690, and PH207 had 40,557 gene sequences. Each dataset was slightly different from each
other in terms of gene length and number of sequences. The differences among the time taken to run each step are mostly due to domain step and the difference is significant between Mo17 and W22 datasets (Figure 5.4). The differences in the time taken cannot be accounted for merely by the number of input sequences. For example, although, PH107 and W22 have the same number of input sequences, the running time is longer for W22. The maize annotation datasets are currently being checked for quality prior to public release.

Discussion

We have developed the GOMAP pipeline for GO annotation of gene models annotated to new plant genome assemblies. The methods that have been produced used in the pipeline were selected from the methods that were used for maize-GAMER project and have been evaluated using curated dataset. The containerization of the pipeline along with the parallelization of the most time-consuming UniProt BLAST step, has enabled the pipeline to annotate new plant genomes under two days using 10 nodes on the PSC Bridges cluster. A conservative estimate puts the UniProt BLAST step at approximately 7 days on PSC Bridges for approximately 40000 sequences, if the step was run on a single node compared to the 14 hours for 10 nodes. Splitting individual GOMAP steps has enabled concurrent execution that will make the real time completion shorter than the totals shown in Table 5.5. Depending on the time it takes on the Argot2 web server, that whole annotation process should be completed under 24 hours for ~40,000 input sequences. The domain and mixed method steps could be parallelized that would further decrease the runtime.

Each of GO annotation methods used in GOMAP have advantages and disadvantages. Sequence similarity methods provide an easy and direct inheritance of GO terms from genes in other species. For an experimentalist, the direct link is invaluable and enables critical evaluation of the GO terms annotated and enables them to generate testable hypotheses. However,
sequence-similarity methods annotate lower number of GO terms. Domain-presence methods provide a balanced approach that produce more GO annotations than sequence-similarity and are still based on sequence domains that can be used to find genes in other species. This enables critical evaluation and testable hypothesis for biological experiments. GO annotations produced by domain-presence methods, due to the higher number, are also more suitable for the interpretation of high-throughput experiments. Both sequence-similarity and domain-presence methods annotate only less than half the genes is annotated. This presents a challenge when interpreting data from high-throughput experiments. State of the art mixed-methods, as determined by the CAFA competition, have better or comparable prediction accuracy to other methods. More interestingly they provide more annotations and annotate more genes than both other methods. The higher number of annotations makes them more suitable for interpreting datasets from large-scale experiments. The utilization of advanced statistical and computational methods makes it difficult to make direct connections to genes from other species.

The tools used in GOMAP have been implemented based on the evaluation with an unbiased dataset obtained from MaizeGDB. The mixed-method tools used in GOMAP were the top performing methods in the first iteration of CAFA (Radivojac et al., 2013). Currently the second and third iteration of the CAFA competition have been completed. More cutting-edge tools have been developed in each iteration (Jiang et al., 2016). With the use of gold-standard dataset more tools could be evaluated and integrated into GOMAP in the future. Several other pipelines exist for GO annotations, but only a few are plant-specific (Amar et al., 2014; Zwaenepoel et al., 2018). Other plant-specific GO annotation pipelines available do not focus on predicting genome-scale GO terms, and instead mainly focus on subsets of gene ontology (Zwaenepoel et al., 2018). GOMAP provides annotations for all genes and all functions.
In summary, the reproducible methods used and evaluated by the maize-GAMER project have been integrated as the GOMAP pipeline that can annotate GO terms to plant proteins. GOMAP has been successfully used to annotate GO terms to gene models from other sequenced maize inbred lines and is currently being applied to rice, cotton, and wheat.

Acknowledgements

We thank Ramona Walls and Darwin Campbell for the help they provided to host the GOMAP data on CyVerse data commons. We thank Nathan Weeks and Steven Canon from USDA-ARS, Ames Iowa for their help with alpha-testing the GOMAP-singularity container and valuable suggestions. We thank the BCB graduate students Ian Braun, Gaurav Kandoi, and Ashish Jain for alpha-testing of GOMAP-Singularity. We thank the rotation students Ha Vu, Dennis Psaroudakis, and Parnal Joshi for beta-testing GOMAP for rice, wheat, and cotton, respectively. Thanks to the members of the Dill Plant Informatics and Computational Lab for critical review and helpful suggestions (dill-picl.org).

Access to PSC Bridges cluster was provided by the XSEDE startup allocation awarded to K.W. and C.J.L.D. This work was supported by funding from the Iowa State University Plant Sciences Institute Faculty Scholars Program to C.J.L.D.; the National Science Foundation [IOS #1027527] to C.J.L.D.; and the United States Department of Agriculture - Agricultural Research Services to C.M.A. IF was supported, in part, by National Science Foundation award ABI-1458359.
Tables

Table 5.1: The number of long-read sequencing datasets published in the NCBI SRA database

<table>
<thead>
<tr>
<th>Year</th>
<th>Datasets</th>
<th>Plant Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>before 2016</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>2016</td>
<td>641</td>
<td>28</td>
</tr>
<tr>
<td>2017</td>
<td>1,000</td>
<td>37</td>
</tr>
<tr>
<td>2018</td>
<td>1,310</td>
<td>65</td>
</tr>
<tr>
<td>Total</td>
<td>2,962</td>
<td>124</td>
</tr>
</tbody>
</table>

Number of long-read sequencing datasets in the NCBI Short Read Archive (SRA) were determined by querying with the following parameters. Platform = “pacbio smrt”[platform] or “oxford nanopore”[platform], organism = (“Embryophyta”[Organism]), molecule type = “biomol dna”[Properties], dataset type = “strategy wgs”

Table 5.2: Public datasets used in GOMAP

<table>
<thead>
<tr>
<th>Database</th>
<th>Type</th>
<th>Format</th>
<th>Version</th>
<th>Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAIR</td>
<td>Protein Sequences</td>
<td>fasta</td>
<td>TAIR 10</td>
<td>Arabidopsis thaliana</td>
</tr>
<tr>
<td>TAIR</td>
<td>GO Annotations</td>
<td>gaf 2.0</td>
<td>TAIR 10 (20170410)</td>
<td>Arabidopsis thaliana</td>
</tr>
<tr>
<td>Gramene 49</td>
<td>Gene Annotations</td>
<td>gff3</td>
<td>5b+</td>
<td>Zea mays</td>
</tr>
<tr>
<td>Gramene 49</td>
<td>GO Annotations</td>
<td>gaf 2.0</td>
<td>5b+</td>
<td>Zea mays</td>
</tr>
<tr>
<td>Phytozome 11</td>
<td>GO Annotations</td>
<td>tsv</td>
<td>5b+</td>
<td>Zea mays</td>
</tr>
<tr>
<td>Uniprot</td>
<td>Protein sequences</td>
<td>fasta</td>
<td>20170410</td>
<td>All species</td>
</tr>
<tr>
<td>Uniprot</td>
<td>Protein sequences</td>
<td>fasta</td>
<td>20170410</td>
<td>All plants</td>
</tr>
<tr>
<td>Uniprot</td>
<td>GO Annotations</td>
<td>gaf 2.0</td>
<td>20170410</td>
<td>All plants</td>
</tr>
<tr>
<td>Pfam</td>
<td>HMMs</td>
<td>hmm</td>
<td>27.0</td>
<td>All species</td>
</tr>
<tr>
<td>PANTHER</td>
<td>HMMs</td>
<td>hmm</td>
<td>10.0</td>
<td>All species</td>
</tr>
</tbody>
</table>
Table 5.3: Software tools used in GOMAP

<table>
<thead>
<tr>
<th>Software</th>
<th>Type</th>
<th>Version</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCBI-BLAST</td>
<td>Sequence similarity</td>
<td>2.6.0</td>
<td>(Altschul, Gish, Miller, Myers, &amp; Lipman, 1990)</td>
</tr>
<tr>
<td>HMMER</td>
<td>HMM scanning</td>
<td>3.1b1</td>
<td>(Finn et al., 2011)</td>
</tr>
<tr>
<td>InterProScan5</td>
<td>GO Annotation</td>
<td>5.15-55.0</td>
<td>(Jones et al., 2014)</td>
</tr>
<tr>
<td>PANNZER</td>
<td>GO Annotation</td>
<td>1.1</td>
<td>(Koskinen et al., 2015)</td>
</tr>
<tr>
<td>Argot2</td>
<td>GO Annotation</td>
<td>2.5 (Server)</td>
<td>(Falda et al., 2012)</td>
</tr>
<tr>
<td>FANN-GO</td>
<td>GO Annotation</td>
<td>1 version</td>
<td>(Clark &amp; Radivojac, 2011)</td>
</tr>
<tr>
<td>AIGO</td>
<td>GO Evaluations</td>
<td>0.1.0</td>
<td>(Defoin-Platel et al., 2011)</td>
</tr>
</tbody>
</table>

Table 5.4: Steps to run the GOMAP-singularity container

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Depends on previous step</th>
</tr>
</thead>
<tbody>
<tr>
<td>setup</td>
<td>Download and extract data from CyVerse Data Commons</td>
<td>NA</td>
</tr>
<tr>
<td>seqsim</td>
<td>Runs sequence similarity steps</td>
<td>No</td>
</tr>
<tr>
<td>domain†</td>
<td>Run InterProScan5</td>
<td>No</td>
</tr>
<tr>
<td>mixmeth-blast‡</td>
<td>Run the BLAST step against UniProt</td>
<td>No</td>
</tr>
<tr>
<td>mixmeth-preproc</td>
<td>Convert output from UniProt BLAST</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Run HMMER</td>
<td>Yes</td>
</tr>
<tr>
<td>mixmeth†</td>
<td>Submit jobs to Argot2.5 webserver</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Run PANNZER</td>
<td>Yes</td>
</tr>
<tr>
<td>aggregate</td>
<td>Get all GO annotations, clean and generate aggregate dataset</td>
<td>Yes</td>
</tr>
</tbody>
</table>

‡This step has been parallelized for HPC, †This step can be parallelized but has not been done yet
Table 5.5: The number of nodes used for each step and the run time for each GOMAP step

<table>
<thead>
<tr>
<th></th>
<th>Nodes</th>
<th>Mo17</th>
<th>W22</th>
<th>PH207</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Genes</td>
<td></td>
<td>38,620</td>
<td>40,690</td>
<td>40,557</td>
</tr>
<tr>
<td>Step</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>seqsim</td>
<td>1</td>
<td>2h45m</td>
<td>3h41m</td>
<td>2h40m</td>
</tr>
<tr>
<td>domain</td>
<td>1</td>
<td>15h46</td>
<td>20h23m</td>
<td>18h4m</td>
</tr>
<tr>
<td>mixmeth-blast</td>
<td>10</td>
<td>14h54m</td>
<td>20h23m</td>
<td>14h54m</td>
</tr>
<tr>
<td>mixmeth-preproc</td>
<td>1</td>
<td>4h26m</td>
<td>4h48m</td>
<td>3h16m</td>
</tr>
<tr>
<td>mixmeth</td>
<td>1</td>
<td>2h25m</td>
<td>2h11m</td>
<td>2h20m</td>
</tr>
<tr>
<td>aggregate</td>
<td>1</td>
<td>0h10m</td>
<td>0h10m</td>
<td>0h10m</td>
</tr>
<tr>
<td>Total</td>
<td>1</td>
<td>2h45m</td>
<td>3h41m</td>
<td>2h40m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40h26m</td>
<td>47h4m</td>
<td>41h19m</td>
</tr>
</tbody>
</table>

† The domain step for PH207 was calculated by averaging the times for the domain step from the other two inbreds.
Figure 5.1: Overview of the GOMAP pipeline

The types of methods and major steps used for the GOMAP pipeline are shown here. Sequence-similarity methods are colored in yellow, domain-presence methods are blue, and mixed-methods are green. Outputs from these methods are combined, duplicates and redundancies are removed, and an aggregate dataset is the final dataset produced.
Figure 5.2: GO annotations using sequence-similarity to Arabidopsis

Detailed steps the approach to annotate GO terms using sequence similarity to Arabidopsis dataset. The same steps are performed against the top 10 annotated species in UniProt as well.
Figure 5.3: Mixed-method based GO annotation of the input sequences

This figure illustrates the detailed steps of how the input sequences are annotated using the mixed-methods Argot2 and PANNZER. The input sequences are preprocessed initially, by a BLAST step to UniProt and HMMER step to Pfam database. The output from the preprocessing is used by Argot2 and PANNZER for GO annotation.
Figure 5.4: Comparison of GOMAP runtimes for annotating maize genomes

This figure provides a comparison of runtimes for each step for the three maize genomes annotated using GOMAP, namely Mo17, W22, and PH207. Each step of the GOMAP pipeline is colored as per the legend, and the steps are given in the order of execution. The domain step for PH207 was calculated by averaging the times for the domain step from the other two inbreds.
References


CHAPTER 6. GENERAL CONCLUSIONS

Summary

The research presented in this thesis has focused on methods to assign function to maize genes. We have improved methods in both experimental and computational approaches to assign gene function.

We have used high-throughput sequencing technology to characterize maize mutants. We applied short-read sequencing technology to positionally clone maize genes. We developed the BSA-GBS method for rough-mapping to identify causal loci in maize mutants and utilized WGS methods to fine-map and clone genes in maize mutants. We have also used long-read sequencing technology to place transgene T-DNAs on the maize genome. A molecular protocol to enrich genomic T-DNA flanks to multiplex and sequence multiple T-DNA insertions in the same experiment was designed as part of the project.

Computational methods are indispensable for high-throughput gene function annotation and utilize the existing data from experimental and curatorial approaches in other species for this purpose. We have annotated gene function using GO terms to the gene models in the maize reference genome using reproducible methods in the maize-GAMER project. The dataset increased the number of genes annotated and the number of annotations, with comparable quality to existing datasets. The critical component of this endeavor was the evaluation of the GO annotation methods using manually curated high-confidence GO annotations. The methods evaluated and used in the maize-GAMER projects were assembled into a streamlined pipeline called GOMAP. The GOMAP pipeline was containerized using singularity to enable portability and reproducibility. We have parallelized GOMAP and significantly reduced the runtime on HPC.
Contributions to collaborative projects of significance

There were several collaborative projects that I participated in during my time as a graduate student. While these projects are not within the scope of the thesis, they are significant in terms of my training and research outcomes. They are listed here with modified abstracts from published work or short descriptions and include my personal contribution to each project. The projects are listed in chronological order of participation.

MaizeGDB expression analysis tool based on MapMan

MaizeGDB is a highly curated, community-oriented database and informatics service to researchers focused on the crop plant and model organism *Zea mays* ssp. *mays*. MaizeGDB hosts several tools for examining maize data, such as BLAST, genome browser, and expression analysis tools. MapMan was developed at the Max Planck Institute for Molecular Plant Physiology in Potsdam, Germany (Usadel et al., 2009). The MapMan software suite allows the visualization of a variety of functional genomics datasets (gene expression, protein, enzyme, and metabolite levels) in the context of a large number of well characterized biochemical processes and metabolic pathways. The microarray expression dataset from 60 maize tissues from Sekhon et al. was processed and incorporated into the MapMan Web interface at MaizeGDB (Sekhon et al., 2011). The interface allowed users to visualize expression values for single tissue or compare expression between two tissues.

My contribution to this project was the processing of the datasets, and integration of the MapMan web module into MaizeGDB for visualization. This involved a collaboration between MaizeGDB and technical staff from MapMan. This work has been part of MaizeGDB publication in Nucleic Acids Research (Andorf et al., 2016).
Gene network variation and alternative paths to convergent evolution in turtles

Diversification of the turtle’s shell comprises remarkable phenotypic transformations. For instance, two divergent species convergently evolved shell-closing systems with shoulder blade (scapula) segments that enable coordinated movements with the shell. We expected these unusual structures to originate via similar changes in underlying gene networks, as skeletal segment formation is an evolutionarily conserved developmental process. We tested this hypothesis by comparing transcriptomes of scapula tissue across three stages of embryonic development in three emydid turtles from natural populations. We found that alternative strategies for skeletal segmentation were associated with interspecific differences in gene co-expression networks.

The participation in this project was facilitated by BCB Lab, which is volunteered consultation provided to researchers who request bioinformatics help. Me and a fellow BCB student Haibo Liu worked on the initial analysis of the RNA-seq produced for this project. This included RNA-seq QC, genome alignment, Haibo Liu performed differential expression analysis, and I constructed de novo transcriptome assemblies for the detection of novel species-specific transcript isoforms. This work has been published in Evolution and Development (Cordero et al., 2018).

The maize W22 genome provides a foundation for functional genomics and transposon biology

The maize W22 inbred has served as a platform for maize genetics since the mid twentieth century. To streamline maize genome analyses, we have sequenced, and de novo assembled a W22 reference genome using short-read sequencing technologies. We show that significant structural heterogeneity exists in comparison to the B73 reference genome at multiple scales, from transposon composition and copy number variation to single-nucleotide...
polymorphisms. The generation of this reference genome enables accurate placement of thousands of Mutator (Mu) and Dissociation (Ds) transposable element insertions for reverse and forward genetics studies.

My contribution to this project was the placement of Ds elements to the genome and the comparison of Ds and Mu insertion locations between the W22 and B73 genomes. This involved obtaining Ds genomic flanks from NCBI and adapting and running an alignment pipeline developed for the Ac/Ds mutagenesis project on the W22 and B73 genomes and post-processing of the alignments. This work has been published in Nature Genetics (Springer et al., 2018).

**Primer Server - A web application to design primers for the amplification of unique DNA targets in complex genomes**

Polymerase Chain Reaction (PCR) is a technique to amplify a specific DNA region. PCR primers are short, single-stranded DNAs that define the section of DNA to be amplified. Two primers are used in each PCR reaction, designed so that they flank the target region. Critically, off-target binding may lead to experimental failure or worse, to misleading results. Thus, potential primers that amplify genomic DNA must be examined for off-target binding across the genome. The purpose was to make a user-friendly tool that can design PCR primers efficiently and accurately as well as visualize the designed primers. Our web-based bioinformatics tool selects optimal primer sequences within the starting material by using a C module called primer3plus and then prioritizing and/or eliminating potential primers based on BLAST. This tool has an easy-to-use interface which was designed using Angular2, and an efficient server-side code written in Python. While similar tools exist, our tool is more user-friendly, efficient and uses extensive form validation to minimize errors in the user input.

This was a collaborative project with an undergraduate student Takao Shibamoto in the Vollbrecht lab. My contribution to the project was software design, and the development of the
primer filtering and analysis tool (primerDAFT), a python package to design and filter primers. This has been presented by Takao at the National Conference on Undergraduate Research (NCUR) 2018 (Iowa State University, 2018).

**Large-scale transcriptomics study on the effect of drought on early maize inflorescence development**

When water availability is limited during the early growing season, early season drought stress disturbs or blocks maize ear development, which negatively impacts yield. A large-scale RNA expression profiling study was conducted to understand the mechanisms by which early season drought stress impacts the developmental processes that define the architecture of the maize ear. A multi-institute experiment was conducted at the automated Pioneer (now known as Corteva) greenhouse in Johnston. A set of multi-stage samples of ear and tassel were collected from drought stressed and well-watered plants. Samples were collected at multiple time points spanning over two weeks by dissecting over 1000 plants. The samples were used to generate a RNA-seq data for 24-samples (72 reps) and is currently being analyzed.

As the member of the local team, my contribution to this project was to lead the effort to carry out the experiment at the Pioneer greenhouse. This included being the contact person for Pioneer personnel who working in the greenhouse, estimating seed planting and dissection dates to plan travel for out of state team members, and staging the tissue to confirm collection dates. I also led the tissue collection efforts to dissect, fix, and transport the tissues back to Ames from Pioneer’s facilities in Johnston. I was also involved in extracting RNA from the collected samples and oversaw the construction of libraries and sequencing to generate the RNA-seq data.
References


APPENDIX A. UREA GENOMIC DNA EXTRACTION PROTOCOL

Protocol to extract genomic DNA from maize tissue adapted from the “Urea-based Plant DNA Miniprep” by [cite Chen and Dellaporta].

**Urea Extraction Buffer:**

- 420 g Urea
- 70 mL 5M NaCl
- 50 mL 1M Tris-HCl pH 8
- 40 mL 0.5M EDTA
- 10 g n-lauroyl sarcosine
- to 1.0 L with d.i. water
- filter through 3mm filterpaper w. funnel

**4.4M NH₄OAc, pH 5.2:**

- 105 mL d.i. water
- 50.5 mL glacial acetic acid
- 45 mL NH₄OH (add slowly in fume hood)

**High-salt TE::**

- 100mM Tris-HCl (pH 7.5); 10 mM EDTA (pH 8.0); 0.7M NaCl
- 10 mL 1 M Tris-HCl pH 7.5
- 2.0 mL 0.5 M EDTA
- 14 mL 5M NaCl
- 74 mL d.i. water

**Protocol**

1. Leaf tissue punches were ground in liquid nitrogen using a mortar and pestle.

2. Ground, frozen leaf tissue (3-5 mL volume in 15 mL screw-cap falcon tube) was mixed with an “equal volume” (3-5 mL) of extraction buffer

3. Tissue and buffer were mixed well by vortexing and shaking.

4. Equal volume (3-5 mL) phenol:chloroform (2:1, where chloroform is 24:1 chloroform:isoamyl alcohol) was added.

5. The tube was mixed well and placed on platform shaker 15 minutes @80 rpm.

6. Spin at 3000 rpm for 20 minutes.
7. Supernatant was poured off into a clean 14 mL falcon culture tube.

8. 0.3 mL 4.4M NH4OAc (or 1/10th volume of the aqueous phase) was added and mixed

9. An equal volume of room-temperature isopropanol was added.

10. The snard of DNA was hooked using a bent glass pasture pipette.

11. DNA was patted dry on a kimwipe and resuspend in 2 mL high salt TE (100mM Tris-HCl (pH 7.5); 10 mM EDTA (pH 8.0); 0.7M NaCl).

12. 5uL RNAse A (10 mg/mL, DNase free) was added and the mixture was incubated for 1 hour or longer at 37C.

13. Tubes were taken out and allowed to reach room temperature for 10 min.

14. 5 mL Ethanol (100%) was added.

15. DNA snard was hooked and placed into a 1.7 mL tube containing 1.0 mL of wash solution I (76%EtOH containing 0.2M NaOAc) for 5-10 minutes.

16. DNA was transferred into another 1.7 mL tube containing 1.0 mL wash buffer II (76%EtOH containing 10mM NH4OAc) for 2-5 minutes.

17. DNA was dried carefully on a Kimwipe and placed into a clean 1.7 mL tube containing 200uL of TE.

18. DNA was heated at 65C for 10 minutes and let sit overnight at 4C.
APPENDIX B.  BSA-GBS PROTOCOL

This is the detailed description of the bulked-segregant-analysis by genotyping-by-sequencing protocol, adapted from the original genotyping-by-sequencing method developed by Rob Elshire in Ed Buckler's lab.

**Barcodes selected for BSA-GBS**

<table>
<thead>
<tr>
<th>ID</th>
<th>Barcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AACT</td>
</tr>
<tr>
<td>2</td>
<td>GATC</td>
</tr>
<tr>
<td>3</td>
<td>TTCTC</td>
</tr>
<tr>
<td>4</td>
<td>TCGTT</td>
</tr>
<tr>
<td>5</td>
<td>ATGCCT</td>
</tr>
<tr>
<td>6</td>
<td>ATATGT</td>
</tr>
<tr>
<td>7</td>
<td>CATCGT</td>
</tr>
<tr>
<td>8</td>
<td>ATTGGAT</td>
</tr>
<tr>
<td>9</td>
<td>GAACTTC</td>
</tr>
<tr>
<td>10</td>
<td>TGGTACGT</td>
</tr>
<tr>
<td>11</td>
<td>TCTCAGTC</td>
</tr>
<tr>
<td>12</td>
<td>CCGGATAT</td>
</tr>
</tbody>
</table>

**Adapter and PCR primer Sequences for BSA-GBS**

The adapter sequences and PCR primers sequences were optimized for Illumina Sequencing, and were taken from (Elshire et al 2011) GBS protocol, but a smaller selection of 16 adapters were selected. Out of that 6 were used for the BSA-GBS trial. The adapters were synthesized at the Iowa State University DNA Facility and were cartridge purified.
<table>
<thead>
<tr>
<th>Name</th>
<th>Sequence</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>adapter_1_F</td>
<td>CWGAACTAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>40</td>
</tr>
<tr>
<td>adapter_1_R</td>
<td>ACACTCTTTCCTCACACGACGCTTTCCCGATCTAGTT</td>
<td>37</td>
</tr>
<tr>
<td>adapter_2_F</td>
<td>CWGGATCGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>40</td>
</tr>
<tr>
<td>adapter_2_R</td>
<td>ACACTCTTTCCTACACGACGCTTCTCCCGATCTAGTC</td>
<td>37</td>
</tr>
<tr>
<td>adapter_3_F</td>
<td>CWGTTCCTAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>41</td>
</tr>
<tr>
<td>adapter_3_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTGAGAA</td>
<td>38</td>
</tr>
<tr>
<td>adapter_4_F</td>
<td>CWGTCTGTAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>41</td>
</tr>
<tr>
<td>adapter_4_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTAAAGCA</td>
<td>38</td>
</tr>
<tr>
<td>adapter_5_F</td>
<td>CWGATGCTCTAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>42</td>
</tr>
<tr>
<td>adapter_5_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTAGCAT</td>
<td>39</td>
</tr>
<tr>
<td>adapter_6_F</td>
<td>CWGATATGTAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>42</td>
</tr>
<tr>
<td>adapter_6_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTACATAT</td>
<td>39</td>
</tr>
<tr>
<td>adapter_7_F</td>
<td>CWGCATCGTAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>42</td>
</tr>
<tr>
<td>adapter_7_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTAGCAT</td>
<td>39</td>
</tr>
<tr>
<td>adapter_8_F</td>
<td>CWGATTGGGATAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>43</td>
</tr>
<tr>
<td>adapter_8_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTATCAAAT</td>
<td>40</td>
</tr>
<tr>
<td>adapter_9_F</td>
<td>CWGGAACCTTCAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>43</td>
</tr>
<tr>
<td>adapter_9_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTAAGTC</td>
<td>40</td>
</tr>
<tr>
<td>adapter_10_F</td>
<td>CWGTGTTACGTAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>44</td>
</tr>
<tr>
<td>adapter_10_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTAGTACCA</td>
<td>41</td>
</tr>
<tr>
<td>adapter_11_F</td>
<td>CWGCTCCTAGTCAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>44</td>
</tr>
<tr>
<td>adapter_11_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTAAGTAGA</td>
<td>41</td>
</tr>
<tr>
<td>adapter_12_F</td>
<td>CWGCCTGATATAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>44</td>
</tr>
<tr>
<td>adapter_12_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTAAGTAGA</td>
<td>41</td>
</tr>
<tr>
<td>adapter_13_F</td>
<td>CWGCATCGTTAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>42</td>
</tr>
<tr>
<td>adapter_13_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTAGTATC</td>
<td>39</td>
</tr>
<tr>
<td>adapter_14_F</td>
<td>CWGTACAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>40</td>
</tr>
<tr>
<td>adapter_14_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTAAGTAGA</td>
<td>37</td>
</tr>
<tr>
<td>adapter_15_F</td>
<td>CWGAATATGCGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>43</td>
</tr>
<tr>
<td>adapter_15_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTAGATATT</td>
<td>40</td>
</tr>
<tr>
<td>adapter_16_F</td>
<td>CWGACGACTACAGATCGGAAGAGCGTCGTTAGGGAAAGAGTGT</td>
<td>44</td>
</tr>
<tr>
<td>adapter_16_R</td>
<td>ACACCTTTTCCCTACACGACGCTTCTCCCGATCTAAGTAGT</td>
<td>41</td>
</tr>
<tr>
<td>adapter_common_F</td>
<td>CWGAGATCGGAAGACGGGTTCAGAGCAGAAATGCCAGG</td>
<td>36</td>
</tr>
<tr>
<td>adapter_common_R</td>
<td>CTGGCCATTCCTCTGCTGAAACCCGTCTCCCGATCT</td>
<td>33</td>
</tr>
<tr>
<td>PCR_F</td>
<td>AATGATACGGGCCGACACCCAGAATTCACACTCTTTCCTACACGACGCTTCTCCCGATCT</td>
<td>58</td>
</tr>
<tr>
<td>PCR_R</td>
<td>CAAGCAGAAGACGGCATACGAGATCGGCTTCCGCATTCTGCTAGCTGAACCGCTCTCCCGATCT</td>
<td>61</td>
</tr>
</tbody>
</table>
Anneal adapters

Prepare adapters

- Suspend the adapters received in TE to make a 2uM solution
- In a PCR plate/8-tube strip bring together the barcoded adapters
- In a 0.2 ml tube bring together the common adapter

Total volume 100ul Annealed Molarity: 50mM

<table>
<thead>
<tr>
<th>Component</th>
<th>Conc</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top Strand</td>
<td>200 uM</td>
<td>25uL</td>
</tr>
<tr>
<td>Bottom Strand</td>
<td>200 uM</td>
<td>25uL</td>
</tr>
<tr>
<td>TE</td>
<td></td>
<td>50uL</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>50 mM</td>
<td>100uL</td>
</tr>
</tbody>
</table>

**Thermocycling parameters**

- 95 degrees for 2 minutes
- Ramp to 25 degrees by 0.1 degree per second
- Hold at 25 degrees for 30 minutes
- Hold at 4 degrees forever

Quantify Adapter Concentration with Qubit or Quantifluor

Adapter stock solution

Make a stock solution of the barcoded (BC) and Common adapter mix. The stock adapter solution is 3ng/uL of adapters.

<table>
<thead>
<tr>
<th>Material</th>
<th>Volume/Mass</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC Adapter</td>
<td>300 ng</td>
</tr>
<tr>
<td>Common Adapter</td>
<td>300 ng</td>
</tr>
<tr>
<td>TE</td>
<td>200uL</td>
</tr>
</tbody>
</table>

Vortex and Spin

Restriction Digestion

We have to set up the restriction digestion of the genomic DNA with ApeKI.

Digestion Master Mix

<table>
<thead>
<tr>
<th>Material</th>
<th>1x</th>
<th>12x</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEB Buffer 3</td>
<td>2uL</td>
<td>24uL</td>
</tr>
<tr>
<td>ApeKI</td>
<td>1uL</td>
<td>12uL</td>
</tr>
<tr>
<td>Water</td>
<td>17uL</td>
<td>204uL</td>
</tr>
<tr>
<td>Water</td>
<td>20uL</td>
<td>240uL</td>
</tr>
</tbody>
</table>

- Add 100 ng of high molecular weight genomic DNA (10uL of 10ng/uL recommended or 100 ng + water to 10uL)
- Add 20uL of digestion master mix
• Incubate at 75°C for 120 minutes
• Hold at 4°C

**Ligation**

The digested genomic fragments can be ligated to the adapters prepared earlier.

**Ligation master mix**

<table>
<thead>
<tr>
<th>Material</th>
<th>1x</th>
<th>12x</th>
</tr>
</thead>
<tbody>
<tr>
<td>10x T4 DNA Ligase Buffer</td>
<td>5uL</td>
<td>60uL</td>
</tr>
<tr>
<td>T4 DNA Ligase</td>
<td>1.6uL</td>
<td>19.2uL</td>
</tr>
<tr>
<td>Water</td>
<td>7.4uL</td>
<td>88.8uL</td>
</tr>
<tr>
<td>Total</td>
<td>14uL</td>
<td>168uL</td>
</tr>
</tbody>
</table>

- Add 6uL of 0.6 ng/uL adapter solution, and this should be unique for each DNA sample.
- Add 14uL of ligation master mix
- Ligate at 22°C for 60 minutes
- Heat for 65°C for 30 minutes
- Hold at 4°C

**Ligation Cleanup**

The Adapter ligated DNA fragments should be cleaned before the next step, and we used Aline PCRCleanDX beads to clean the Ligated fragments

- Add 1.8x (90uL) Aline PCRCleanDX beads to each tube
- Pipette 10 times or vortex the mixture
- Incubate at room temp for 5 min
- Place the reactions on a magnetic stand for 5-10 min
- Remove the solution
- Wash with 200uL of 70% ethanol while leaving the samples on the stand
- Repeat ethanol wash
- Leave to dry on the magnetic stand for 10 min
- Elute in 20uL of TE (Take extra care with smaller elution volume)

**PCR for Illumina Library Construction**

The PCR is to be performed on individual DNA pools instead of combining all the samples as with the original GBS protocol.
PCR Master Mix

<table>
<thead>
<tr>
<th>Material</th>
<th>1x</th>
<th>12x</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taq Master-mix</td>
<td>25</td>
<td>300</td>
</tr>
<tr>
<td>PCR Primer F (12.5 uM)</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>PCR Primer R (12.5 uM)</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>Water</td>
<td>21</td>
<td>252</td>
</tr>
<tr>
<td>Total</td>
<td>47</td>
<td>564</td>
</tr>
</tbody>
</table>

PCR Protocol

- Add 2uL of clean ligated DNA fragments
- Add 47uL of PCR master-mix
- Run PCR Protocol

PCR Cycling Parameters

1. 5 minutes at 72deg C
2. 30 seconds at 94deg C
3. 18 cycles of:
   1. 10 seconds at 94deg C
   2. 30 seconds at 65deg C
   3. 30 seconds at 72deg C
4. 5 minutes at 72deg C
5. Hold at 4deg C

Library Clean-up

The library should be cleaned up using the same process used for cleaning up the ligated fragments. Please refer to Aline bead-based clean-up instructions.

The library should be size selected at this point to make sure that the adapter dimers are omitted before the next step, adapter dimers will amplify efficiently in the Illumina Sequencing and waste large proportion of reads.

Check Quality and Quantity

- The library quality can be checked by BioAnalyzer or Fragment Analyzer
- The library can be quantified by Qubit or Quantifluor
  - Calculate the concentration of each barcoded

Normalize and Combine Barcode Libraries for multiplex sequencing

Follow best practices [here](#)

- Calculate the average DNA size of each sample from the BioAnalyzer 2100 high-sensitivity DNA chip
- Calculate ng/uL DNA concentration by Qubit
- Calculate the nM concentration of each library using the average DNA size and ng/uL concentration
- Dilute each sample to 2nM concentration
• Pool equal volume of each sample to prepare a multiplexed-library
• Supply the necessary volume to the sequencing facility
APPENDIX C. T-DNA FLANK AMPLICON LIBRARY CONSTRUCTION

This is the detailed description of the protocol used to construct the PacBio libraries to identify the T-DNA flanking sequences in transgenic maize.

**Pool Setup**

The experiment was used to map insertions from multiple transgenic events in a single sequencing run. The protocol was used to map insertions for 20 transgenic events in a single sequencing run. The table below shows a pooling setup.

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>668.11.1</td>
<td>668.2.1</td>
<td>668.4.1</td>
<td>668.8.5</td>
</tr>
<tr>
<td>R2</td>
<td>668.13.4</td>
<td>668.15.3</td>
<td>668.5.2</td>
<td>668.9.5</td>
</tr>
<tr>
<td>R3</td>
<td>668.14.2</td>
<td>668.5.4</td>
<td>668.2.4</td>
<td>668.7.1†</td>
</tr>
<tr>
<td>R4</td>
<td>668.15.1</td>
<td>668.3.1</td>
<td>668.8.1</td>
<td>707.26.1</td>
</tr>
<tr>
<td>R5</td>
<td>668.3.4</td>
<td>668.8.2</td>
<td>668.14.2</td>
<td>668.15.11</td>
</tr>
</tbody>
</table>

Tissue collection is optimized for maize genome, and approximately 300 leaf punches/per pool were collected. This number varied based on the number of events in each pool. Three individuals were punched for each event, and the final number of punches per individual varied between 20-25 punches for the columns and rows.

†In LB1, The DNA extracted from the 707.26.1 event was pooled in C4 and R4 pools. In LB2 and RB1 the 707.26.1 event was prepared independently using different barcode (0054_Rev).

**DNA Extraction and Quantitation**

Genomic DNA was extracted from the collected leaf tissue using the modified Urea-based protocol described in APPENDIX A. (Cite Dellaporta). The DNA was quantified using the Promega Quantifluor dsDNA assay following manufacturer instructions.
DNA Fragmentation

DNA (5 ug) from each pool was sheared using Covaris ME220 ultrasonicator with miniTUBE (Red-PN 520066) following the conditions provided by the manufacturer (www.covaris.com/wp-content/uploads/pn_010301.pdf). The sheared DNA is cleaned using a PCR clean-up column (Qiagen) and eluted in 60uL EB.

T-DNA flank selection Protocol Steps

1. DNA end repair and dA-tailing This step uses NEBNext® Ultra™ End Repair/dA-Tailing Module (NEB E7442)

   Reaction Mix

<table>
<thead>
<tr>
<th>Item</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sheared DNA</td>
<td>55uL</td>
</tr>
<tr>
<td>End Repair Buffer (10x)</td>
<td>6.5uL</td>
</tr>
<tr>
<td>End Repair Enzyme mix</td>
<td>6.5uL</td>
</tr>
</tbody>
</table>

   Thermo-cycling parameters

<table>
<thead>
<tr>
<th>Step</th>
<th>Temperature</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Denaturation</td>
<td>20°C</td>
<td>30min</td>
</tr>
<tr>
<td>Denaturation</td>
<td>65°C</td>
<td>30min</td>
</tr>
</tbody>
</table>

   Cleanup This step requires the use of PacBio compatible SPRI beads. The item used was Aline Biosciences™ PCRClean DX® (C-1003), which were cleaned using PacBio instructions.

   The reaction was cleaned using 0.6x SPRI beads and eluted in 35uL of nuclease free water

2. Adapter Ligation The adapters were ligated to the end repaired fragments from the previous step. T4 Ligase from NEB was used for this (NEB M0202)

   Reaction Mix

<table>
<thead>
<tr>
<th>Item</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repaired DNA</td>
<td>34uL</td>
</tr>
<tr>
<td>Adapters (15uM LA/30 SA)</td>
<td>10uL</td>
</tr>
<tr>
<td>NEB ligation buffer (10x)</td>
<td>5uL</td>
</tr>
<tr>
<td>T4 DNA Ligase</td>
<td>1uL</td>
</tr>
</tbody>
</table>
**Thermo-cycling parameters**

<table>
<thead>
<tr>
<th>Step</th>
<th>Temperature</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Denaturation</td>
<td>20°C</td>
<td>60min</td>
</tr>
<tr>
<td>Denaturation</td>
<td>70°C</td>
<td>10min</td>
</tr>
</tbody>
</table>

3. Clean up Add 100uL water to ligation mix, and clean using 0.5x SPRI beads and elute in 50uL of nuclease free water

4. T-DNA target enrichment

   LB1: This was performed by linear amplification using biotinylated bLB or bRB primers.

   **Reaction Mix**

<table>
<thead>
<tr>
<th>Item</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter Ligated DNA</td>
<td>2.5uL</td>
</tr>
<tr>
<td>bLB Primer</td>
<td>2.5uL</td>
</tr>
<tr>
<td>Q5 Master Mix (2x)</td>
<td>12.5uL</td>
</tr>
<tr>
<td>Water</td>
<td>7.5uL</td>
</tr>
</tbody>
</table>

**Thermo-cycling parameters**

<table>
<thead>
<tr>
<th>Step</th>
<th>Temperature</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 Cycles of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Denaturation</td>
<td>98°C</td>
<td>20s</td>
</tr>
<tr>
<td>Annealing</td>
<td>70°C</td>
<td>20s</td>
</tr>
<tr>
<td>Extension</td>
<td>72°C</td>
<td>30s</td>
</tr>
<tr>
<td>Final Extension</td>
<td>72°C</td>
<td>2min30s</td>
</tr>
<tr>
<td>Hold at</td>
<td>4°C</td>
<td>Inf</td>
</tr>
</tbody>
</table>

RB2 and LB2: This was performed by Low cycle amplification with bLB or bRB + AP using Q5 master-mix from NEB

   **Reaction Mix**

<table>
<thead>
<tr>
<th>Item</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter Ligated DNA</td>
<td>2.5uL</td>
</tr>
<tr>
<td>bLB or bRB Primer</td>
<td>2.5uL</td>
</tr>
<tr>
<td>bAP Primer</td>
<td>2.5uL</td>
</tr>
<tr>
<td>Q5 Master Mix (2x)</td>
<td>12.5uL</td>
</tr>
<tr>
<td>Water</td>
<td>5.0uL</td>
</tr>
</tbody>
</table>
Thermo-cycling parameters

<table>
<thead>
<tr>
<th>Step</th>
<th>Temperature</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Denaturation</td>
<td>98°C</td>
<td>30s</td>
</tr>
<tr>
<td>15 Cycles of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Denaturation</td>
<td>98°C</td>
<td>20s</td>
</tr>
<tr>
<td>Annealing</td>
<td>69°C</td>
<td>20s</td>
</tr>
<tr>
<td>Extension</td>
<td>72°C</td>
<td>30s</td>
</tr>
<tr>
<td>Final Extension</td>
<td>72°C</td>
<td>2min30s</td>
</tr>
<tr>
<td>Hold at</td>
<td>4°C</td>
<td>Inf</td>
</tr>
</tbody>
</table>

5. Cleanup Add 50uL water to ligation mix, and clean using 0.5x SPRI beads and elute in 20uL of nuclease free water

6. Target selection Target selection using Streptavidin Dynabeads kilobase BINDER (ThermoFisher cat# 60101) and resuspended in 20uL

Barcode PCR Reaction Mix

<table>
<thead>
<tr>
<th>Item</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beads and DNA</td>
<td>20uL</td>
</tr>
<tr>
<td>Primer Mix</td>
<td>5uL</td>
</tr>
<tr>
<td>Q5 Master Mix (2x)</td>
<td>25uL</td>
</tr>
</tbody>
</table>

Thermo-cycling parameters

<table>
<thead>
<tr>
<th>Step</th>
<th>Temperature</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Denaturation</td>
<td>98°C</td>
<td>30s</td>
</tr>
<tr>
<td>15 Cycles of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Denaturation</td>
<td>98°C</td>
<td>20s</td>
</tr>
<tr>
<td>Annealing</td>
<td>70°C</td>
<td>20s</td>
</tr>
<tr>
<td>Extension</td>
<td>72°C</td>
<td>30s</td>
</tr>
<tr>
<td>Final Extension</td>
<td>72°C</td>
<td>2min30s</td>
</tr>
<tr>
<td>Hold at</td>
<td>4°C</td>
<td>Inf</td>
</tr>
</tbody>
</table>

7. Clean up Add 50uL water to ligation mix, and clean using 0.5x SPRI beads and elute in 40uL of nuclease free water

8. Quantification Quantify using Quantifluor dsDNA system, following manufacturer instructions

9. Library characterization Measure the library size using Agilent BioAnalyzer 2100 using the
high-sensitivity DNA chip.

**Target amplicon library construction**

10. Barcoded amplicons were pooled using the calculator (www.pacb.com/wp-content/uploads/HLA-Pooling-Calculator.xls)

Primer and Adapter Sequences

Adapter primers

Adapter primers used for the ligation step after DNA shearing

<table>
<thead>
<tr>
<th>Primer</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long Adapter (LA)</td>
<td>AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCT</td>
</tr>
<tr>
<td>Short Adapter (SA)</td>
<td>/5Phos/ GATCGGAAGAGCG /3AmMO/</td>
</tr>
</tbody>
</table>

Target Enrichment Primers

Two types of primers are used for initial target amplification and enrichment using streptavidin beads

1. biotinylated T-DNA specific primers for left and right borders

2. Adapter Primer

<table>
<thead>
<tr>
<th>Primer</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>biotinylated Left Border (bLB)</td>
<td>/bio/ GAAGCGAATTAGCTTGGCACTGG</td>
</tr>
<tr>
<td>biotinylated Right Border (bRB)</td>
<td>/bio/ TCGGGAAACCTGTGCCGTGCC</td>
</tr>
<tr>
<td>Adapter Primer (AP)</td>
<td>AATGATACGGCGACCACCG</td>
</tr>
</tbody>
</table>
Barcoded Primers

Barcoded primers contain three parts of sequences

1. 10-bp spacer
2. Barcode (PacBio recommended sequence)
3. T-DNA specific border (LB or RB) primer or Adapter primer

<table>
<thead>
<tr>
<th>Pool</th>
<th>Barcode #</th>
<th>Barcode ID</th>
<th>Spacer</th>
<th>Barcode</th>
<th>Left Border Primer</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>1</td>
<td>0001_Rev</td>
<td>AGCGATCGTA</td>
<td>TCAGACGATGCGTCAT</td>
<td>CCTGGCGTTACCCAACTTAATCG</td>
</tr>
<tr>
<td>R2</td>
<td>2</td>
<td>0009_Rev</td>
<td>AGCGATCGTA</td>
<td>CTGCGTGCTCTACGAC</td>
<td>CCTGGCGTTACCCAACTTAATCG</td>
</tr>
<tr>
<td>R3</td>
<td>4</td>
<td>0017_Rev</td>
<td>AGCGATCGTA</td>
<td>CATAGCGACTATCGTG</td>
<td>CCTGGCGTTACCCAACTTAATCG</td>
</tr>
<tr>
<td>R4</td>
<td>7</td>
<td>0029_Rev</td>
<td>AGCGATCGTA</td>
<td>GCTCGACTGTGAGAGA</td>
<td>CCTGGCGTTACCCAACTTAATCG</td>
</tr>
<tr>
<td>C1</td>
<td>9</td>
<td>0034_Rev</td>
<td>AGCGATCGTA</td>
<td>ACTCTCGCTCTGTAGA</td>
<td>CCTGGCGTTACCCAACTTAATCG</td>
</tr>
<tr>
<td>C2</td>
<td>11</td>
<td>0038_Rev</td>
<td>AGCGATCGTA</td>
<td>TGCTCGCAGTATCA</td>
<td>CCTGGCGTTACCCAACTTAATCG</td>
</tr>
<tr>
<td>C3</td>
<td>12</td>
<td>0040_Rev</td>
<td>AGCGATCGTA</td>
<td>CAGTGAGAGCGCGATA</td>
<td>CCTGGCGTTACCCAACTTAATCG</td>
</tr>
<tr>
<td>C4</td>
<td>15</td>
<td>0048_Rev</td>
<td>AGCGATCGTA</td>
<td>TCACACTCTAGAGCGA</td>
<td>CCTGGCGTTACCCAACTTAATCG</td>
</tr>
<tr>
<td>C5</td>
<td>17</td>
<td>0052_Rev</td>
<td>AGCGATCGTA</td>
<td>GCAGACTCTCACAACGC</td>
<td>CCTGGCGTTACCCAACTTAATCG</td>
</tr>
<tr>
<td>Dwarf</td>
<td>18</td>
<td>0054_Rev</td>
<td>AGCGATCGTA</td>
<td>GCAGACTCTCACAACGC</td>
<td>CCTGGCGTTACCCAACTTAATCG</td>
</tr>
</tbody>
</table>
### Barcoded Right Border Primers

<table>
<thead>
<tr>
<th>Pool</th>
<th>Barcode #</th>
<th>Barcode ID</th>
<th>Spacer</th>
<th>Barcode</th>
<th>Right Border Primer</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>1</td>
<td>0001_Rev</td>
<td>AGCGATCGTA</td>
<td>TCAGACGATGCGTCAT</td>
<td>AGAGCCGGTTTGCATATTGGAGC</td>
</tr>
<tr>
<td>R2</td>
<td>2</td>
<td>0009_Rev</td>
<td>AGCGATCGTA</td>
<td>CTGCGTGCTCTACGAC</td>
<td>AGAGCCGGTTTGCATATTGGAGC</td>
</tr>
<tr>
<td>R3</td>
<td>4</td>
<td>0017_Rev</td>
<td>AGCGATCGTA</td>
<td>CATAGCGACTATCGTG</td>
<td>AGAGCCGGTTTGCATATTGGAGC</td>
</tr>
<tr>
<td>R4</td>
<td>7</td>
<td>0029_Rev</td>
<td>AGCGATCGTA</td>
<td>GCTCGACTGTGAGAGA</td>
<td>AGAGCCGGTTTGCATATTGGAGC</td>
</tr>
<tr>
<td>C1</td>
<td>9</td>
<td>0034_Rev</td>
<td>AGCGATCGTA</td>
<td>ACTCTCGCTCTGTAAGA</td>
<td>AGAGCCGGTTTGCATATTGGAGC</td>
</tr>
<tr>
<td>C2</td>
<td>11</td>
<td>0038_Rev</td>
<td>AGCGATCGTA</td>
<td>TGCTCGCGATGTACACA</td>
<td>AGAGCCGGTTTGCATATTGGAGC</td>
</tr>
<tr>
<td>C3</td>
<td>12</td>
<td>0040_Rev</td>
<td>AGCGATCGTA</td>
<td>CAGTAGAGACGCGATA</td>
<td>AGAGCCGGTTTGCATATTGGAGC</td>
</tr>
<tr>
<td>C4</td>
<td>15</td>
<td>0048_Rev</td>
<td>AGCGATCGTA</td>
<td>TCAACCTCTAGAGCGA</td>
<td>AGAGCCGGTTTGCATATTGGAGC</td>
</tr>
<tr>
<td>C5</td>
<td>17</td>
<td>0052_Rev</td>
<td>AGCGATCGTA</td>
<td>GCAGACTCTACAGCGA</td>
<td>AGAGCCGGTTTGCATATTGGAGC</td>
</tr>
<tr>
<td>dwarf</td>
<td>18</td>
<td>0054_Rev</td>
<td>AGCGATCGTA</td>
<td>GATATATATCTACACAC</td>
<td>AGAGCCGGTTTGCATATTGGAGC</td>
</tr>
</tbody>
</table>

### Barcoded Adapter Primers

<table>
<thead>
<tr>
<th>Pool</th>
<th>Barcode #</th>
<th>Barcode ID</th>
<th>Spacer</th>
<th>Barcode</th>
<th>Adapter Primer</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>1</td>
<td>0001_Rev</td>
<td>AGCGATCGTA</td>
<td>ATGACGATCGTCAGTG</td>
<td>AATGATACGGCCGACCCACCC</td>
</tr>
<tr>
<td>R2</td>
<td>2</td>
<td>0009_Rev</td>
<td>AGCGATCGTA</td>
<td>GTCGTAAGAACCACAG</td>
<td>AATGATACGGCCGACCCACCC</td>
</tr>
<tr>
<td>R3</td>
<td>4</td>
<td>0017_Rev</td>
<td>AGCGATCGTA</td>
<td>CAGATACTCGTCATATG</td>
<td>AATGATACGGCCGACCCACCC</td>
</tr>
<tr>
<td>R4</td>
<td>7</td>
<td>0029_Rev</td>
<td>AGCGATCGTA</td>
<td>TCTCTCAGAGAGAGAC</td>
<td>AATGATACGGCCGACCCACCC</td>
</tr>
<tr>
<td>C1</td>
<td>9</td>
<td>0034_Rev</td>
<td>AGCGATCGTA</td>
<td>TCTACAGAGAACAGAG</td>
<td>AATGATACGGCCGACCCACCC</td>
</tr>
<tr>
<td>C2</td>
<td>11</td>
<td>0038_Rev</td>
<td>AGCGATCGTA</td>
<td>TGATATACGTACACAC</td>
<td>AATGATACGGCCGACCCACCC</td>
</tr>
<tr>
<td>C3</td>
<td>12</td>
<td>0040_Rev</td>
<td>AGCGATCGTA</td>
<td>TATCGCGCTCTACTCTG</td>
<td>AATGATACGGCCGACCCACCC</td>
</tr>
<tr>
<td>C4</td>
<td>15</td>
<td>0048_Rev</td>
<td>AGCGATCGTA</td>
<td>TCGCTCTAGAGTGTGA</td>
<td>AATGATACGGCCGACCCACCC</td>
</tr>
<tr>
<td>C5</td>
<td>17</td>
<td>0052_Rev</td>
<td>AGCGATCGTA</td>
<td>GCGTGAGAGGACTGTC</td>
<td>AATGATACGGCCGACCCACCC</td>
</tr>
<tr>
<td>dwarf</td>
<td>18</td>
<td>0054_Rev</td>
<td>AGCGATCGTA</td>
<td>GATATATATCTACACAC</td>
<td>AATGATACGGCCGACCCACCC</td>
</tr>
</tbody>
</table>
Hierarchical Precision

For gene $g$ with annotation $AS_i$ in annotation set and annotation $GS_j$ in the gold standard the $hPr$ is calculated as follows

$$hPr_{(g,AS_i,GS_j)} = \frac{GO_{AS_i} \cap GO_{GS_j}}{GO_{AS_i}}$$

Where:

$GO_{AS_i}$: GO terms inferred for annotation $AS_i$ by propagating the GO hierarchy till the root term

$GO_{GS_j}$: GO terms inferred for annotation $GS_j$ by propagating the GO hierarchy root term

For a gene $g$ with annotation $AS_i$ in annotation set in the GO ontology $o$ the $hPr$ is calculated as followed.

$$hPr_{(g,AS_i)} = \sum_{j \in GS_{g,o}} \frac{hPr_{(g,AS_i,GS_j)}}{|GS_{g,o}|}$$

Where:

$GS_{g,o}$ : GO terms annotated to gene $g$ in the ontology $o$ in gold standard $GS$

For gene $g$, $hPr$ for ontology $o$ is calculated as follows

$$hPr_{(g)} = \sum_{i \in AS_{g,o}} \frac{hPr_{(g,AS_i)}}{|AS_{g,o}|}$$

Where:

$AS_{g,o}$: GO terms annotated to gene $g$ in the ontology $o$ in the annotation set $AS$.

Note: Only genes with GO terms annotated in ontology $o$ in both $AS$ and $GS$ can be used for this calculation.

$hPr$ for a given annotation set $AS$ for the ontology $o$ is calculated as followed
\[
    hPr_{(AS,o)} = \sum_{g \in (AS_o \cap GS_o)} \frac{hPr(g)}{|AS_o \cap GS_o|}
\]

Where:

- \( AS_o \): Genes annotated in the annotation set \( AS \) in the ontology \( o \)
- \( GS_o \): Genes annotated in the gold standard \( GS \) in the ontology \( o \)

Hierarchical Recall

For gene \( g \) with annotation \( AS_i \) in annotation set and annotation \( GS_j \) in the gold standard the \( hRc \) is calculated as follows

\[
    hRc_{(g,AS_i,GS_j)} = \frac{GO_{AS_i} \cap GO_{GS_j}}{GO_{GS_i}}
\]

Where:

- \( GO_{AS_i} \): GO terms inferred for annotation \( AS_i \) by propagating the GO hierarchy till the root term
- \( GO_{GS_j} \): GO terms inferred for annotation \( GS_j \) by propagating the GO hierarchy root term

For a gene \( g \) with annotation \( GS_i \) in gold standard in the GO ontology \( o \) the \( hRc \) is calculated as followed.

\[
    hRc_{(g,GS_i)} = \sum_{i \in AS_{g,o}} \frac{hRc_{(g,AS_i,GS_j)}}{|AS_{g,o}|}
\]

Where:

- \( AS_{g,o} \): GO terms annotated to gene \( g \) in the ontology \( o \) in annotation set \( AS \)

For gene \( g \), \( hRc \) for ontology \( o \) is calculated as follows

\[
    hRc_{(g)} = \sum_{j \in GS_{g,o}} \frac{hRc_{(g,GS_j)}}{|GS_{g,o}|}
\]

Where:

- \( GS_{g,o} \): GO terms annotated to gene \( g \) in the ontology \( o \) in the gold standard \( GS \).

Note: Only genes with GO terms annotated in ontology \( o \) in both \( AS \) and \( GS \) can be used for this calculation.
\[ hRC(AS_o) = \sum_{g \in (AS_o \cap GS_o)} \frac{hPr_g}{|AS_o \cap GS_o|} \]

Where:

\( AS_o \): Genes annotated in the annotation set \( AS \) in the ontology \( o \)

\( GS_o \): Genes annotated in the gold standard \( GS \) in the ontology \( o \)

**Datasets**

Table 6: Public datasets used in the project

<table>
<thead>
<tr>
<th>Database</th>
<th>Type</th>
<th>Format</th>
<th>Version</th>
<th>Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAIR</td>
<td>Protein Sequences</td>
<td>fasta</td>
<td>TAIR 10</td>
<td><em>Arabidopsis thaliana</em></td>
</tr>
<tr>
<td>TAIR</td>
<td>GO Annotations</td>
<td>gaf 2.0</td>
<td>TAIR 10 (20170410)</td>
<td><em>Arabidopsis thaliana</em></td>
</tr>
<tr>
<td>Gramene 49</td>
<td>Gene Annotations</td>
<td>gff3</td>
<td>5b+</td>
<td><em>Zea mays</em></td>
</tr>
<tr>
<td>Gramene 49</td>
<td>GO Annotations</td>
<td>gaf 2.0</td>
<td>5b+</td>
<td><em>Zea mays</em></td>
</tr>
<tr>
<td>Phytozome 11</td>
<td>GO Annotations</td>
<td>tsv</td>
<td>5b+</td>
<td><em>Zea mays</em></td>
</tr>
<tr>
<td>Uniprot</td>
<td>Protein sequences</td>
<td>fasta</td>
<td>20170410</td>
<td>All species</td>
</tr>
<tr>
<td>Uniprot</td>
<td>Protein sequences</td>
<td>fasta</td>
<td>20170410</td>
<td>All plants</td>
</tr>
<tr>
<td>Uniprot</td>
<td>GO Annotations</td>
<td>gaf 2.0</td>
<td>20170410</td>
<td>All plants</td>
</tr>
<tr>
<td>Pfam</td>
<td>HMMs</td>
<td>hmm</td>
<td>27.0</td>
<td>All species</td>
</tr>
<tr>
<td>PANTHER</td>
<td>HMMs</td>
<td>hmm</td>
<td>10.0</td>
<td>All species</td>
</tr>
</tbody>
</table>

**Software Tools and Versions**

Table 7: Software tools used in the project. Exact parameters used are specified in Methods section of Chapter 4

<table>
<thead>
<tr>
<th>Software</th>
<th>Type</th>
<th>Version</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCBI-BLAST</td>
<td>Sequence similarity</td>
<td>2.6.0</td>
<td>(Altschul, Gish, Miller, Myers, &amp; Lipman, 1990)</td>
</tr>
<tr>
<td>HMMER</td>
<td>HMM scanning</td>
<td>3.1b1</td>
<td>(Finn, Clements, &amp; Eddy, 2011)</td>
</tr>
<tr>
<td>InterProScan</td>
<td>GO Annotation</td>
<td>5.15-55.0</td>
<td>(Jones et al., 2014)</td>
</tr>
<tr>
<td>PANNZER</td>
<td>GO Annotation</td>
<td>1.1</td>
<td>(Koskinen, Törönen, Nokso-Koivisto, &amp; Holm, 2015)</td>
</tr>
<tr>
<td>Argot2</td>
<td>GO Annotation</td>
<td>2.0 (Web Server)</td>
<td>(Falda et al., 2012)</td>
</tr>
<tr>
<td>FANN-GO</td>
<td>GO Annotation</td>
<td>One version only</td>
<td>(Clark &amp; Radivojac, 2011)</td>
</tr>
<tr>
<td>AIGO</td>
<td>GO Evaluations</td>
<td>0.1.0</td>
<td>(Defoin-Platel et al., 2011)</td>
</tr>
<tr>
<td>FASTX-Toolkit</td>
<td>Fasta manipulation</td>
<td>0.0.13</td>
<td>(Gordon &amp; Hannon, 2010)</td>
</tr>
</tbody>
</table>

**Annotation of Maize Genes**

**Obtaining Input Datasets**

1. Query Sequences
   1. Downloaded maize RefGen_v3 5b+ protein sequences as a fasta file from Gramene
2. Longest translated protein sequence among the transcript models for each gene was retained to represent a gene
3. Transcript model id was renamed to the gene model ID and generate a maize-filtered fasta file
4. Made a BLAST database using the maize-filtered fasta file

   ```
   makeblastdb -in 'maize-filtered.fa'
               -dbtype 'prot'
               -hash_index
               -out 'maize-filtered'
               -title 'maize-filtered'
   ```

2. Arabidopsis
   1. Downloaded Arabidopsis transcript protein sequences from TAIR v10
   2. Made an arabidopsis-filtered fasta by filtering for the transcript model with the longest protein sequence to represent each gene and rename transcript model IDs to gene model IDs
   3. Made a blast database using the arabidopsis-filtered fasta file

   ```
   makeblastdb -in 'arabidopsis-filtered.fa'
               -dbtype 'prot'
               -hash_index
               -out 'arabidopsis-filtered'
               -title 'arabidopsis-filtered'
   ```

4. Downloaded the Arabidopsis GO annotation GAF file arabidopsis.gaf from TAIR v10
5. Filter the GO annotations from arabidopsis.gaf to retain the annotations which have curated evidence codes and convert to a GAF file named filtered-arabidopsis.gaf
   a. Selected [EXP, IDA, IPI, IMP, IGI, IEP, ISS, ISO, ISA, ISM, IGC, IBA, IBD, IKR, IRD, RCA, TAS, IC]
   b. Omitted [IEA, ND, NAS]

3. UniProt Plants
   1. Curated GO annotations for plants were downloaded from UniProt using QuickGO

   ```
   wget -O annot/tmp.gaf
   "http://www.ebi.ac.uk/QuickGO/GAnnotation\?format=gaf&limit=-1&q=!evidence=IEA,ND,NAS\&tax=33090"
   ```

2. GO annotations from Top 10 plant species with highest number of annotations from tmp.gaf were filtered and saved to uniprot-hc-plant.gaf
3. NCBI taxonomy IDs of the top 10 plant species at the time of downloading UniProt data are
   (15368, 29760, 3055, 3218, 3694, 3702, 3847, 39947, 4081, 4558)
4. Plant Protein sequences with curated GO annotations were downloaded from UniProt using QuickGO tool
wget -O fa/tmp.fa
"http://www.ebi.ac.uk/QuickGO/GAnnotation?format=fasta&limit=-1&q=!evidence=IEA,ND,NAS&tax=33090"

5. Convert the IDs in tmp.fa to match the protein IDs in the uniprot-hc-plant.gaf file and make a new fasta file tmp2.fa
6. tmp2.fa was filtered for IDs present in the uniprot-hc-plant.gaf and save to uniprot-hc-plant.fa
7. Make a BLAST database for the uniprot-hc-plant.fa

makeblastdb -in 'uniprot-hc-plant.fa'
-dbtype 'prot'
-out 'uniprot-hc-plant'

**Sequence-Similarity methods**

1. *Arabidopsis*
   1. Used the maize-filtered fasta as query and searched against uniprot-hc-plant blast database
      
      blastp -db 'arabidopsis-filtered'
      -query 'maize-filtered.fa'
      -max_target_seqs '20'
      -out 'mz-ara-aa.txt'
      -outfmt '6 qseqid sseqid qlen qstart qend slen sstart se
      -num_threads '16'
      
      evaluate bitscore score length pident nident gaps'

2. Used the arabidopsis-filtered fasta as query and searched against maize-filtered database
   
   blastp -db 'maize-filtered'
   -query 'arabidopsis-filtered.fa'
   -max_target_seqs '20'
   -out 'ara-mz-aa.txt'
   -outfmt '6 qseqid sseqid qlen qstart qend slen sstart se
   -num_threads '16'

3. Used custom R script to read ‘ara-mz.txt’ & ‘mz-ara-aa.txt’ and obtained Reciprocal-Best-Hits (RBH) saved as ‘maize-v3-vs-tair10.rbh.txt’
   
   Specific parameters used within the script
   • BLAST hits were filtered by using an e-value cut off of 10e-10 form both datasets
   • Hits were ranked by the score in descending order

4. Used custom R script to read ‘filtered-arabidopsis.gaf’ and maize-v3-vs-tair10.rbh.txt inherited the curated GO terms from *Arabidopsis* to maize and created a ‘maize-arabidopsis.gaf’ file
2. **UniProt Plants**
   1. Used the maize-filtered fasta as query and searched against uniprot-hc-plant database

   ```bash
   blastp -db 'uniprot-hc-plant'
   -query 'maize-filtered.fa'
   -max_target_seqs '20'
   -out 'mz-uniprot-aa.txt'
   -outfmt '6 qseqid sseqid qlen qstart qend slen sstart se
   evalue bitscore score length pident nident gaps'
   -num_threads '16'
   ```

   2. Used the uniprot-hc-plant fasta as query and searched against maize-filtered database

   ```bash
   blastp -db 'maize-filtered'
   -query 'uniprot-hc-plant.fa'
   -max_target_seqs '20'
   -out 'uniprot-mz-aa.txt'
   -outfmt '6 qseqid sseqid qlen qstart qend slen sstart se
   evalue bitscore score length pident nident gaps'
   -num_threads '16'
   ```

   3. Used custom R script to read ‘uniprot-mz-aa.txt’ & ‘mz-uniprot-aa.txt’ and obtained Reciprocal-Best-Hits (RBH) saved as ‘maize-v3-vs-uniprot.rbh.txt’

   Specific steps and parameters used within the script
   - BLAST hits were filtered by using an e-value cut off of 10e-10 form both datasets
   - Hits were ranked by the score in descending order
   - RBH assignment was performed for each of the 10 plant species in uniprot-hc-plant.gaf and maize the steps were exactly similar to *Arabidopsis*

   4. Used custom R script to read ‘uniprot-hc-plant.gaf’ and maize-v3-vs-uniprot.rbh.txt inherited the curated GO terms from Plants in Uniprot dataset to maize and created a ‘maize-uniprot.gaf’ file

**Domain-presence method**

1. InterProScan5 pipeline was downloaded an configured in a local server
2. Necessary PANTHER database was downloaded added to the data location
3. InterProScan5 pipeline was run on maize-filtered.fa file to assign putative domains and assign GO terms to genes

   ```bash
   interproscan.sh -i 'maize-filtered.fa'
   -goterms
   -f 'tsv'
   -o 'maize-filtered.iprs.out'
   ```
4. A custom Rscript was used to convert ‘maize-filtered.iprs.out’ to ‘maize-interproscan.gaf’

**Mixed-method pipelines**

1. Common Pre-processing steps
   1. All protein sequences was downloaded from UniProt as uniprot.fa
   2. The uniprot.fa was used to make a blast database
      
      ```
      makeblastdb   -in ‘uniprot.fa’
      -dbtype ‘prot’
      -out ‘uniprot’
      ```
   3. Pfam-A and Pfam-B HMM models were downloaded and uncompressed
   4. A Pfam-AB file was created by concatenating Pfam-A and Pfam-B
      
      ```
      cat "Pfam-A.hmm" and "Pfam-B.hmm" > "Pfam-AB.hmm"
      hmmpress "Pfam-AB.hmm"
      ```

2. Argot2
   1. “maize-filtered.fa” was split into smaller files with only 5000 sequences per file. Argot2 webserver allows a max 5000 sequences to be batch processed at a time.
   2. BLASTP was run against the UniProt database for each split file
      
      ```
      blastp  -outfmt ‘6 qseqid sseqid evalue’
      -num_threads ‘16’
      -query ‘maize-filtered.1.fa’
      -db ‘uniprot’
      -out ‘maize-filtered.1.blast’
      ```
   3. Hmmer was used for each split file to search against the Pfam HMMs
      
      ```
      hmmscan   --cpu ‘16’
      --tblout ‘maize-filtered.1.hmmer’
      ‘Pfam-AB.hmm’
      ‘maize-filtered.1.fa’
      ```
   4. All output files were compressed as zip files
   5. Output files from BLAST and Hmmer for each split fasta file was submitted as a new job for batch processing on the Argot2 website [http://www.medcomp.medicina.unipd.it/Argot2/](http://www.medcomp.medicina.unipd.it/Argot2/)
   6. Argot2 Results for each part was downloaded and renamed according to input file names
   7. Argot2 results were converted to GAF 2.0 format using an Rscript
   8. This was saved as ‘maize-argot2.gaf’ file

3. PANNZER
   1. PANNZER files and database were downloaded from [http://ekhidna.biocenter.helsinki.fi/pannzer/Download.html](http://ekhidna.biocenter.helsinki.fi/pannzer/Download.html)
   2. PANNZER tool was setup according to the instructions provided in the manual
   3. BLASTP was used to query the maize-filtered split fasta files used for Argot2 against the uniprot blast database
4. A config file required for PANNZER for each BLAST xml file output in the previous step

```
[GENERAL_SETTINGS]
INPUT_TYPE=BLASTXML
INPUT_FILE=maize-filtered.1.xml
XML=True
DATA_FOLDER=PANNZER/db/
DB=uniprot
RESULT_FOLDER=output/
RESULT_BASE_NAME=maize-filtered.1
INPUT_BASE_NAME=Prefix_of_the_desc_file
INPUT_FOLDER=xml/
QUERY_TAXON=4577
GET_TAXON=False
GENERATE_IDF=False
MULTIPLE_SPECIES=False

[TRESHOLD_VALUES]
BITSCORE=50
SEQUENCE_LENGTH=20
IDENTITY_PERCENT=50
E-VALUE=0
TARGET_COVERAGE=0.6
QUERY_COVERAGE=0.6
INFORMATIVE=30
INFORMATIVE_HITS=100
CLUSTER=0.3

[MYSQL]
SQL_DB_HOST=localhost
SQL_DB_PORT =
SQL_DB_USER = pannzer
SQL_DB_PASSWORD = pannzer
SQL_DB = pannzer

[TAXONOMY]
DB=taxonomy-all.tab
CALCULATE=True
NODE_SELECTOR=1
TRACK_GROUPS=False
TRACKED_GROUPS=
ONLY_ONE_HIT_PER_SPECIE=False

[GO]
```
WRITE_GO=True
OBO=gene_ontology_ext.obo
ID_MAPPING=idmapping_selected.tab
ENZYME=enzyme.dat

[LEVEL_OF_PRINTING]
SIMPLE_OUTPUT=True
CLUSTER=True
CLUSTER_MEMBERS=False
ALL=False
ERROR=True
DEBUG=False
INFO=False

[EVALUATION] ### IF YOU DON'T KNOW WHAT YOU ARE DOING, DON'T CHANGE ANYTHING FROM THIS ON!!!!
PRINT_EVAL=False
TEST=False
OTHER=False

5. PANNZER was run for each config file
6. All results files from each split fasta files were concatenated and converted to a GAF 2.0 file
7. GAF file was saved as ‘maize-pannzer.gaf’

4. FANN-GO
1. FANN-GO tool was downloaded from
   http://montana.informatics.indiana.edu/fanngo/fanngo.html,
   uncompressed and installed.
2. The “maize-filtered.fa” file was used as input and FANN-GO was run using the following code

```bash
echo off all
cd code
[Headers, Sequences] = fastaread('maize-filtered.fa')
PRED=MAIN(Sequences)
Headers = transpose(Headers)
Headers = regexprep(Headers, ' .*', '')
tnames = horzcat('gene_id', PRED.accessions)
tnames = strrep(tnames, ':', '_')
scores = num2cell(PRED.scores)
all_scores = horzcat(Headers, scores)
all_scores = cell2table(all_scores)
all_scores.Properties.VariableNames = tnames
writetable(all_scores, '../scores.txt', 'Delimiter', '	')
```
3. “scores.txt” was converted into a GAF 2.0 file and saved as ‘maize-fanngo.gaf’
Selection and Cleaning of Maize Annotations

Selection of high-confidence mixed-method annotations

1. Score thresholds determined for the mixed-method pipeline annotations by evaluation against the MaizeGDB gold standard dataset. (Please refer to the main paper for how the score thresholds were determined)
   a. Argot2
      i. BP : 0.15
      ii. CC : 0.05
      iii. MF : 0.15
   b. FANN-GO
      i. BP : 0.3
      ii. MF : 0.65
   c. PANNZER
      i. BP : 0.4
      ii. CC : 0.2
      iii. MF : 0.55

2. ’maize-argot2.gaf’, ‘maize-fanngo.gaf’ and ‘maize-pannzer.gaf’ files were filtered to retain annotations with a score greater than or equal to the thresholds mentioned in the previous step

List of component datasets

1. Sequence-Similarity
   a. maize-arabidopsis.gaf
   b. maize-uniprot.gaf
2. Domain-Presence
   a. maize-interproscan.gaf
3. Mixed-method Pipelines
   a. maize-argot2.gaf
   b. maize-pannzer.gaf
   c. maize-fanngo.gaf

Cleaning duplications

1. Datasets mentioned in section 0 were cleaned for duplicate annotations
2. Duplicate annotation is when the same GO term is annotated to the same gene more than one time
3. Duplicate annotations were replaced by a single instance of the Gene-GO term pair independently for each dataset
4. The unique annotations for each component dataset was saved as gaf file

Cleaning redundancy

1. Redundancy is the annotation of one or more ancestral GO terms to a gene which is annotated to a more specific offspring GO term
2. Unique datasets mentioned in section 0 were cleaned by removing annotations with redundant GO terms
3. The non-redundant annotations for each component dataset was saved as gaf file

Making aggregate dataset

1. All the non-redundant component datasets from section 0 were concatenated together
2. The duplication introduced by merging multiple component datasets was cleaned using the steps from section 0
3. The redundancy introduced by merging multiple component datasets was cleaned using the steps from section 0
4. The non-redundant aggregate dataset was saved as ‘maize-aggregate.gaf’