GENETIC ASSOCIATIONS IN ACUTE LEUKEMIA PATIENTS AFTER MATCHED UNRELATED DONOR ALLOGENEIC HEMATOPOETIC STEM CELL TRANSPLANTATION

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Abbas A Rizvi

Graduate Program in Pharmaceutical Sciences

The Ohio State University

2019

Dissertation Committee:

Lara E Sucheston-Campbell, MS, PhD, Adviser

Guy Brock, PhD

Moray Campbell, PhD

Shili Lin, PhD

Copyright © Abbas A Rizvi 2019

All Rights Reserved

ABSTRACT

Here I will be writing an abstract that summarizes my dissertation results

DEDICATION

Dedicated to Ezgi, my parents, my siblings and their kids, and my sweet little pooch Bernie.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my adviser Dr. Lara Sucheston-Campbell. She has been an incredible mentor for me and has set me up with any and every opportunity that I had my eyes set on.

I am forever indebted to Dr. Barbara Foster, for taking a chance on me and giving me the opportunity to get an interview for the PhD program at RPCI, at the eleventh hour before the academic year began in 2015.

Thank you to Dr. Martin Morgan for taking time out of his extremely busy schedule to work with me one-on-one and teach me the inner workings of R.

I would not have gone done the road of computational biology/bioinformatics if it were not for two very important mentors of mine: Dr. Moray Campbell and Dr. Sebastiano Battaglia. Dr. Campbell set me up with a life defining opprtunity: the Cancer and Systems Biology program at the University of Luxembourg and VU University Amsterdam. And Dr. Battaglia for his mentorship during my Master's thesis.

And thank you to Ezgi Karaesmen – if it were not for you, I truly believe I would not had the level of success I have had for past four years. Your love, your support, your intelligence, and your ability to challenge me on my scientific thought, my coding, and my ability to think have been instrumental my development as a scientist – I cannot thank you enough.

This work was supported by the National Institute of Health/National Heart, Blood, and Lung Institute R01HL102278 and National Institute of Health/National Cancer Institute R03CA188733. Support provided by the Center for Computational Research at the University at Buffalo.

VITA

2012	B.S. Biology, SUNY Fredonia
2015	M.Sc. Integrated Systems Biology, University of Luxembourg
2015	M.S. Natural Sciences, University at Buffalo
2016-present	Graduate Research Associate, Department of Pharmaceutics, The Ohio State University

Publications

[1] Abbas A. Rizvi, Ezgi Karaesmen, Martin Morgan, Leah Preus, Junke Wang, Michael Sovic, Theresa Hahn, and Lara E. Sucheston-Campbell, *gwasurvivr: an R package for genome wide survival analysis*, Bioinformatics (2018).

[2] Mark D. Long, Prashant K. Singh, James R. Russell, Gerard Llimos, Spencer Rosario, Abbas A. Rizvi, Patrick R. van den Berg, Jason Kirk, Lara E. Sucheston-Campbell, Dominic J. Smiraglia, and Moray Campbell, *The miR-96 and RARγ signaling axis governs androgen signaling and prostate cancer progression*, Oncogene (2018).

[3] Lara E. Sucheston-Campbell, Alyssa I. Clay-Gilmour, William E. Barlow, G. Thomas Budd, Daniel O. Stram, Christopher A. Haiman, Xin Sheng, Li Yan, Gary Zirpoli, Song Yao, Chen Jiang, Owzar Kouros, Dawn Hershman, Kathy S. Albain, Daniel F. Hayes, Halle C. Moore, Timothy J. Hobday, James A. Stewart, Abbas A. Rizvi, Claudine Isaacs, Muhammad Salim, Jule R. Gralow, Gabriel N. Hortobagyi, Robert B. Livingston, Deanna L. Kroetz, and Christine B. Ambrosone, *Genome-wide meta-analyses identifies novel taxane-induced peripheral neuropathy-associated loci*, Pharmacogenetics and Genomics **28** (2018), no. 2, 49-55.

[4] Ezgi Karaesmen, Abbas A. Rizvi, Leah M. Preus, Philip L. McCarthy, Marcelo C. Pasquini, Kenan Onel, Xiaochun Zhu, Stephen Spellman, Christopher A. Haiman, Daniel O. Stram, Loreall Pooler, Xin Sheng, Qianqian Zhu, Li Yan, Qian Liu, Qiang Hu, Amy Webb, Guy Brock, Alyssa I. Clay-Gilmour, Sebastiano Battaglia, David Tritchler, Song Liu, Theresa Hahn, and Lara E. Sucheston-Campbell, *Replication and validation of genetic polymorphisms as-*

sociated with survival after allogeneic blood or marrow transplant, Blood **130** (2017), no. 13, 1585-1596.

[5] Alyssa I. Clay-Gilmour, Theresa Hahn, Leah M. Preus, Kenan Onel, Andrew Skol, Eric Hungate, Qianqian Zhu, Christopher A. Haiman, Daniel O. Stram, Loreall Pooler, Xin Sheng, Li Yan, Qian Liu, Qiang Hu, Song Liu, Sebastiano Battaglia, Xiaochun Zhu, AnneMarie W. Block, Sheila N. J. Sait, Ezgi Karaesmen, Abbas A. Rizvi, Daniel J. Weisdorf, Christine B. Ambrosone, David Tritchler, Eva Ellinghaus, David Ellinghaus, Martin Stanulla, Jacqueline Clavel, Laurent Orsi, Stephen Spellman, Marcelo C. Pasquini, Philip L. McCarthy, and Lara E. Sucheston-Campbell, *Genetic association with B-cell acute lymphoblastic leukemia in allogeneic transplant patients differs by age and sex*, Blood Advances **1** (2017), no. 20, 1717-1728.

[6] Abbas A. Rizvi, *Reprogramming androgen receptor and lysine-specific demethylase 1 transcriptome in castration-resistant prostate cancer*, University at Buffalo, 2015.

[7] Maxwell S. DeNies, Jordan Johnson, Amanda B. Maliphol, Michael Bruno, Annabelle Kim, Abbas A. Rizvi, Kevyn Rustici, and Scott Medler, *Diet-induced obesity alters skeletal muscle fiber types of male but not female mice*, Physiological Reports **2** (2014), no. 1, e00204.

Fields of Study

Major Field: Pharmaceutical Sciences

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: Introduction

Broadly, this dissertation examines germline genetic variation in the context matched unrelated donor (MUD) hematopoetic stem cell transplantation (HSCT). This dissertation seeks to identify and characterize genetic variants in patients who have acute myeloid leukemia (AML), acute lymphoblastic leukemia (ALL), or myelodysplastic syndrome (MDS) and received an HSCT from an HLA matched-unrelated donor (MUD). This dissertation also seeks to enhance the computational workflows that are used in these fields. The significance is three-fold, first, we can identify clinically relevant markers that may improve donor selection beyond traditional methods; second, we can characterize pre-transplant risk of disease or transplant related death within the first year; and third, we can help facilitate other researchers studying similar problems with similar data by developing open-source software.

The project that this dissertation contributes to has the opportunity to be a real life example of a translational study (from computational analysis of biological data to bedside). The dissertation is broken down into 6 chapters. This chapter (Chapter 1) first introduces genetic association studies and important related concepts (genetic and statistical) that are needed to appreciate and understand the underlying analysis. Chapter 1 also introduces allogeneic HSCT and DISCOVeRY-BMT genome wide-association study (GWAS), including the corresponding clinical data in detail. Chapter 2 will discuss a replication and validation study of all previous literature that examined genetic variation in the same context as our study, which we published on in *Blood*. Chapter 3 discusses the R package that we developed and it details the development and testing procedures executed. The package

is available on R/Bioconductor and was published in *Bioinformatics*. Chapter 4 is the application of the R package and custom pipeline developed to perform large automated GWAS, specialized to our lab, but can be generalized to other large scale projects. Chapter 5 discusses the discovery and inference of markers in ALL donor and recipient pairs. This dissertation ends with Chapter 6, which comprises preliminary data on genetic contributions to early death after transplant (the first 100 days) and the future directions that should be undertaken.

Genetic Association Studies

Genetic association studies test for correlations between genetic variation as it relates to disease risk or to physical quantitative traits (i.e. height or weight) (C. M. Lewis and Knight 2012). These studies have been successful in identifying certain variants as being predictive of disease susceptibility or drug response and have helped us understand that many diseases have complex genetic signatures that need to be further understood (Visscher et al. 2012). The human genome consists of over 3 billion base pairs, all of which are contained in every nucleated cell in the body. A genome sequence is the complete collection of all nucleotides (A, C, T, or G for DNA genomes) that make up all the chromosomes in individuals or species (Lander et al. 2001). The vast majority of nucleotides (>99.5%) are identical between individuals within a species, however, genetic variation arises within individuals and populations over time and different spaces. Indeed, the fundamental source of genetic variation is mutation, where permanent alterations occur to a single nucleotide or larger structural changes in the genome of a species. A mutation at single position (locus) in a DNA sequence that occurs in at least 1% of a population is called a single nucleotide polymorphism (SNP). SNPs are the most widely used marker to describe genetic variation. Larger structural variations may include mi-

crosatellite regions, insertion/deletions (indels), copy number variations (CNVs), or variable-tandem repeats (VNTRs) all of which have importance in understanding the genetic architecture and disease etiology (Sudmant et al. 2015). For purposes of this document, unless otherwise specified, genetic variation will refer to single nucleotide polymorphisms (SNPs). SNPs will also be referred to as simply as polymorphisms, genetic markers, or markers interchangeably.

Different forms of the same variant are called *alleles*. For diploid organisms, one allele is passed from each parent. When the same variation is passed both parents it is called homozygous and when they are different it is known as heterozygous. When mutations are passed between generations they are known as *germline mutations*. Importantly, in humans (and other eukaroytes), genetic recombination occurs during meiosis, where large chunks of genetic materials are exchanged and shuffled between parents and their offspring. A particular combination of alleles that lie on the same chromosome are called *haplotypes*.

While genetic variation only occurs in 0.1% to 0.5% of the human genome, modern genomic tools have revealed even in this small proportion of the genome, the underlying architecture is very complex. Sequence variations can occur coding regions of genes, non-coding regions of genes, or intergenic regions. SNPs that are within a coding regions are of two types: synonymous and non-synonymous SNPs. Synonymous SNPs do not change amino acid sequences and therefore do not change protein structure, while non-synonymous SNPs change amino acid sequences. Non-synonymous SNPs are further stratified into two types: missense and nonsense polymorphisms. Missense SNPs result in codon changes that code for different amino acids (Z. Shi and Moult 2011). Nonsense SNPs are genetic alterations that yield a premature stop codon that often yield a non-functional or truncated protein product. Polymorphisms that are in non-coding regions may alter impor-

tant transcriptional properties such as gene splicing, transcription factor binding, or messenger RNA (mRNA) decay (Green et al. 2003). SNPs may affect gene expression (and may be upstream or downstream from a gene) are called *expression quantiative trait loci* (eQTL).

Historically, gene mapping studies were used to determine associations between genomic DNA sequence variations and phenotypic variability (Visscher et al. 2012). These studies were quite successful, particularly in Mendelian traits (e.g. single gene disorders) (Botstein and Risch 2003). Over the past two decades, research has increasingly evolved from looking at specific regions of interest (candidate gene association studies) to more agnostic approaches that investigate larger portions of the genome, such as genome wide association studies (GWAS), whole-exome sequencing studies (WES studies) and whole-genome sequencing studies (WGS studies) (Timpson et al. 2018). This dissertation is primarily focused on GWAS, specifically the application of using GWAS in the context of identifying common variants in after hematopoetic stem cell transplantation (HSCT), where more details will be discussed in subsequent subsections of this chapter.

GWAS employ genotyping microarrays to measure genetic variation – and they have become the standard platform in academic and industry to test for association of phenotype with common genetic variants. Common genetic variants are defined as those with a *minor allele frequency* (MAF) of $\geq 1\%$ and rare variants are defined as those with a MAF $\leq$ than 1%. Genotyping microarrays are designed to contain common variants but optionally can contain rare variants. GWAS ask if the allele of a genetic variant is found more often than what would be expected than by random chance in individuals with the phenotype of interest (e.g. the disease being studied). If the variant (one allele) occurs more in those affected by the disease than those without the disease, then the variant deemed as being *associ-*

ated with the disease. Nonetheless, GWAS have been very successful at revealing new pathways involved in disease, but often the post-GWAS understanding of the associations is poorly understood. That is, identification of causal variants, biological relevance and the interaction that these associations have with other genetic or environmental factors.

Linkage Disequilibrium

GWAS are heavily based on the principal of linkage disequilibrium (LD) at the population level. LD is the non-random dependence of allele frequencies at two more loci in the general population (Jorde 2000). LD reflects the relationship between alleles at different loci. In other words, LD is a measure of two alleles or specific sequences being inherited together. The unit of measure for LD is r^2 (squared correlation coefficient) (J. K. Pritchard and Przeworski 2001). In general, loci that are in close proximity exhibit stronger LD (J. K. Pritchard and Przeworski 2001). Perfect LD ($r^2 = 1$) means that no recombination occurred on this chunk of genome. Regions that are further apart on a chromosome exhibit weaker LD (D. E. Reich et al. 2001). Low LD means that recombination occurred and that there are lots of possible rearrangements that may have occurred during meiosis. LD decay influences the number of SNPs needed to "tag" a haplotype, and that number of SNPs is just a small subset of the number of segregating polymorphisms in the population (D. E. Reich et al. 2001). Knowledge of haplotype structure makes it possible to retrieve more information from GWAS. Tagging SNPs with known haplotype block structure can capture much of the genetic information in a region.

With the rapid growth of genetic association studies and statistical methods assessing genetic variation, researchers routinely exploit LD to map regions in the human genome. While costs of genotyping have lowered over the past decade, a

major barrier to overcome when conducting large scale studies, was the expense of searching the entire genome for disease associations (International HapMap Consortium 2005). The International HapMap Project (International HapMap Consortium 2005) was the first attempt to address these challenges. HapMap was a large scale multi-institutional international project that finely mapped common genetic variation (or establish a "haplotype map"). HapMap demonstrated that genomic blocks are shared in common areas across continental population. HapMap alleviated high costs of studies by preferentially selecting 'tag' SNPs that covered the entire genome, and due to LD structure, inference could be drawn about nearby variants that were not genotyped (Bakker et al. 2005). HapMap ended with a catalogue across several populations for 420 haplotypes at 3.5 million SNPs(International HapMap Consortium et al. 2007). Afterwards, the 1000 Genomes Project, with similar ambitions to HapMap, aimed to create a more complete and thorough catalogue of human genetic variation, which could be leveraged for GWAS investigating disease (1000 Genomes Project Consortium 2015). The consortium aimed to discover >95% of variants with MAF as low as 1% across the genome, as well as estimate population specific allele frequencies, haplotype maps and LD patterns of alleles (1000 Genomes Project Consortium et al. 2010). The results provided a more comprehensive picture of human genetic variation than what was previously available (5,008 haplotypes at over 88 million SNPs in 26 worldwide populations) (1000 Genomes Project Consortium et al. 2010). And even more recently, the Haplotype Reference Consortium (HRC) has described nearly 65,000 human haplotypes at ~40 million SNPs via whole-gene sequence data from predominantly European ancestry (McCarthy et al. 2016). Other reference panels have been developed, particularly population specific ones, but are beyond the scope of this document.

Genetic Imputation

Genetic imputation (will also be simply referred to as imputation) has had significant contributions to genetic association studies. Imputation can be defined as predicting unobserved genotypes that were not directly assayed in a sample of individuals. The term refers to when a reference panel of haplotypes at set of a SNPs is used to impute SNPs that have been genotyped at a subset of that set of SNPs (Marchini et al. 2007). Genotype imputation is useful for three reasons: (1) by boosting the number of SNPS that can be tested for association, thus increasing the power of the study, (2) homogenizes variant sets for meta-analyses, and (3) help control false positive for which genotype calling is challenging (Marchini and Howie 2010).

Today, several reference panels are available, such as (but not limited to) HapMap2 (International HapMap Consortium 2005), 1000 Genomes Phase 3 (1000 Genomes Project Consortium 2015), HRC (McCarthy et al. 2016), which differ by the number of samples, sites (chromosomes 1-22, X), and number of haplotypes. These reference panels are widely used to carry out accurate imputation in studies. HRC can impute SNPs with MAF as low as 0.1% (McCarthy et al. 2016). Most studies take a two step approach that first will impute the missing genotypes using the reference panel without consideration of the phenotype. The imputed genotypes are then tested for association with the phenotype in the second pass. Multiple phenotypes can be tested for association without the need for re-imputation.

Several imputation algorithms and software packages are available as standalone software, such as IMPUTE2 (B. N. Howie, Donnelly, and Marchini 2009), MaCH (Li et al. 2010), BEAGLE (B. L. Browning and Browning 2016) or from imputation web services, such as the Sanger imputation server (McCarthy et al.

2016, R. Durbin (2014)) or the University of Michigan imputation server (Das et al. 2016). The details of the imputation methods are beyond the scope of this dissertation, but very briefly, each algorithm is an extension of the hidden Markov model (HMMs) to carry out inference when modeling LD or haplotype estimation (also called phasing) (N. Li and Stephens 2003). The imputation methods vary in terms slight methodological differences in estimating haplotypes, computational performance and error rates.

Meta-analysis

Although GWAS have been successful at identifying novel loci that are associated to some disease or trait, the finding typically have modest effects and large sample sizes are needed to detect common variants with small effect sizes. In order to improve the power to detect variants with small effect sizes, meta-analyses have been used. Meta-analysis using summary statistics has been important for GWAS of complex genetic diseases and traits (Bakker et al. 2008). Researchers combine the effects of multiple studies without having to integrate both genotype and phenotype data. After imputation, meta-analysis is useful for different cohorts that are used on different genotyping chips to boost power.

A popular tool to perform meta-analysis on GWAS is METAL (Willer, Li, and Abecasis 2010). METAL can combine either test statistics and standard errors, or p-values across studies (while taking direction of effect and sample size into account). The results are combined using fixed-effects or random-effects models.

Hematopoetic Stem Cell Transplantation

Blood cells continuously go through a self-renewing maturation process from less differentiated precursor cells to mature cells in a process called hematopoesis

(Copelan 2006). The process begins with hematopoetic stem cells (HSCs) which are located in center of bone marrow. HSCs differentiate into either lymphoid or myeloid progenitor cells and further develop into one of three lineages: red blood cells (erythrocytes), lymphocytes (T-cells, B-cells, and natural killer (NK) cells), and myeloid cells (granulocytes, megakaryocytes, and macrophages). All of these blood cells have vital roles in the human body. Tumors arise from malignant stem cells that usually originate from normal stem cells but retain the self-renewal property. Leukemic cells are limited in their ability to proliferate and incessantly are replenished from leukemic stem cells. Acute leukemias are characterized by the rapid increase of immature blood cells, such that the bone marrow is unable to produce health blood cells. Immediate treatment is required. Acute leukemias can be treated with some combination of chemotherapy, radiation therapy or an HSCT. When all other treatment options have been exhausted, an HSCT is used as last resort.

HSCT is an established therapeutic procedure that is used as a potentially curative treatment for life-threatening congenital or acquired blood disorders (malignant or non-malignant) (Henig and Zuckerman 2014). HSCT involves the intravenous infusion of autologous or allogeneic hematopoietic progenitor cells to restore normal function in patients whose bone marrow is compromised. Autologous HSCT involves self-donation of marrow stem cells, whereas allogeneic HSCT is when stem cells are transferred from a HLA-matched related donor (MRD) or a HLA-matched unrelated donor (MUD). Although a matched sibling donor is preferred, only approximately 30% of patients who may benefit from HSCT have such a donor available. In the United States, the number of allogeneic transplants yearly has dramatically risen over the past decade, across all diseases (M. Pasquini et al. 2013). Patients with acute myeloid leukemia (AML), acute

lymphoblastic leukemia (ALL), or myelodysplastic syndrome (MDS), represent the largest group treated with allogeneic HSCT. While both patient care and matching has improved over the past few decades almost half of all high-resolution 10/10 HLA MUD-HSCT recipients die within one-year post-transplant due to either their disease or transplant-related causes (M. Pasquini et al. 2013). These trends also show transplant-related causes are a larger contributor to mortality within the first 100-days post-transplant and shift towards primary disease after approximately six months post-transplant (D'Souza et al. 2017). Reducing TRM without increasing risk of disease death and vice versa continue to represent a substantial clinical challenge.

The four elements of HSCT are:

1. Graft Source

The graft sources are either from bone marrow or peripheral blood stem cells (PBSCs). For bone marrow grafts, hematopoetic stem cells (HSCs) are typically extracted from the center of the posterior iliac crest (pelvis) using a large needle while the donor is under general anesthesia (Copelan 2006). HSCs are continually going through a cycle of detaching from the bone marrow and entering circulation and back into the bone marrow, making it very convenient to use the peripheral blood as a source.

2. Graft Type

The graft can either be autologous, syngeneic, allogeneic, or from umbilical cord blood. Autologous HSCTs are self-donating, where a patient's marrow is taken and treated exogeneously. In that time the patient is treated with chemotherapy to reduce the tumor burden and then cells are donated back to the patient. Syngeneic transplants are when the donor is an identical twin. And allogeneic HSCTs are when a related or unrelated donor is the graft source. Umbilical cord blood

transplants are rich in HSCs but limited in volume and are less common than the other forms of transplant.

3. HLA matching

HLA genes are closely linked on chromosome 6 and are inherited as haplotypes. HLA encodes for the major histocompatibility complex (MHC). MHC are class of molecules that are found on antigen presenting cells and are important for imitating immune response. MHC has two primary classes, class I and class I. MHC class I is encoded by HLA-A, -B, and -C. MHC class II is encoded by HLA-DP, -DM, -DOA, -DOB, -DQ, and -DR. The preferred HLA match for an allogeneic donor is high resolution typed 10/10, which means the matched alleles are at HLA-A, -B, -C, -DQ, and -DP. Survival varies considerably depending on which HLA alleles are matched. A single mismatch is a significant risk factor for development of GVHD and is associated with higher mortality and decreased survival (Hamilton and Copelan 2012).

4. Pre-transplant conditioning regimens

Patients are given pre-transplant chemotherapy to reduce tumor burden of the leukemia. HSCTs are most successful when patients are in first complete remission (CR1). These can include myeloablative chemotherapy or reduce intensity therapy.

DISCOVeRY-BMT

Determining the Influence of Susceptibility COnveying Variants Related to one-Year mortality after Bone Marrow Transplant (DISCOVeRY-BMT) is a GWAS. This GWAS aims at identifying and characterizing non-human leukocyte antigen (HLA) genetic variation in the context of survival outcomes in acute lymphoblastic leukemia (ALL), acute myelogenous leukemia (AML) and myelodysplastic syndrome (MDS) patients and their matched unrelated donor (MUD) after

hematopoietic stem cell transplantation (HSCT) (L. E. Sucheston-Campbell et al. 2015). HSCT is also called bone marrow transplantation (BMT) and hence both terms will be used interchangeably throughout this document. Additionally, the terms recipients and patients will be used interchangeably.

Patient Characteristics

DISCOVeRY-BMT investigates donor and recipient genetic factors that contribute to 1 year cause-specific mortality after MUD-HSCT (Hahn et al. 2015; L. E. Sucheston-Campbell et al. 2015; Clay-Gilmour et al. 2017). This GWAS comprises two independent cohorts. All patients in DISCOVeRY-BMT were reported to the Center for International Blood and Marrow Research (CIBMTR) and had banked biorepository samples (blood samples) from recipients and donors at the National Marrow Donor Program (NMDP). Data that were reported to CIBMTR was collected from 151 transplant centers in the USA. Furthermore, all patients were de-identified and approved by the Roswell Park Cancer Institute Institutional Review Board and signed informed consent for research. Although no patients were excluded based on gender, age, or race, over 90% of DISCOVeRY-BMT cohorts were of self-reported European American ancestry. Additional exclusion criteria included: umbilical cord blood graft, *ex vivo* T-cell depleted graft or the patient underwent a prior autologous or allogeneic transplant.

Blood samples were genotyped and comprise the genotype data is the "genetic data" that is reported throughout this dissertation. In preparation for this GWAS, all post-mortem cause-specific deaths of the patients were reviewed and judged by a post-mortem by an expert panel (Hahn et al. 2015). The review and adjudicated of cause of death was conducted because many of the 151 centers the data are inconsistent in reporting and classification of outcome attributions. Thus the panel

Table 1.1: Donor and Recipients Disease Proportions by Cohort in DISCOVeRY-BMT.

	Donor (N/Percent %)		Recipient (N/Percent %)	
	Cohort 1	Cohort 2	cohort 1	cohort 2
ALL	468 (23%)	93 (12%)	483 (23%)	94 (12%)
AML	1247 (61%)	478 (63%)	1282 (61%)	488 (63%)
MDS	337 (16%)	192 (25%)	345 (16%)	195 (25%)
Total	2052	763	2110	777

Shown here are disease proportions of ALL, AML and MDS in DISCOVeRY-BMT cohorts. The percentage in each cell is computed column-wise as the proportion of sample size (N) within a cohort across disease group.

was established for concordance in cause of death, logic, and quality. Re-assessment by the panel was performed using autopsy reports and death certificates. The panel consisted of three HSCT oncologists and a HSCT clinical epidemiologist (Hahn et al. 2015).

Both cohorts comprise AML, ALL, or MDS patients who were T-cell replete and treated with myeloablative (MA) or reduced intensity (RIC) conditioning regimens prior to transplant. Cohort 1 included 2110 HSCT recipients and 2052 10/10 HLA matched unrelated donors (matched at HLA-A, -B, -C, -DRB1, and -DQB1) from 2000 to 2008 (**Table 1.1**). Cohort 2 included included 763 donors and 777 recipients from the years 2000 to 2011 (**Table 1.1**). A subset of patients (n=281) in cohort 2 were 8/8 HLA matched (matched at HLA-A, -B, -C, -DRB1), while the remaining patients (n=496) had the same matching criteria as Cohort 1 and are 10/10 HLA matched.

The proportion of patients with AML is relatively consistent between cohorts at approximately 60% of patients (**Table 1.1**). However, the proportion of ALL and MDS patients shifts between cohorts. For ALL, in cohort 1 the proportion is 23%, while in cohort 2, the proportion is 12% (**Table 1.1**). Similarly, for MDS, in

cohort 1 the proportion of patients with this disease is 16% and in cohort 2 it is 25% (**Table 1.1**).

Why do the ALL proportions go down in cohort 2

Note: Inquire about changes in MDS definition between cohorts

Survival Outcome Definitions:

The survival outcomes are cause-specific deaths during the first year post-HSCT that were reviewed and re-assessed with high confidence (described above). The primary outcomes included are: disease related mortality (DRM), and transplant related mortality (TRM) (**Table 1.2**). The secondary outcomes were: overall survival (OS), progression-free survival (PFS), and relapse (REL) (**Table 1.2**).

TRM subtypes were further stratified into graft-versus-host disease (GVHD), organ failure (OF), infection (INF), or other. Other causes of death included more rare events, such as (but not limited to) secondary malignancies, primary or secondary graft failure, or hemorrhage. Despite being adjudicated, TRM subtypes are difficult to differentiate from one another, and some times more than one may present in the autopsy or report. The boundaries aren't necessarily clear so there may be overlap between these patients. Nevertheless, GVHD deaths included acute or chronic GVHD, conditional upon a patient actively receiving treatment for GVHD at the time of death. Deaths that were considered to be infection were identified as arising from bacterial, fungal, viral, and/or protozoan organisms that caused organ damage. OF deaths were defined as organ toxicity relating to the transplant and not due to disease progression, GVHD, or infection.

The phenotypes that are tested in DISCOVeRY-BMT are stratified into different disease groups:

1. Mixed Disease – which includes AML, MDS, and ALL (the full cohort).

Table 1.2: Definitions of Survival OUtcomes

Outcomes	Definitions
Primary Outcomes	
Disease Related Mortality (DRM)	broadly defined as deaths relating to leukemia/MDS relapse/progression, including death attributed to toxicity or infection from anti-leukemic treatments post-HSCT.
Transplant Related Mortality (TRM)	defined as any cause of death except the underlying disease, pre-existing disease, accidental death or suicide, or death unrelated to the transplant.
Secondary Outcomes	
Overall Survival (OS)	defined as any patient (recipient) that died at any point within the first 12 month window post-HSCT of this observational study.
Progression Free Survival (PFS)	is defined as the time to relapse. All patients were analyzed as time to progression of disease
Relapse (REL)	patients who were not in CR pre-HSCT and the disease returns (relapse) after HSCT.

2. AML + MDS – the myeloid malignancies being grouped together, excluded the lymphoid lineage (ALL)

3. AML only – includes only AML

4. ALL only – includes only ALL

5. MDS only – includes only MDS. This will not be tested or discussed in much detail.

In agreement with published CIBMTR statistics, about 40% of patients die after 1 year in both DISCOVeRY-BMT cohorts (**Table 1.3**, **Figure 1.1A**). Patients dying from transplant related causes is about 22% between both DISCOVeRY-BMT

cohorts (**Table 1.3**). Similarly, about 18-19% of patients die of their disease within the first year after transplant (**Table 1.3**). Dying due to disease is the leading cause of death in AML + MDS and AML only as well. Conversely, for ALL, transplant related mortality is a larger contributor to overall death than the other subgroups. Progression free survival is about 50% for the full cohort and disease subsets. AML alone has the most relapse compared to all other groups. Interestingly, when looking at the survival curves it seems like most of the TRM events happen early and then DRM supersedes TRM as the year progresses (after about 6 months) for all diseases and myeloid subtypes (**Figure 1.1A**). The ALL only curve shows disease contributing to death early on and that if ALL patients die after 6 months, DRM begins to equilibrate with TRM (**Figure 1.1A**).

Additional information that is included from the CIBMTR were recipient/donor age, recipient/donor sex, recipient BMI, graft source, Karnofsky performance score, disease status (early, intermediate, advanced), year of HSCT, and conditioning regimens given prior to transplant. These meta-data will be incorporated in the statistical models that will be discussed in the next subsection as well as other chapters.

Genotyping and Quality Control

All samples were genotyped on a Illumina Human OmniExpress BeadChip whole-genome genotyping microarray. This chip had 637,655 tagged SNPs available that were strategically selected from all three phases of the HapMap project to capture the greatest amount of common SNP variation (>5% MAF).

Samples were assigned to plates to ensure the even distribution of patient characteristics and potential confounding variables using Optimal Sample Assignment Tool (OSAT), an R/Bioconductor software package (Yan et al. 2012). Over

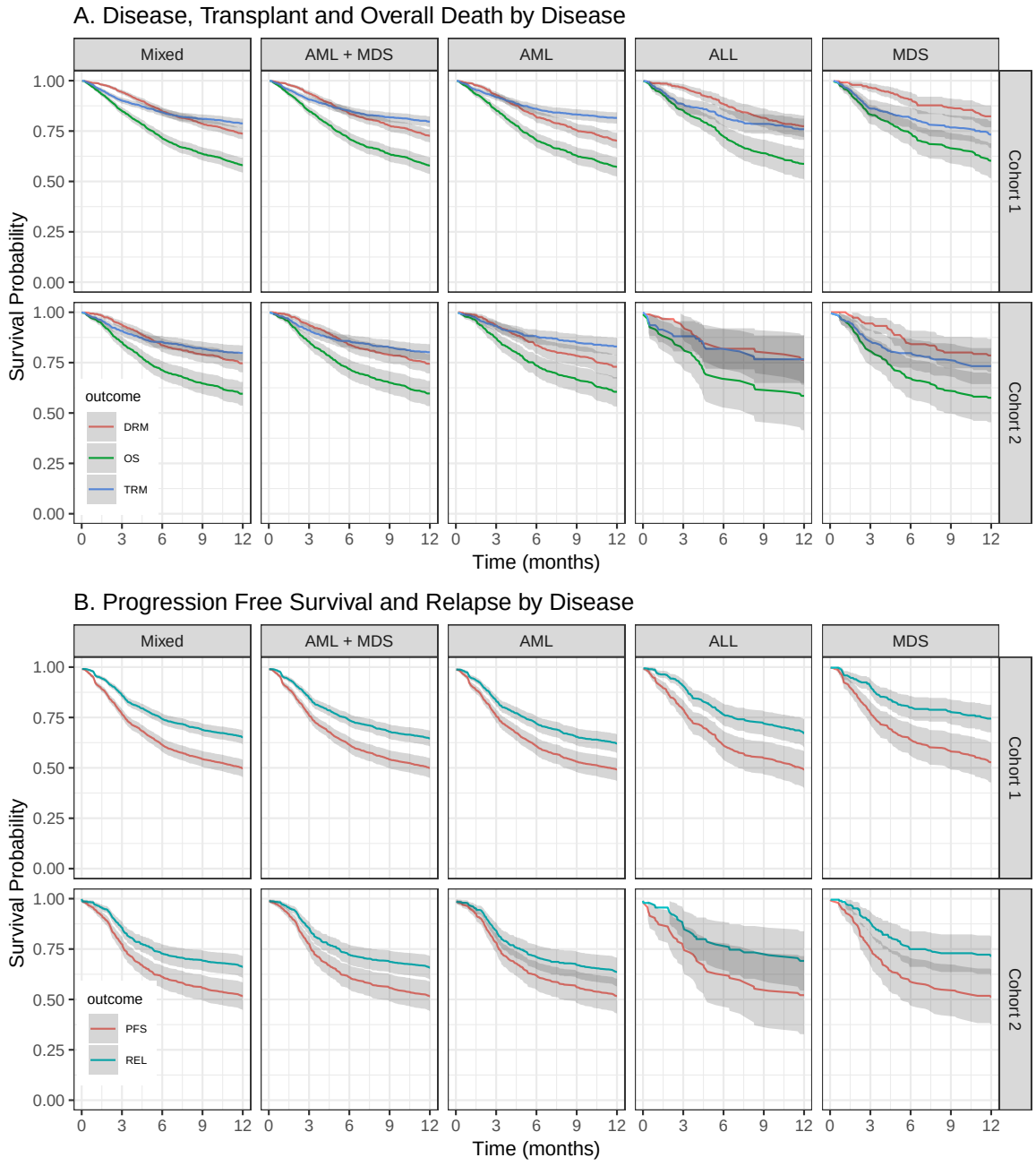

Figure 1.1: Survival curves for all disease groups being tested (Mixed disease, AML+MDS, AML only, and ALL only). The x-axis is survival probability. The y-axis is time in months. Cohorts 1 and 2 are shown in the left and right panel respectively. The gray shaded areas are 95 percent confidence intervals. (A) Red is disease-related mortality (DRM), green is overall survival (OS), and blue is transplant related mortality (TRM). (B) Cyan is relapse and red is progression free survival (PFS)

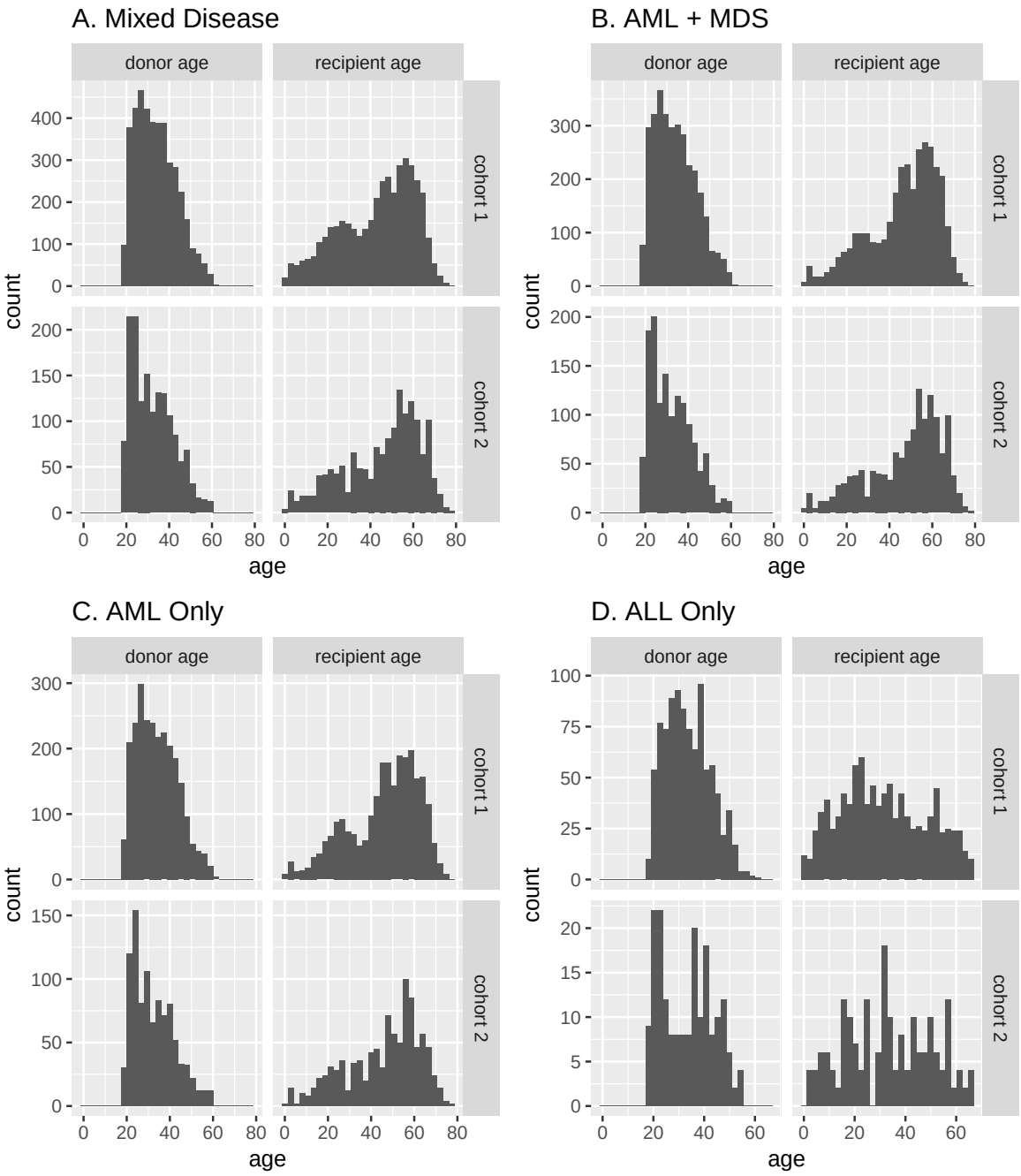

Figure 1.2: Histogram of Donor and Recipient Age Distributions in (A) Mixed Disease (AML, ALL, and MDS), (B) AML + MDS, (C) AML only, and (D) ALL only.

90% of DISCOVeRY-BMT patients self-reported as European American, Caucasian or White and thus replication and validation analyses are performed on these recipient-donor pairs.

An important concept in statistical genetics is *population stratification* and the necessity to adjust for it when doing association studies. The association could be due to the underlying structure of the population and not a disease associated locus. QC is done to control from those using one of several different available software. Stringent quality control was performed on both samples and SNPs within this population. Population outliers were removed using EIGENSTRAT (A. L. Price et al. 2006) (n=73). Additional sample quality control removed samples with missing call rate 2% (n=54), sex mismatch (n=9), abnormal inbreeding coefficients (n=20), and evidence of cryptic relatedness (n=17), yielding 2107 and 777 donor-recipient pairs in cohorts 1 and 2, respectively. Typed SNPs were removed if the call rate was <98%, there was deviation from Hardy-Weinberg equilibrium proportions (C. C. Laurie et al. 2010) or discordance between duplicate samples was >2%.

Originally, DISCOVeRY-BMT was imputed using IMPUTE2 and 1000 Genomes Phase 3 data. Again, as the most of the population are of European ancestry, when HRC was released, DISCOVeRY-BMT was reimputed using HRC to have higher quality imputation specific to this population.

Statistical Analysis

Cox Proportional Hazards Model

Survival models examine the time it takes for events to occur. Specifically, survival models examine the relationship between survival (time that passes before

some event occurs) and one or more *covariates* (predictors) that may be associated with that quantity of time. The Cox proportional hazards regression model (Cox 1972) is used for survival analysis and is our main statistical model of choice.

need to explain dosage need to explain we test every SNP individually

Assumptions of the Cox model:

1. the regression coefficient (β) is constant over time (proportional hazards assumption)
2. linear combination of covariates
3. link function is exponential

Mathematical concepts and notations

The Cox model is as follows:

Consider a population of subjects, i, we observe either time to event or censoring. For the censored individuals, we know that time to event is greater than censoring time. So the survival function is $S(t)$. Let T represent survival time. T is a random variable:

Cumulative distribution function (CDF):

$$P(t) = Pr(T \leq t) \tag{1.1}$$

Probability density function (PDF):

$$p(t) = \frac{dP(t)}{dt} \tag{1.2}$$

The survival function $S(t)$ is the complement of the CDF:

$$S(t) = Pr(T \geq t) = 1 - P(t) \tag{1.3}$$

and the hazard function is $\lambda(t)$ (or age specific failure rate). The hazard function, $\lambda(t)$, is the distribution of survival times, which assesses the instantaneous risk of dying at time t, conditional on survival to that time:

$$\begin{aligned} \lambda(t) &= \lim_{\Delta t \to 0} \frac{Pr\Big[t \leq T < t + \Delta t) | t \geq T\Big]}{\Delta t} \\ &= \frac{f(t)}{S(t)} \end{aligned} \tag{1.4}$$

Let $X_i = X_{i1}, ..., X_{ip}$ be realized values of covariates for subject i. The hazard function for the Cox model has the form:

$$\begin{aligned} \lambda(t|X_i) &= \lambda_0(t) \cdot \exp(\beta_1 X_{i1} + ... + \beta_p X_{ip}) \\ &= \lambda_0(t) \cdot \exp(X_i^T \cdot \beta) \end{aligned} \tag{1.5}$$

This expression gives us the hazard function at time t for subject i with covariate vector X_i. The baseline hazard is a nuisance parameter and is completely removed. For simplicity, we assume that there are no tied failure times, although there are methods for modifying the partial likelihood in the case of ties (Breslow 1974, Efron (1977)).

The probability of the event to be observed occurring with subject i at time

Y_i:

$$
\begin{aligned}
L_i(\beta) &= \frac{\lambda(Y_i|X_i)}{\sum_{j:Y_j \geq Y_i} \lambda(Y_i|X_j)} \\
&= \frac{\lambda_0(Y_i)\exp(X_i^T \cdot \beta)}{\sum_{j:Y_j \geq Y_i} \lambda_0(Y_i)\exp(X_j^T \cdot \beta)} \\
&= \frac{\exp(X_i^T \cdot \beta)}{\sum_{j:Y_j \geq Y_i} \exp(X_j^T \cdot \beta)}
\end{aligned}
\tag{1.6}
$$

where Li is between 0-1. This is in fact the partial likelihood function. This is useful to estimate the beta coefficients without having to model a hazard function that is dependent on time. ** I will finish out writing the likelihood function and maximizing the likelihood using Newton-raphson the hessian of the PL and how we use it to estimate standard errors … this is important for gwasurvivr **

** Don't forget to mention hazard ratio computation ** Hazard ratio is the ratio of hazard rates described by

Model Diagnostics

The Cox model can be evaluated in two ways. The proportional hazards assumption can be tested using Schoenfeld residuals graphically or using a goodness-of-fit test (Schoenfeld 1982). The model itself can be validated by simulation. Here will we show both methods.

Schoenfeld residuals are based on the effects of the predictor variables that are assumed to be independent of time, plotting these residuals versus time is done to visually assess the effect of the predictor variable and its relationship with time. A smooth line is fit to the plot of the residuals (Grambsch and Therneau 1994). If the smoothed line has a slope and intercept of approximately 0, then the proportional hazards assumption has been met (Grambsch and Therneau 1994).

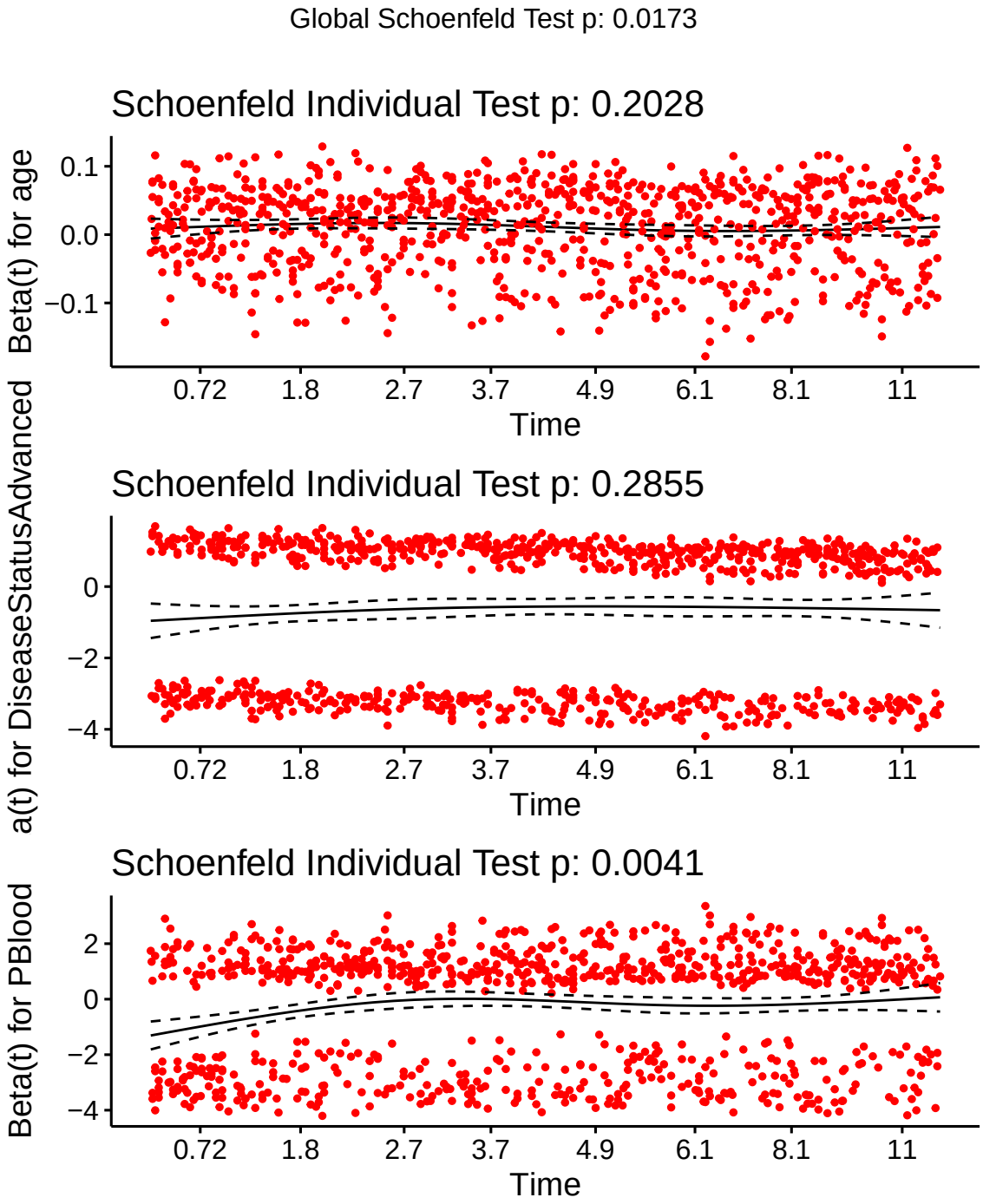

Figure 1.3: Schoenfeld Residuals for Overall Survival (OS) adjusted for age, disease status and graft source. The Global Schoenfeld Test is a two-sided chi-square test. Each individual Schoenfeld test is a per-variable chi square test. $P < 0.05$ is statistically significant.

The simulation plots are sampled directly from the survival function $S(t)$. Random variables, u, can be generated numerically using pseudocontinuous counts. The cumulative hazard function $H(t)$ can be computed from the hazard function (equation 1.4) numerically (**Figure 1.4**, left panel) as well as the survival function (**Figure 1.4**, right panel).

$$\begin{aligned} H(t) &= \int_0^t \lambda(u)du \\ &= \int_0^t \frac{f(u)}{S(u)}du \\ &= \int_0^t \frac{d[1-S(u)]}{S(u)} \\ &= -\log\left(S(t)\right) \\ S(t) &= \exp\left(-H(t)\right) \end{aligned} \tag{1.7}$$

And since $f(t) = h(t) \cdot S(t) = \lambda(t) \cdot \exp(-H)$, the effect of covariates can be computed by generating survival curves for each individual i in the sample by multiplying the $H(t)$ by the exponentiated linear predictor, such that $f(t) = \exp\left(-H(t) \cdot \exp\left(X_i \cdot \beta\right)\right)$.

The curves in **Figure 1.5** are predicted from the survival model. The distributions of the simulated survival times that were computed numerically resemble the same pattern as the observed data for overall survival in cohort 1 (**Figure 1.6**). These simulations were performed for cohort 2 and the other outcomes as well (see Appendix). The simulations were able to replicate the pattern of the observed data (see Appendix).

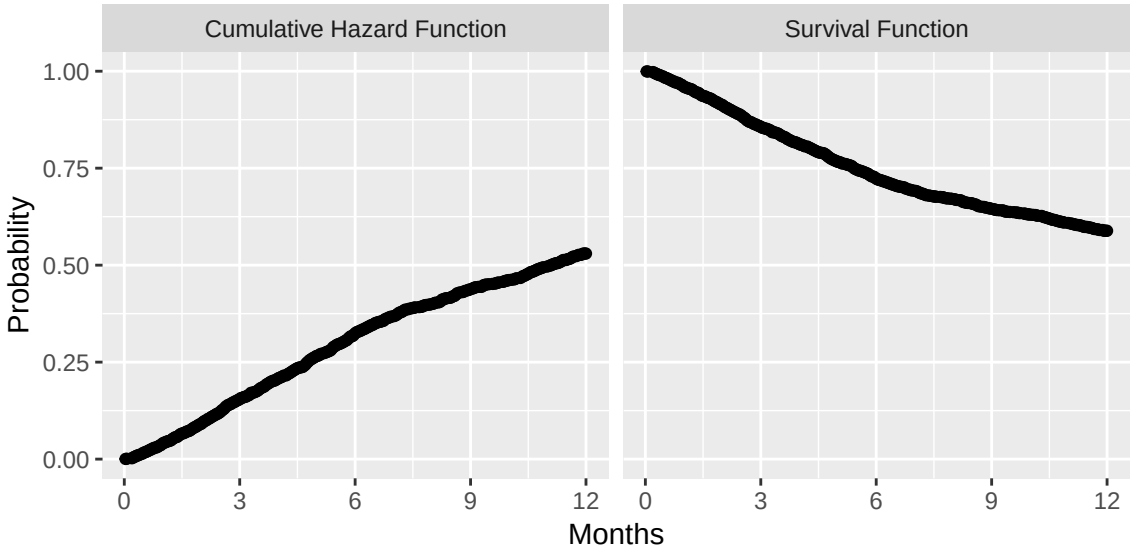

Figure 1.4: Model Diagnostics. Predicted Survival Probabilities

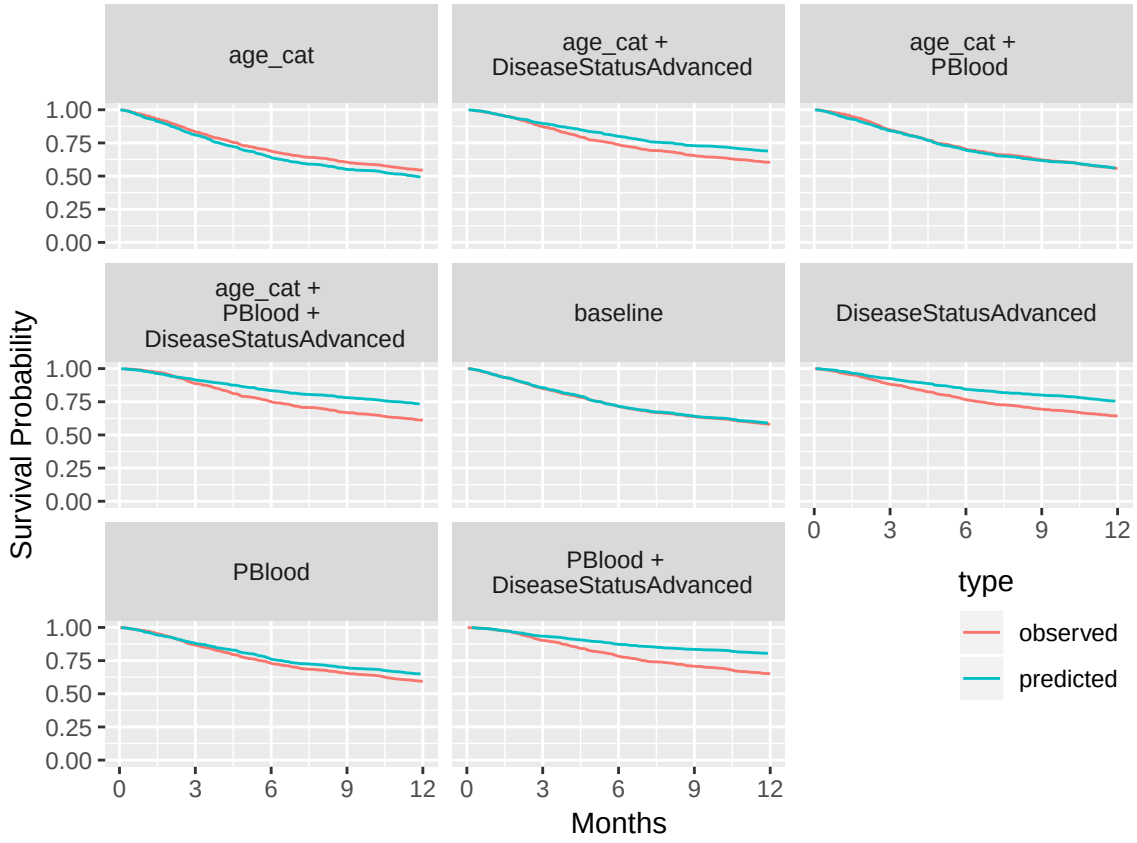

Figure 1.5: Model Diagnostics. Predicted Survival Curves

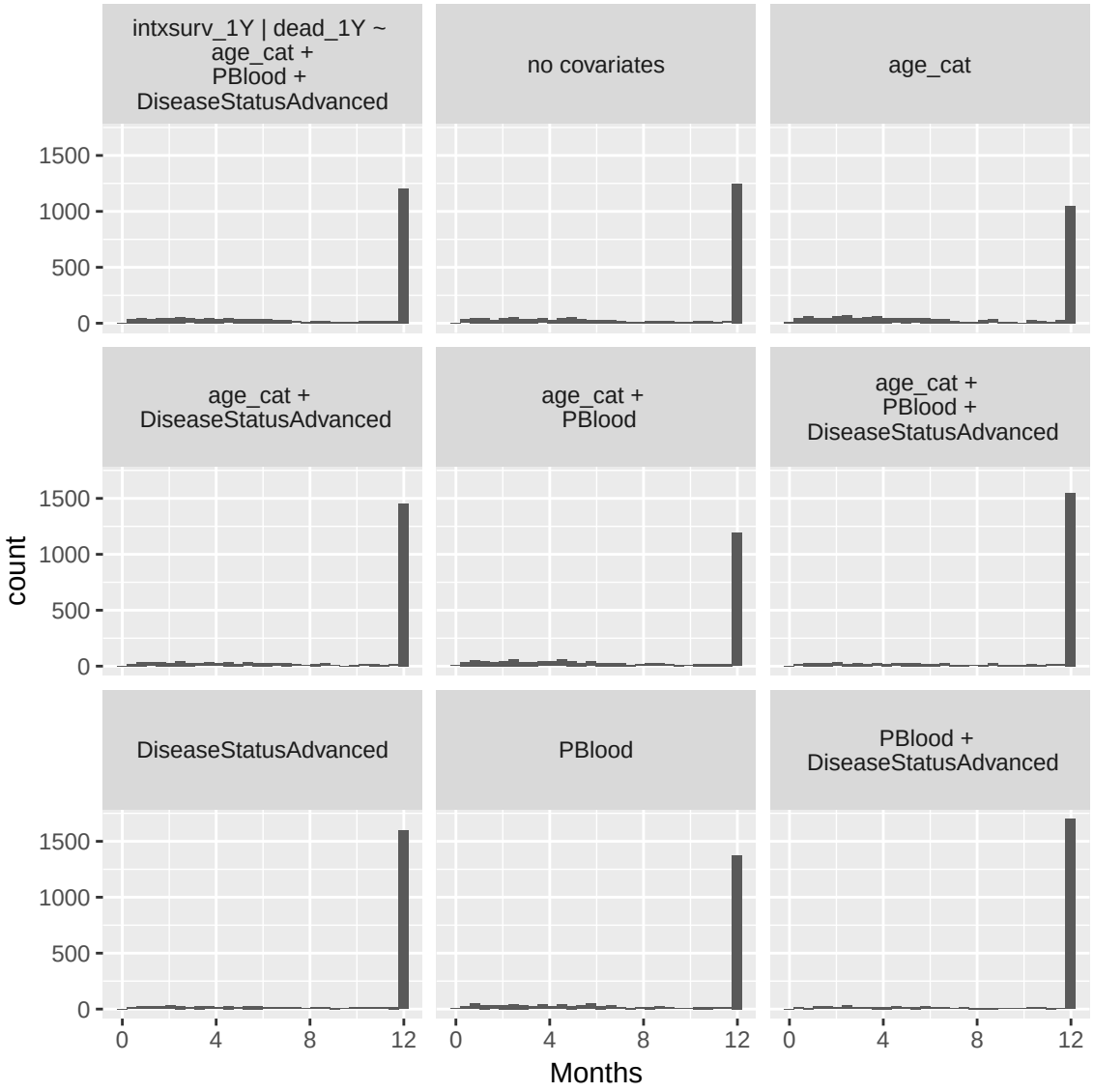

Figure 1.6: Model Diagnostics. Histogram of observed and predicted survival times

Power Calculations

We conducted meta-analyses to combine the effects of both cohorts (discussed in next section below), as such, power calculations were done considering the sample size of both cohorts combined. Minimum detectable hazard ratios of recipient and/or donor depend on three variables: 1.) proportion of individuals experiencing an event, 2.) frequency of a causal variant, and 3.) the quality of genotyping SNPs that capture the genetic variation underlying the hazard of an event. Events are measured at 1 year and at the most updated observation time (most recent phenotype data available is May 5th, 2017) post-HSCT. The events that will be measured are death due to transplant (TRM) and specific causes of death (organ failure, infection, GVHD) attributable to TRM. OS is a function of TRM and thus we will present minimum detectable hazard ratios for OS. The proportion of events (Figure 3) ranges from infrequent events to (i.e. TRM subtypes) to frequent events (i.e. OS). We will assume lower and upper bound causal variant frequency between 5% and 40%, respectively. We assumed SNPs selected for genotyping capture 85% of the variation across each gene; thus, all power calculations are corrected by setting our effective total sample size equal to 0.85 x 3532 = ~3000. We present the range of hazard ratios detectable for varying proportions of events and allele frequencies in a univariate model assuming 80% power to detect genome-wide significance at $5\text{x}10^{-8}$. With 3,532 recipient-donor pairs, the minimum detectable hazard ratio under these assumptions is identical for recipient genotype, donor genotype, and the mismatch between donor and recipients. Given the minimum proportion of events experienced in TRM subtypes and overall TRM are between 0.10 and 0.30 with a common allele (MAF=0.40), we have power to detect hazard ratios between 1.69 and 1.35, respectively. Under these same proportion of events, with more rare

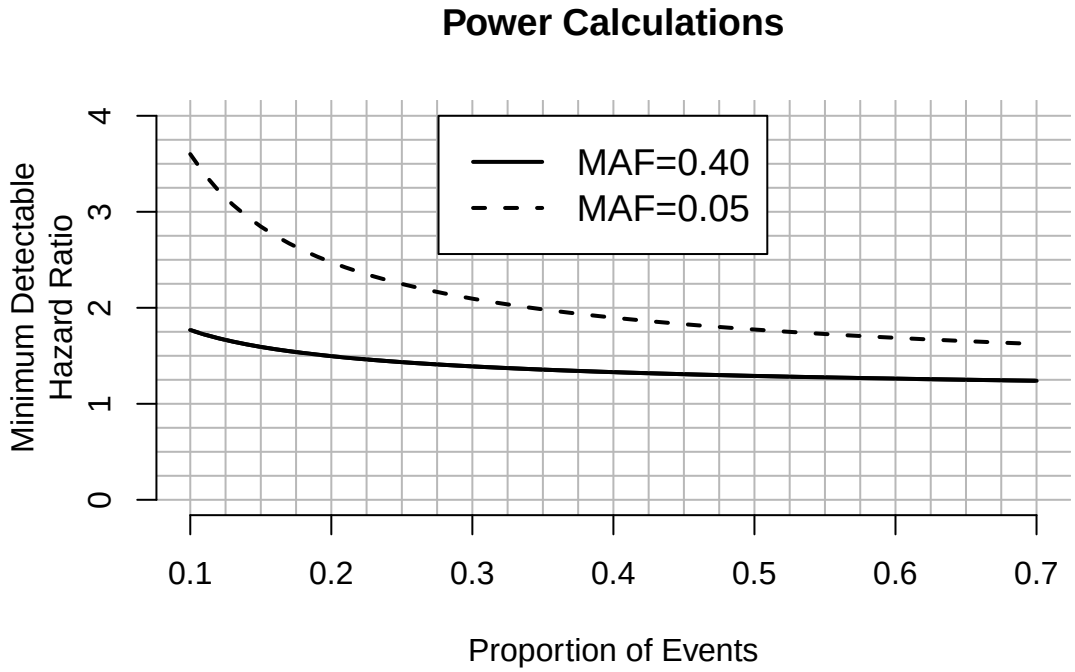

Figure 1.7: Hazard ratios associated with survival models.

variants (MAF=0.05), we have the power to detect hazard ratios between 3.3 and 2.0, respectively. For OS models, assuming the overall rate of death is 0.50, we can detect SNPs in with hazard ratios between 1.26 (MAF=0.40) and 1.7 (MAF=0.05). Lower bound is based off each TRM subtype being at least 0.10 (10%) of all patients.

Table 1.3: Proportion of Events by Survival Outcome

	Recipient (N/Percent %)	
Outcome	Cohort 1	Cohort 2
Mixed Disease		
Disease related mortality (DRM)	474 (22.46%)	168 (21.62%)
Transplant related mortality (TRM)	405 (19.19%)	141 (18.15%)
Overall survival (OS)	879 (41.66%)	309 (39.77%)
Relapse (REL)	639 (30.28%)	233 (29.99%)
Progression free survival (PFS)	1055 (50%)	374 (48.13%)
Graft-versus-host disease (GVHD)	134 (6.35%)	59 (7.59%)
Infection (INF)	122 (5.78%)	36 (4.63%)
Organ failure (OF)	104 (4.93%)	26 (3.35%)
AML + MDS		
Disease related mortality (DRM)	383 (23.54%)	149 (21.82%)
Transplant related mortality (TRM)	297 (18.25%)	121 (17.72%)
Overall survival (OS)	680 (41.79%)	270 (39.53%)
Relapse (REL)	506 (31.1%)	208 (30.45%)
Progression free survival (PFS)	810 (49.78%)	329 (48.17%)
Graft-versus-host disease (GVHD)	97 (5.96%)	55 (8.05%)
Infection (INF)	90 (5.53%)	27 (3.95%)
Organ failure (OF)	76 (4.67%)	21 (3.07%)
AML Only		
Disease related mortality (DRM)	333 (25.98%)	115 (23.57%)
Transplant related mortality (TRM)	211 (16.46%)	73 (14.96%)
Overall survival (OS)	544 (42.43%)	188 (38.52%)
Relapse (REL)	431 (33.62%)	161 (32.99%)
Progression free survival (PFS)	648 (50.55%)	234 (47.95%)
Graft-versus-host disease (GVHD)	61 (4.76%)	26 (5.33%)
Infection (INF)	69 (5.38%)	14 (2.87%)
Organ failure (OF)	59 (4.6%)	18 (3.69%)
ALL Only		
Disease related mortality (DRM)	91 (18.84%)	19 (20.21%)
Transplant related mortality (TRM)	108 (22.36%)	20 (21.28%)
Overall survival (OS)	199 (41.2%)	39 (41.49%)
Relapse (REL)	133 (27.54%)	25 (26.6%)
Progression free survival (PFS)	245 (50.72%)	45 (47.87%)
Graft-versus-host disease (GVHD)	37 (7.66%)	4 (4.26%)
Infection (INF)	32 (6.63%)	9 (9.57%)
Organ failure (OF)	28 (5.8%)	5 (5.32%)

CHAPTER 2: Replication and Validation of Previous HSCT Literature

Introduction

Genetic associations studies usually fall under two main subcategories, candidate gene association studies (CGAS) or genome wide association studies (GWAS). CGAS are association studies that investigate specific genes or regions of interest. Typically these are performed when researchers believe that the underlying biology is understood and they want to identify specific markers that contribute to genetic variation in these 'known' regions. For over a decade, researchers in the hematology and hematological transplant field have conducted CGAS that investigated the relationship between non-HLA genetics and survival outcomes after allogeneic transplant. The rationale for conducting the CGAS was to increase knowledge about clinical management or to serve as a potential target for novel therapeutics. We exhaustively searched PubMed for CGAS where the phenotype of interest was survival outcomes (DRM, TRM, OS, PFS) in patients with ALL, AML, or MDS after HLA-matched -related-donor (MRD) or -unrelated-donor (MUD) HSCT (Karaesmen et al. 2017). We identified 70 studies that reported 45 SNPs in 36 genes as significantly associated with survival outcomes after transplant. DISCOVeRY-BMT was used to replicate or validate these published studies (Karaesmen et al. 2017). The majority of these studies tested for associations in small datasets, ranging from a few dozen to a few hundred patients and donors, included heterogeneous diseases spanning benign to malignant hematological diseases, related and/or unrelated donors with various degrees of HLA-matching and patients treated across multiple decades, from the 1980s through early 2000s. In addition to reproducing previous

findings, we were interested in agnostically evaluating whether genes that had been previously reported upon had an aggregate effect that could be detected and contributed to survival after transplant.

Methods

Literature Review

An extensive literature search of PubMed was performed using to identify peer-reviewed scientific studies (published on or before December 30, 2016) that reported non-HLA genetic polymorphisms associated with survival outcomes after allogeneic BMT, including disease-related mortality (DRM), progression-free survival (PFS), transplant-related mortality (TRM) and/or overall survival (OS) (Karaesmen et al. 2017). The PubMed search terms, filtering approach are described below:

```
(SNP[Text Word] OR ("polymorphism, genetic"[MeSH Terms] OR
("polymorphism"[All Fields] AND "genetic"[All Fields]) OR
"genetic polymorphism"[All Fields] OR
"polymorphism" [All Fields])) AND
(allo-HSCT[All Fields] OR allo-HCT[All Fields] OR
("unrelated"[All Fields] AND ("donor"[All Fields] OR
"donors"[All Fields]) AND ("transplant"[All Fields] OR
"transplantation"[All Fields])) OR ("allogeneic"[All Fields] AND
("transplant"[All Fields] OR "transplantation"[All Fields])) OR
("hematopoietic"[All Fields] AND ("transplant"[All Fields] OR
"transplantation"[All Fields]))) AND (("mortality"[Subheading] OR
"mortality"[All Fields] OR "mortality"[MeSH Terms]) OR
("mortality"[Subheading] OR "mortality"[All Fields] OR
"survival"[All Fields] OR "survival"[MeSH Terms]) OR
(non[All Fields] AND ("recurrence"[MeSH Terms] OR
"recurrence"[All Fields] OR "relapse"[All Fields])) OR
non-relapse[All Fields]) AND English[Language]
```

The Inclusion Criteria comprised of:

Inclusion criteria:

1. non-HLA genes
2. survival after BMT as phenotype

Excluded:

1. Non-English papers
2. Working group studies
3. Reviews
4. SNPs not in build hg19
5. Haplotypes
6. Chronic Lymphocytic Leukemia (CML) or multiple myeloma (MM) or lymphoma only papers
7. Autosomal only
8. Microsatellites, CNVs, VNTRs, or other variation markers

Defintions of Replication and Validation

In principle, results from CGAS or GWAS should be reproduced in an independent study to confirm findings (Colhoun, McKeigue, and Smith 2003; Martin et al. 2016). Two distinctive terms have gained popularity amongst researchers to describe reproducibility – specifying if there are differences between the original population that was studied and the confirmation study: replication and validation (Igl, Konig, and Ziegler 2009). Replication is deemed to be when the inclusion criteria are near or completely identical (i.e. same ancestral population), so that any differences between the samples in the study can attributed to random variation (Igl, Konig, and Ziegler 2009). Validation reproducibility is when the original and confirmation study have similar but slightly different inclusion criteria (i.e. different ancestral populations. In the validation case, the underlying differences between

the original and confirmation study can be attributed to systematic variation (Igl, Konig, and Ziegler 2009).

Thus, replication analyses were conducted when the original study included HLA MUD-HSCT in patients of European ancestry. Validation analyses were performed on studies of leukemia patients of non-European ancestry, patient populations who received MRD-HSCT, or patient populations that were mixed between those who received a MRD-HSCT and MUD-HSCT (Karaesmen et al. 2017). For studies of outcomes involving multiple hematologic malignancies, the entire DISCOVeRY-BMT study population was analyzed. If the original study population was specified as AML, ALL and/or MDS, the same disease inclusion criteria were applied so that the replication/validation study population aligned with that of the original study population.

Genotyping data

All samples were genotyped using the Illumina Human OmniExpress BeadChip and the Illumina HumanExome BeadChip (University of Southern California Genomics Facility). In total, 637,655 and 632,823 SNPs from the OmniExpress BeadChip were available for imputation in cohorts 1 and 2, respectively, using 1000 Genomes Project phase 3. The missing genotypes were imputed using IMPUTE2 software (B. N. Howie, Donnelly, and Marchini 2009). QCTOOL was used to remove imputed genotypes with an info score > 0.7, certainty > 0.7, and a minor allele frequency ≥ 0.005. To test the joint effect of recipient and donor genetic variation, the recipient-donor (R-D) *mismatch* genome computed by taking the absolute value of the difference of minor alleles between recipient and donor at each SNP. For example, at a given SNP where the recipient is homozygous minor (2 minor alleles) and the donor is heterozygous (1 minor allele), the mismatch allele dosage

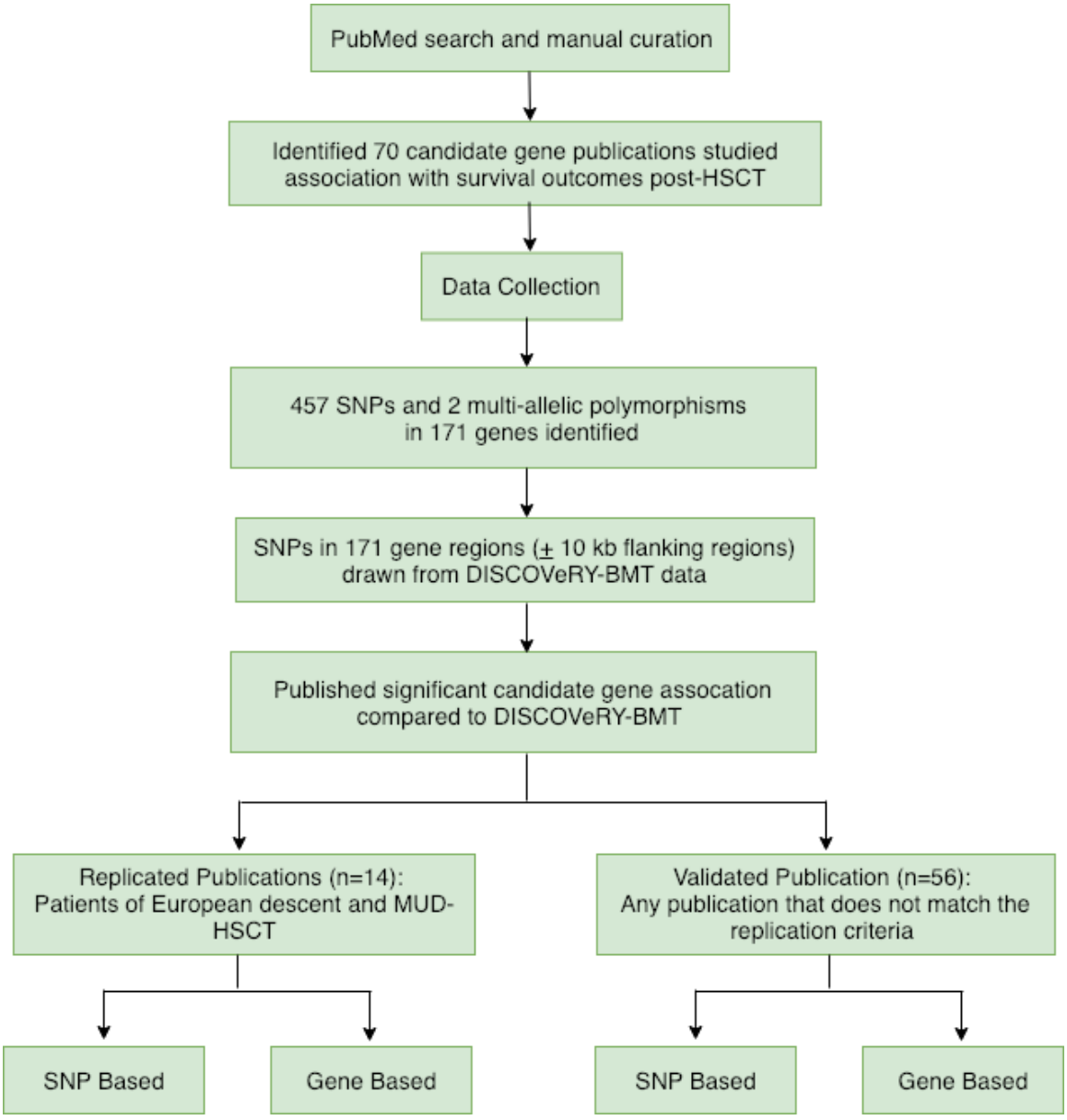

Figure 2.1: Pipeline performed for reproducing previous candidate gene association study literature

would be $|2-1|=1$. Rs2066847 (SNP13) in *NOD2/CARD15* was the only variant analyzed from the Illumina HumanExome, as it was not typed on the OmniExpress chip or available following imputation.

Genetic Models

Prior to genetic analyses, clinical covariates for inclusion in genome-wide survival models were selected using bidirectional stepwise regression (Venables and Ripley 2002) on Cox proportional hazard models (Cox 1972) of OS, PFS, TRM and DRM using R statistical software (R Core Team 2018). Cox proportional hazard models of OS, TRM and DRM evaluated SNPs associated with time to death with all survivors censored at 1 year post-HSCT (Therneau and Grambsch 2013). PFS was defined as the time to disease progression or death. The covariates that were included in the model were (see appendix for R code):

1. OS – recipient age, disease status (early or advanced), graft source (bone marrow or peripheral blood)

2. DRM – recipient age, disease status (early or advanced)

3. TRM – recipient age, body mass index (BMI) (underweight, normal, overweight, obese), graft source (bone marrow or peripheral blood)

4. PFS – recipient age, disease status (early or advanced)

Depending on the disease group, disease types were adjusted for by creating an indicator variables. E.g. For analyses without ALL (AML and MDS), the indicator variable AML dummy and MDS dummy were added to the model. Competing risk models (Fine and Gray 1999) were used for TRM and DRM (because a patient can only die once). Dosage data accounting for the probability of each genotype were used in all analyses of imputed data. Effect size estimates and standard errors from DISCOVeRY-BMT Cohorts 1 and 2 were compared and combined using

a fixed-effects inverse variance meta-analyses in METAL (Willer, Li, and Abecasis 2010). For each SNP, heterogeneity of effect size estimates between cohorts 1 and 2 was assessed using p-values from significance tests of heterogeneity (P_{het}) and I^2. Variants with $P_{het} < 0.05$ and $I^2 > 50$ were meta-analyzed with a random effects models using meta (Schwarzer 2007) in R.

Multi-allelic models

NOD2/CARD15 was the most common gene studied in the HSCT literature. The associations were based on a 3-variant R-D pair model (rs2066844 [SNP8], rs20666845 [SNP12], rs2066847 [SNP13]) and single SNP associations with SNP13 (Ernst Holler et al. 2004). The null type was when the R-D pair are homoyzgous common allele for all 3 SNPs and the effect allele combination was considered the presence of $/geq$ 1 minor allele in any of the 3 SNPs within the R-D pair. *CCR5* haplotype was studied in one replication study (McDermott et al. 2010). The risk H1/H1 haplotype was defined as presence of homozgous genotype for the major allele at rs1799987 (AA), rs1800023 (AA), rs333 (ACAGTCAGTATCAATTCTG-GAAGAATTTCCAG); individuals not homozygous common were considered null. rs333, a 32-basepair deletion, was not typed or imputed in the DISCOVeRY-BMT cohort, however we selected a proxy SNP (rs1133418) in strong linkage disequilibrium ($r^2 = 0.97$) with rs333. The presence of the G allele in the proxy SNP corresponds to ACAGTCAGTATCAATTCTGGAAGAATTTCCAG in rs333. McDermott and colleagues (2010) defined 3 risk subgroups:

1. R-D Group 1 – R-D pairs lacked *CCR5* H1/H1 haplotype
2. R-D Group 2 – donors only had *CCR* H1/H1 haplotype
3. R-D Group 3 – recipients only had *CCR5* H1/H1 haplotype

The imputed data for *CCR5* was in the IMPUTE2 output data chunk for chromo-

some 3 (`BMT093013_forImpute.chr3-45000000-50000000.impute2`). QCTOOL was used to extract the data out from the IMPUTE2 and output it into the a VCF file. The location for these files was stored on the University at Buffalo Computational Center Resource (UB CCR) supercomputer. See Appendix for full code on analysis.

```
qctool \
    -g ./BMT093013_forImpute.chr3-45000000-50000000.impute2 \
    -s ./BMT093013_forImpute.chr16-50000000-55000000.impute2_samples \
    -og ./ccr5_rep_dosages_threshold.vcf \
    -incl-rsids ccr5_snps.txt
```

The data wrangling to prepare the data for these models was done in R. The Cox survival models were written and automated with custom R code that leveraged the survival package (Therneau and Grambsch 2013) (see Appendix for code used for the full analysis).

Gene-based assocation testing

VErsatile Gene-based Association Study 2 (VEGAS2) software was used for gene-based association testing (Mishra and Macgregor 2015). VEGAS2 uses 10^6 Monte Carlo simulations to test the global significance of an association for sets of SNPs in defined genomic regions. VEGAS2 reports a gene-based P-value for each gene determined using individual SNP association P-values. Directional effects are not incorporated into analyses; thus, all SNPs can be aggregated without dampening an association signal. For the gene-based replication or validation analyses, the P-values from typed and imputed SNPs in DISCOVeRY-BMT ($\pm$ a 10kb flanking region) meta-analyses of OS, PFS, TRM and DRM were used as input into the VEGAS2 software. Gene-based P-values were calculated for donor, recipient, and R-D mismatch analyses of the full cohort (ALL, AML and MDS patients) or homo-

geneous disease subgroups (ALL or AML or MDS patients) corresponding to the analyses performed in the original studies (Karaesmen et al. 2017).

To run VEGAS2, a flat text file is needed that has two unlabeled columns (rsid and GWAS P-value [P_{meta}]).

For example:

```
#snps.txt
rs6696752   0.827182998293298
rs72638700  0.874653327370856
```

And then a simple command prompting VEGAS2 is run on the command line:

```
vegas2 \
    snps.txt \
    -pop 1000GEURO \
    -subpop EURO \
    -genesize 10kbloc \
    -top 100 \
    -sex BothMnF \
    -max 1000000 \
    -out ./results/output_V2out
```

VEGAS2 analyses were using SNPs from all of the identified genes and p-values from DISCOVeRY-BMT (for all outcomes and 3 genomes) on UB CCR (see Appendix).

Functional Annotation

To gain a deeper understanding of genetic associations, often bioinformatics approaches leverage the plethora of curated, publically available datasets and databases (Gallagher and Chen-Plotkin 2018). The databases used in this study were RegulomeDB (Boyle et al. 2012), Blood expression quantitative trait loci (eQTL) Browser (Westra et al. 2013), and Variant Effect Predictor (VEP) (McLaren et al. 2010), which were used to annotate the SNPs in the candidate

genes. For RegulomeDB and Blood eQTL Browser, the full database consisting of raw data scores, P-values, and annotations were downloaded. VEP annotation was done directly on VEP's website.

RegulomeDB stores variant annotation with known and predicted regulatory DNA elements in humans. The elements include DNase hypersensitivity, transcription factor (TF) binding sites, and promoter regions that have been characterized to alter transcription (Boyle et al. 2012). RegulomeDB database compiled these annotations using the publically available data sets from the Encyclopedia of DNA elements (ENCODE) project and the Roadmap Epigenome Consortium and Gene Expression Omnibus (GEO). RegulomeDB scores are the following: 1a-1f likely to affect TF binding and linked to altering expression of target genes; 2a-2c likely to affect TF binding; 3a-3b less likely to affect TF binding, and > 3 has minimal binding evidence. RegulomeDB DB scores are assigned (only one per SNP) based on the level of effect and evidence of functional modification that is attributable to the SNP across multiple cell lines from differing tissues. The scores are supported by experimental evidence – with scores from 1 to 7, with 1 having the greatest functional effect, and scores of 7 show no evidence of having modifying effects.

Blood eQTL database was built from a study (N=5000) that investigated the correlation between genetic expression and genetic polymorphisms and replicated the results in an independent study ($N \approx 3000$), Westra et al. (2013)]. We only considered *cis*-eQTLs, defined as < 250KB distance between the SNP chromosomal position and the probe midpoint for gene expression. Furthermore, VEP was used to determine the hypothetical (predicted) functional importance of missense and nonsense variants based on SIFT, Mutation Taster and PolyPhen-2 software.

Results

Candidate Gene Studies of Survival Outcomes

The literature search identified 70 publications that studied a total 458 SNPs and 2 multi-allelic polymorphisms in 171 genes (**Figure 2.1**). Studies included patients who received a transplant from an MUD-HSCT (19 articles), a MRD-HSCT (23 articles), or both (28 articles) (**Table 2.1**). Study populations included patients and donors of European ancestry (53 articles), Asian ancestry (15 articles), or mixed genomic ancestry (2 articles) (**Table 2.1**).

A total of 14 articles assessed genetic variation in HLA MUD-HSCT patients of European ancestry, but only 7 of these articles reported significant associations ($P < 0.05$ or an author specified significance threshold) and thus comprise our replication study (**Table 2.1**, **Figure 2.2**). A total of 56 articles tested associations in either a combination of MRD and MUD-HSCT, only MRD and/or in non-European populations; 39 of these 56 articles reported at least one significant SNP association with survival outcome and we attempted to validate the significant findings from these 39 articles (**Table 2.1**, **Figure 2.3**) (Karaesmen et al. 2017).

Replication

To reproduce previous reported results, DISCOVeRY-BMT was used to replicate findings that comprised of acute leukemia or MDS patients with European ancestry, treated with MUD-HSCT (Karaesmen et al. 2017). Seven articles were included in the replication analyses. Multi-allelic models, *CCR5* and *NOD2/CARD15*, were tested in 2 articles; single SNP associations in *CD274*, *CD40*, *HMGB1*, *IL1A*, *IL1B*, *NOD2/CARD15*, *TGFB1*, and *TNFSF4* were tested in 5 articles (**Table 2.1**, **Figure 2.2**) (Karaesmen et al. 2017).

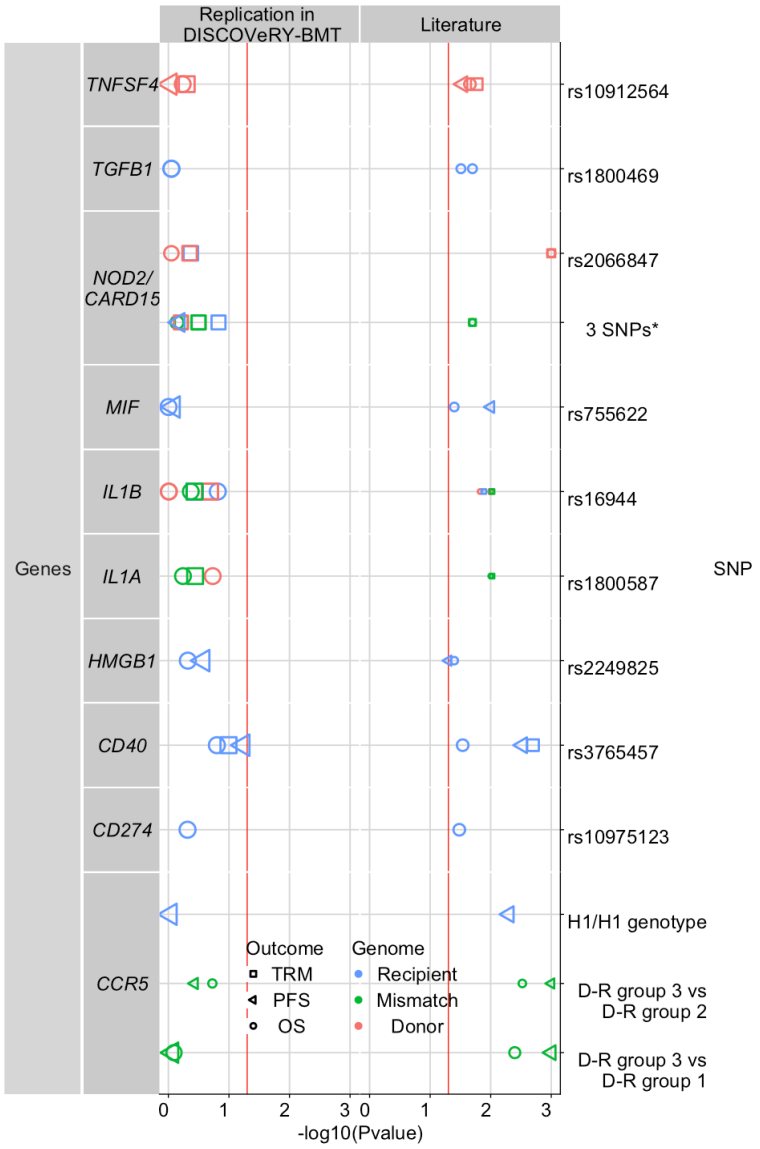

Figure 2.2: Replication attempts of previously reported significant candidate gene-association studies in DISCOVeRY-BMT. Survival association P values as reported in previous literature (Right panel) and replication attempts of these associations in DISCOVeRY-BMT cohort (Left panel) are shown as data points. Vertical panels indicate the genes that these polymorphisms and haplotypes are located in or close to as reported by the previous studies. Shapes represent associations with survival outcomes OS, PFS, or TRM; colors correspond to donor, recipient, or donor-recipient mismatch polymorphisms. The size of the point represents the sample size of the study, with larger points reflecting a bigger sample size. Shown on y-axis are the 9 polymorphisms from the literature reporting associations at $P < .05$ with OS, PFS, or TRM by 1 or more previously published studies; the x-axis is the $\log_{10}$(P value). The red vertical lines in (left panel) and (right panel) indicate $P < .05$. Details on the haplotypes are described in "Results" under "Replication."

Table 2.1: Count of reports with SNPs that were studied at least twice (in addition to CCR5 studies) that were attempted for replication or validation in DISCOVeRY-BMT.

SNP	Gene	Reports of SNPs with any HSCT outcome	Significant SNPs with any HSCT outcome	Reports of SNPs European ancestry MUD-HSCT populations	Significant SNPs in European ancestry MUD-HSCT populations	Reports of SNPs tested non-European ancestry and/or MRD-HSCT populations	Significant SNPs in non-European ancestry and/or MRD-HSCT populations
rs1045642	ABCB1	3	1	0	0	3	1
rs2032582	ABCB1	2	1	0	0	2	1
^ R-D Group 3 vs R-D Group 1	CCR5	1	1	1	1	0	0
^ R-D Group 3 vs R-D Group 2	CCR5	1	1	1	1	0	0
H1/H1 genotype	CCR5	1	1	1	1	0	0
rs2569190	CD14	2	1	0	0	2	1
rs231775	CTLA4	9	4	1	0	8	4
rs3087243	CTLA4	9	2	1	0	8	2
rs5742909	CTLA4	6	1	0	0	6	1
rs4553808	CTLA4	5	1	1	0	4	1
rs4244285	CYP2C19	2	1	0	0	2	1
rs8192917	GZMB	2	1	0	0	2	1
rs1800587	IL1A	3	1	0	0	3	1
rs16944	IL1B	4	1	0	0	4	1
rs1800587	IL1B	3	1	0	0	3	1
rs11209026	IL23R	3	2	0	0	3	2
rs1800795	IL6	5	3	0	0	5	3
rs1800797	IL6	3	1	0	0	3	1
rs1801133	MTHFR	4	3	0	0	4	3
rs1801131	MTHFR	3	1	0	0	3	1
3 SNPs*	NOD2/ CARD15	9	4	2	1	7	3
rs2066847	NOD2/ CARD15	4	2	1	1	3	1
rs2066842	NOD2/ CARD15	3	1	0	0	3	1
rs9658254	NOS1	2	1	0	0	2	1
rs1800469	TGFB1	3	1	1	1	2	0
rs4986790	TLR4	2	1	0	0	1	1
rs731236	VDR	5	2	0	0	5	2
rs7975232	VDR	5	2	0	0	5	2

^ CCR5 H1/H1 genotypes and risk groups defined using multiallelic models described in the original publication (McDermott et al 2010) and in the main text.
3 SNPs* – The NOD2/CARD15 3 SNP multiallelic model. Described in the text.

As previously discussed, candidate genes and candidate SNPs are selected due to *a priori* biological knowledge of the phenotype. The general thought in HSCT is that patients are either dying from their disease or from the transplant. And transplant related causes are mostly infection, organ failure, or GVHD. Relevant biological pathways that are included in these phenotypes are likely to involve immunological and inflammatory pathways. Indeed, the most frequently studied gene was *NOD2/CARD15*, which is a susceptibility gene for inflammatory bowel disease and may be involved in Crohn's disease (E Holler et al. 2008) (**Table 2.1**). The two *NOD2/CARD15* associations were based on a three-variant R-D pair model [rs2066844 (SNP8), rs2066845 (SNP12) and rs2066847 (SNP13)] and single SNP associations with SNP13 (E Holler et al. 2008). The null genotype is when the R-D pair are homozygous common allele for all 3 SNPs and the effect allele combination is the presence of 1 or more minor alleles at any of the 3 SNPs within the R-D pair. The *NOD2/CARD15* multi-SNP model was significantly associated with OS (RR: 1.6, 95% CI 1.1-2.4, $P = 0.02$) and TRM (RR: 1.6, 95% CI 1.1-2.4, $P = 0.02$) in a study of 196 patients who received a MUD-HSCT for AML or ALL. However, no associations were found for OS (HR: 1.03, 95% CI 0.9-1.2, $P = 0.72$) or TRM (HR: 1.1, 95% CI 0.8-1.4, $P = 0.6$) in DISCOVeRY-BMT, which is a larger study (N=1597) with AML and ALL patients treated with MUD-HSCT (**Figure 2.2**). Furthermore, a study of 342 AML or ALL patients after MUD-HSCT (E Holler et al. (2008)) reported donor genotype rs2066847 (SNP13) significantly increased risk of TRM and OS approximately 3-fold ($P = 0.001$) and 2.5 ($P = 0.001$), respectively. When this SNP was tested in DISCOVeRY-BMT donors, no associations were found with either TRM (HR: 1.17, 95% CI 0.78-1.74, $P = 0.45$) or OS (HR: 0.98, 95% CI 0.73-1.31, $P = 0.89$, in ALL or AML patients) (**Figure 2.2**).

One of the largest CGAS (N=1370) reported significant associations for *CCR5*

H1/H1 genotype (N=163) in recipients (McDermott et al. 2010). McDermott and colleagues also defined genotype risk subgroups and OS (**Figure 2.2**, **Table 2.1**). These associations were tested in DISCOVeRY-BMT and neither *CCR5* H1/H1 genotype (N=294) nor the genotype risk groups defined by H1/H1 status were significantly associated with PFS or OS (**Figure 2.2**, **Table 2.1**). The genotype risk groups (Group 3 vs Group 1 and Group 3 vs Group 2) were substantially smaller than the full cohort. We tested these in DISCOVeRY-BMT and found no associations. Considering the fact that DISCOVeRY-BMT cohorts were approximately twice as large as those in the original study and adequately powered to detect these associations (**Figure 1.11**), these risk group associations were not real (Karaesmen et al. 2017). In DISCOVeRY-BMT these subgroups were approximately twice as large as those in the original study and adequately powered to detect these associations.

DISCOVeRY-BMT was unable to replicate another large CGAS of 1170 patients (Jindra et al. 2016), which reported an association between rs10912564 (*TNFSF4*) and TRM ($P = 0.017$), OS ($P = 0.022$), and PFS (HR: 0.8, 95% CI [0.9-1.2], $P = 0.03$) (**Figure 2.2**). Similarly DISCOVeRY-BMT could not replicate rs2249825 in *HMGB1* (Kornblit et al. 2010, N=276), rs1800469 in *TGFB1* (Arrieta-Bolaños et al. 2016, N=493), rs755622 in *MIF* (Chang et al. 2009, N=454), nor SNPs in *IL1A* and *IL1B* (**Figure 2.2**).

Validation

Validation attempts were conducted on 36 genetic polymorphisms in 26 genes from 39 previously published CGAS (**Table 2.1**). The genes that were included: *ABCB1*, *CD14*, *CTLA4*, *CYP2C19*, *DAAM2*, *EP300*, *ESR1*, *GSTA2*, *GZMB*, *ICAM1*, *IL23R*, *IL6*, *IRF3*, *KLRK1*, *LIG3*, *MTHFR*, *MUTYH*, *NOD2/CARD15*,

NOS1, *P2RX7*, *TDG*, *TIRAP*, *TLR4*, *TYMP*, and *VDR* (**Figure 2.3**) (Karaesmen et al. 2017). Each of these studies reported at least one significant genetic associations with survival in patients who received a HLA MRD-HSCT (19 articles) or had a study population including MRD- and MUD-HSCT patients, without stratification of results (17 articles). Validation attempts for survival associations reported in non-European leukemia patients who received an MUD-HSCT (3 articles) (**Table 2.1**). The variants that had been reported to be associated to survival outcomes after transplant and our attempts to validate the results in DISCOVeRY-BMT are shown in **Figure 2.3**.

Only one of our validation attempts were successfully reproduced – where donor genotype rs1800795 (*IL-6*) was associated with OS (HR: 1.11, 95% CI 1.0-1.2, $P = 0.02$) (**Figure 2.3**, note: this did not reach genome-wide significance at the $P < 5x10^{-08}$ threshold). The rs1800795 association to OS (HR: 1.29, 95% CI 1.07-1.55, $P = 0.007$) was reported in a single study in patients with acute leukemia, CML, or lymphoma (N=743) treated with MRD- or MUD-HSCT (Balavarca et al. 2015).

Similar to our replication set, *NOD2/CARD15* was the most frequently studied gene and reported the most of all CGAS in our validation set (**Table 2.1**). Three studies reported associations between the presence of *NOD2/CARD15* multi-SNP polymorphism and either TRM (Ernst Holler et al. 2004) or PFS. None of these studies were validated in DISCOVeRY-BMT cohorts (**Figure 2.3**). Likewise, DISCOVeRY-BMT was unable to validate single SNP analyses in *NOD2/CARD15* for rs2066842 in MRD- or MUD-HSCT donors with PFS or for TRM at rs2066847 (SNP13) in recipients of MRD- or MUD-HSCT (**Figure 2.3**).

CTLA4 is a member of the immunoglobulin superfamily that is expressed by activated T-cells and transmits inhibitory signal to T=cells. Due to these known

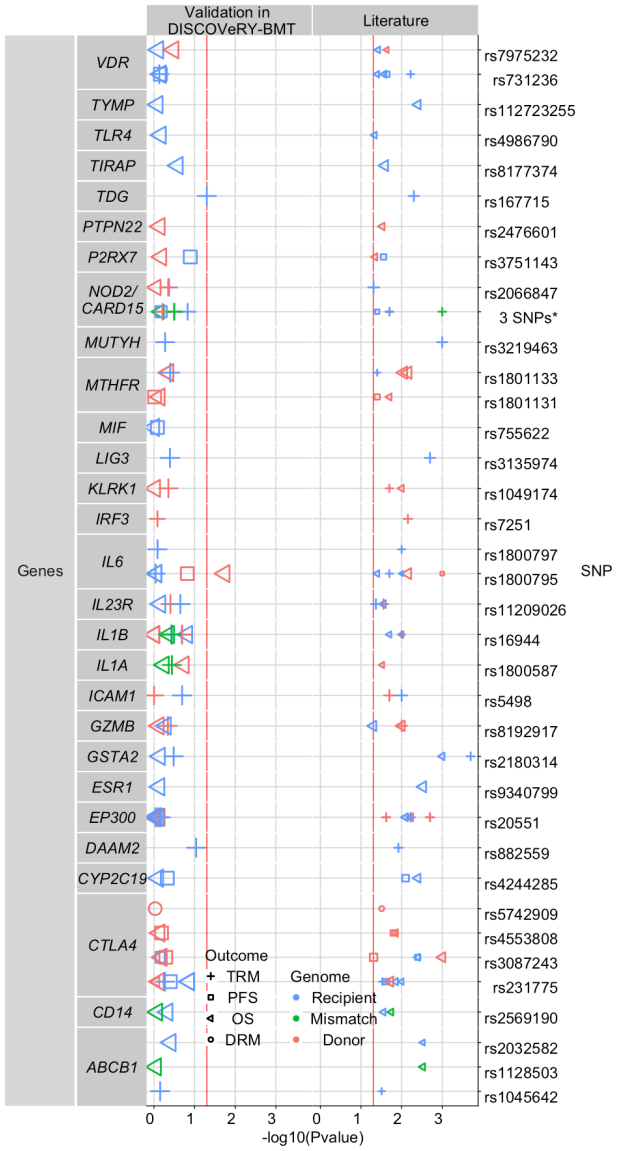

Figure 2.3: Validation attempts of previously reported significant candidate gene-association studies. Survival association P-values as reported in previous literature (Right panel) and validation attempts of these associations in DISCOVeRY-BMT cohort (Left panel) are shown as data points. Vertical panels indicate the genes that these 17 polymorphisms are located in or closest to as reported by the previous studies. Shapes represent associations with survival outcomes DRM, OS, PFS, and TRM; colors correspond to donor, recipient, or donor-recipient mismatch polymorphisms. The size of the point represents the sample size of the study, with larger points reflecting a bigger sample size. Shown on y-axis are the 17 polymorphisms from the literature reporting associations at $P < .05$ with OS, PFS, or TRM by 1 or more previously published studies; the x-axis is the $\log_{10}$(P-value). The red vertical lines in (left panel) and (right panel) indicate $P < .05$. Details on the haplotypes are described in "Results" under "Validation."

functions and perceived implications in transplant biology, associations with multiple SNPs in *CTLA4* have been tested in numerous transplant populations (**Table 2.1**) (Pérez-García et al. 2007). In previous CGAS, four *CTLA4* SNPs (rs3087243, rs231775, rs4553808, rs5742909) were significantly associated with survival after MRD- or MUD-HSCT in patients with acute leukemia, CML, lymphomas, MDS, and other hematological disorders (**Table 2.1**). Attempts to validate CTLA4 SNPs with DRM, PFS, OS, and TRM were unsuccessful in the DISCOVeRY-BMT cohorts (**Figure 2.3**). The remaining results of the 25 additional candidate genes containing SNPs that were tested in the DISCOVeRY-BMT cohorts are summarized in **Figure 2.3**; no SNP associations were found at $P < 0.05$. Importantly, the P-value distribution of the single SNP associations showed no deviation from the null expectation with 95% confidence intervals (**Figure 2.4**), suggesting we cannot reject the null hypothesis of no association with survival outcome (Karaesmen et al. 2017).

Gene based replication and validation of previous studies

From the previous literature, candidates genes were first selected genes based on their hypothesized or known function, and subsequently authors selected variants within that gene for single SNP or haplotype testing (Karaesmen et al. 2017). Thus, while SNPs and haplotypes were tested individually for association, the hypotheses from the literature can be considered gene-based. The density of typed and imputed markers in the DISCOVeRY-BMT recipients and donors allows us to measure the aggregate effect of all SNPs within each candidate gene on survival. Genes were selected for testing from the same literature summarized above for the replication and validation SNP and haplotype analyses. VEGAS2 gene-based testing did not reveal any associations at $P < 0.05$ with any of the survival outcomes in either the replication or validation groups (Karaesmen et al. 2017).

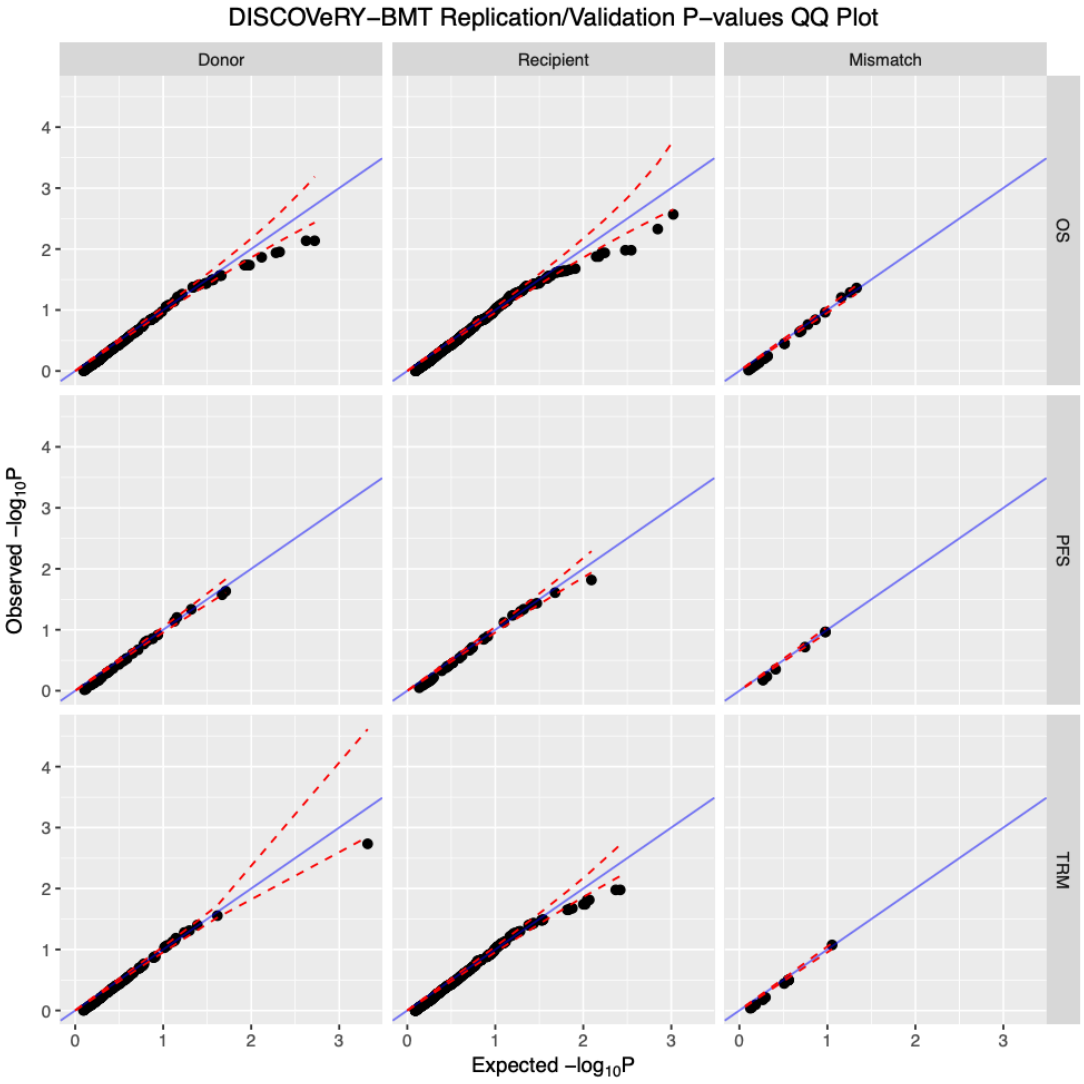

Figure 2.4: Quantile-quantile (QQ) plot of SNP p-values in DISCOVeRY-BMT for all previously studied SNPs. From left to right, vertical panels represent the donor, recipient, and mismatch genome, horizontal panel represents the survival outcome that was tested, from top to bottom overall survival (OS), progression free survival (PFS) and transplant related mortality (TRM). Dashed red lines indicate 95% confidence intervals

Candidate polymorphism annotation

Candidate gene SNPs were analyzed using the RegulomeDB, VEP and Blood eQTL Browser databases to assess their functional characteristics and better understand their biological framework. Eighty percent of previously reported SNPs had RegulomeDB scores greater than 3 (**Figure 2.5**), indicating that these SNPs have minimal to no effect on modifying transcription. This distribution aligns with the overall distribution of SNPs in the genome, thus the candidate SNPS are not enriched for their impact on gene expression or transcription factor binding. Our replication and validation analyses includes 2 protein coding variants, VEP shows that only, rs2066845 (SNP12) in *NOD2/CARD15*, is predicted to be damaging and disease causing.

We were interested in better understanding the biological framework of candidate SNPs that had previously been identified. The candidate SNPs were analyzed using RegulomeDB, Blood eQTL Browser, and VEP to assess functional characteristics. The majority (*geq* 80%) candidate SNPs had RegulomeDB scores greater than 3 (**Figure 2.5**). This indicates that the SNPs had minimal to no evidence of modifying DNA transcription. The observed distribution coincides with the overall distribution of SNPs in the genome. Therefore, from our point of view, the candidate SNPs are not enriched in terms of impact on TF binding or gene expression. VEP analyses revealed that the candidate SNPs in our replication and validation analyses only included 2 protein coding variants, and that rs2066845 (SNP12) in *NOD2/CARD15* is predicted to be damaging and disease causing (Karaesmen et al. 2017).

We looked at the Blood eQTL browser to determine if the candidate SNPs played a roll in *cis* gene expression of the candidate genes. Rough half ($\approx$ 52%) of

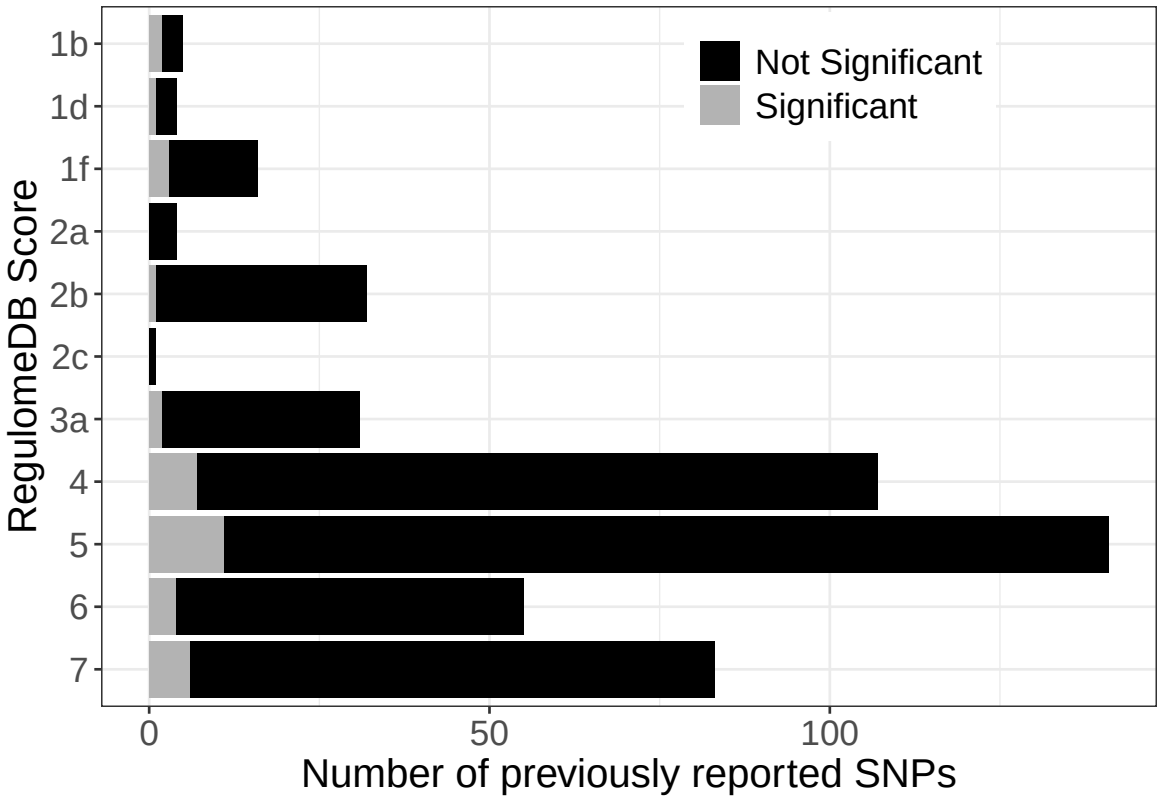

Figure 2.5: RegulomeDB score distribution of previously studied polymorphisms. RegulomeDB categories are shown on the x-axis; counts of SNPs falling into RegulomeDB score category are shown on the y-axis. Blue portion of the bar indicates the counts of SNPs that were tested but not reported significant; red portion shows the counts of SNPs that were reported significant at least once. Score descriptions are given below the image. 1b indicates eQTL 1 transcription factor (TF) binding 1 any motif 1 DNase footprint 1 DNase peak; 1d, eQTL 1 TF binding 1 any motif 1 DNase peak; 1f, eQTL 1 TF binding/DNase peak; 2a, TF binding 1 matched TF motif 1 matched DNase footprint 1 DNase peak; 2b, TF binding 1 any motif 1 DNase footprint 1 DNase peak; 2c, TF binding 1 matched TF motif 1 DNase peak; 3a, TF binding 1 any motif 1 DNase peak; 4, TF binding 1 DNase peak; 5, TF binding or DNase peak; 6, motif hit; 7, no evidence.

the 171 genes that are included from the literature search had at least one significant (probe level false discovery rate, FDR < 0.05) *cis*-eQTL. And when compared to the entire genome, approximately 44% of genes have blood *cis* eQTLs (FDR $P < 0.05$). However, when we look at the the candidate SNPs, only 13% are blood *cis*-eQTLs. So while blood eQTLs have indeed been identified in these genes, the SNPs that previous literature studied, were not the ones that affected expression. Furthermore, almost half of the eQTLs in the CGAS are correlated with expression, they alter expression in neighboring or nearby genes rather than the gene they are located in. For example, rs7975232 (*VDR*) is an eQTL for *SLC48A1* while the *CTLA4* SNPs studied are eQTLs for *CD28* (Karaesmen et al. 2017). The remaining eQTLs were correlated with expression of the candidate gene of interest, but in most cases, were also significant eQTLs for several other nearby genes (Karaesmen et al. 2017).

Discussion

In this study, we aimed to replicate or validate all existing CGAS literature that investigated the relationship between non-HLA genetic effects on survival after allogeneic HSCT (Karaesmen et al. 2017). Considering that the main purpose of CGAS is to select SNPs due to known function at the gene level, we conducted single SNP analyses and gene-based analyses to determine the joint effects of marginally associated SNPs within candidate genes while considering the underlying genetic correlation structure due to LD.

Donor rs1800795 in *IL6* was the only association (OS) that was able to be reproduced using DISCOVeRY-BMT (only in the validation set; no candidate SNPs replicated). The original authors (Balavarca et al. 2015) studied *IL6* due to its known immunological function, as well as the fact that two prior studies reported

significant associations in different phenotypes: to GVHD (Cavet et al. 2001) and chronic hepatitis C virus therapy (Yee et al. 2009). No evidence of an association of rs1800795 at $P < 0.05$ in DISCOVeRY-BMT with death due to GVHD and/or infection (data not shown). Furthermore, rs1800795 is found in an intronic region of *IL6*, and has no effect on *IL6* expression or levels, but rather is an eQTL for two other nearby genes (Arnold et al. 2015, GTEx Consortium (2013)). We made an extra effort to validate interesting CGAS findings, such as in *CCR5* and the H1/H1 risk groups from McDermott et al. 2010. These associations were found in the largest study we attempted to validate, as well as having samples from CIBMTR (earlier years than DISCOVeRY-BMT), and the reported effects started two years after HSCT. We were not able to validate these associations with either outcome at $P < 0.10$ (the threshold McDermott 2010 used, **Figure 2.3**).

CTLA4 was another gene that was studied frequently (**Table 2.1**), this gene is a good example of the heterogeneity to studies of genetic polymorphisms in HSCT and may be a plausible explanation of why were unable to reproduce any of the previous associations. The SNP rs5742909 in *CTLA4* was studied in 6 independent studies of HLA MRD-HSCT donor-recipient pairs. For donors, rs5742909 was reported to be associated with DRM in a smaller study (N=120). The lack of consistency between studies is apparent – previously two *CTLA4* SNPS, rs231775 and rs231774 were reported associated in recipients for PFS ($P = 0.025$) – and before this study, only 1 of 9 studies tried to validate these results. These SNPs are like many candidate gene hypotheses such that they have not been tested in the same genome for the same outcome in similar populations, and if they have, the sample size is very small.

The inclusion criteria with respect to disease, donor relation, or to differences in our endpoint of 1-year survival versus longer-term survival could be attributed

as to why previous results are not reproducible. As previous genetic associations were hypothesized to be independent of the underlying disease, one may expect that results should be reproducible in a homogeneous patient populations such as DISCOVeRY-BMT, which unfortunately was not the case. This study meticulously attempted to align the study population in DISCOVeRY-BMT to the original studies (i.e. subset the full population for this only with AML). DISCOVeRY-BMT primarily focuses on 1-year survival after HSCT and this may have different genetic effects than later survival, however, the survival curves from the CGAS often showed separation of death occurring well into the first year. DISCOVeRY-BMT has a large sample size that is adequately powered (L. E. Sucheston-Campbell et al. 2015) to reproduce results from previously published CGAS. However, we were unable to reproduce almost all of the findings. This is in agreement with other studies in this field, Particularly in the context of death due to GVHD as an outcome (Chien et al. 2012; Martin et al. 2016). Several other reports have also concluded that published CGAS have presented false positive associations (Ioannidis et al. 2001).

Confirmatory studies are vital to identify true positive genetic variants that are associated to complex phenotypes. False positives may lead researchers to allocate resources and time to research that will lead to a dead end. In the long-term this may delay treatment to patients or discovery of markers that would be relevant in improving survival after HSCT. As only one SNP had a predicted damage or deleterious effect and that a only a small proportion of the previous 70 studies conducted research on SNPs that correlated with gene expression, we conclude that the transplant field can look agnostically at the entire genome to hone in on potential candidate genes and SNPs (Karaesmen et al. 2017). And while we could not reproduce previous associations, the SNPs selected are not linked to functional an-

notation nor are they clearly related to the candidate genes. This underscores a fundamental flaw with CGAS which are contingent on the scientific knowledge at the time. We assert that by using adequately powered testing (i.e. larger studies, GWAS) with transplant will be vital to identifying promising targets and further help us understand the underlying biological framework and how it relates to survival outcomes, with the global objective of improving patient outcomes (Karaesmen et al. 2017).

CHAPTER 3: gwasurvivr: an R Bioconductor package for genome wide survival analysis

Introduction

Genome-wide association studies (GWAS) are population-level experiments that investigate genetic variation in individuals to observe single nucleotide polymorphism (SNP) associations with a phenotype. Genetic variants tested for association are genotyped on an array and imputed from a reference panel of sequenced genomes using 1000 Genomes Project or Haplotype Reference Consortium (HRC) (Das et al. 2016; 1000 Genomes Project Consortium 2015). Imputation increases genome coverage from hundreds of thousands or a few million to upwards of 30 million SNPs, improves power to detect genetic associations, and/or homogenizes variant sets for meta-analyses (Das et al. 2016). Imputed SNPs can be tested for association with binary outcomes (case/control) and quantitative outcomes (i.e. height) using a range of available software packages including SNPTEST (Marchini et al. 2007) or PLINK (Purcell et al. 2007).

However, existing software options for performing survival analyses, genipe (Lemieux Perreault et al. 2016), SurvivalGWAS_SV (Syed, Jorgensen, and Morris 2017), or GWASTools (S. M. Gogarten et al. 2012), across millions of imputed SNPs either require user interaction with raw output, were not initially designed for survival and/or have other limitations that could deter more introductory users developed an R/Bioconductor package, gwasurvivr, for genome wide survival analyses of imputed data in multiple formats with flexible analysis and output options (A. A. Rizvi et al. 2018).

Methods

Data Structure

Gwasurvivr can analyze data in IMPUTE2 format (B. N. Howie, Donnelly, and Marchini 2009), in VCF files derived from Michigan (Das et al. 2016) or Sanger imputation servers (McCarthy et al. 2016), and directly genotyped PLINK format (Purcell et al. 2007). Data from each are prepared in gwasurvivr by leveraging existing Bioconductor packages GWASTools (S. M. Gogarten et al. 2012) or VariantAnnotation (Obenchain et al. 2014) depending on the imputation file format.

IMPUTE2 Format: IMPUTE2 (B. N. Howie, Donnelly, and Marchini 2009) format is a standard genotype (.gen) file which store genotype probabilities (GP). We utilized GWASTools in R to compress files into genomic data structure (GDS) format (S. M. Gogarten et al. 2012). This allows for efficient, iterative access to subsets of the data, while simultaneously converting GP into dosages (DS) for use in survival analyses.

VCF Format: VCF files generated from these Michigan or Sanger servers include a DS field and server-specific meta-fields (INFO score [Sanger] or r^2 [Michigan], as well as reference panel allele frequencies) that are iteratively read in by VariantAnnotation (Obenchain et al. 2014).

PLINK Format: Plink bed files contain genotype information encoded in binary format. Fam and bim files include the information of phenotype and marker location, respectively (Purcell et al. 2007).

gwasurvivr implements a Cox proportional hazards regression model (Cox 1972) to test each SNP with an outcome with options for including covariates and/or SNP-covariate interactions. To decrease the number of iterations needed for convergence when optimizing the parameter estimates in the Cox model we

modified the R package survival (Therneau and Grambsch 2013). Covariates in the model are first fit without the SNP, and those parameter estimates are used as initial points for analyses with each SNP. If no additional covariates are added to the model, the parameter estimation optimization begins with null initial value. (**Figure 3.1**).

gwasurvivr calculations

Minor Allele Frequency (MAF)

For a given SNP with alleles A and B, where n_{AB} and n_{BB} are the number of individuals with AB and BB genotype respectively, and N is the sample size, the expected allele frequency of allele B ($freq_B$) be can be calculated as:

$$freq_B = \frac{n_{AB} + 2n_{BB}}{2N}$$

For individual i, the allele dosage of SNP j (D_{ij}) with alleles A and B, where allele B is the effect allele and p_{AB} and p_{BB} are the posterior genotype probabilities as computed by the imputation, is calculated as:

$$D_{ij} = p_{AB_{ij}} + 2 \cdot p_{BB_{ij}}$$

For SNP j The estimated allele frequency of an effect allele B (θ_{B_j}) can therefore be calculated as:

$$\theta_{B_j} = \frac{\sum_{i=1}^{N} D_{ij}}{2N}$$

In R, the genotypes are represented as a matrix allele dosages, where each column is a sample and each row is a SNP. The R code is shown below:

```
library(matrixStats)
exp_freq_A1 <- round(rowMeans2(genotypes) * 0.5,
    4)
MAF <- ifelse(exp_freq_A1 > 0.5, 1 - exp_freq_A1,
    exp_freq_A1)
```

Imputation quality metric

Michigan Imputation Server

For the Michigan imputation server, imputation is performed using the minimac3 algorithm (Das et al. 2016). minimac3 computes and outputs an imputation quality metric known as R^2. R^2 is the estimated value of the squared correlation betwen imputed genotypes and true, unobserved genotypes (Das et al. 2016). The R^2 value is extracted directly from the Michigan imputation output VCF in `gwasurvivr::michiganCoxSurv`

Sanger Imputation Server

For the Sanger imputation server, we grab the `INFO` field directly from the VCF file in `gwasurvivr::sangerCoxSurv`. The INFO field is the IMPUTE2 (B. N. Howie, Donnelly, and Marchini 2009) score as calculated by the `bcftools + impute-info` plugin from posterior genotype probabilities (McCarthy et al. 2016).

IMPUTE2 Imputation

The INFO score for IMPUTE2 (B. N. Howie, Donnelly, and Marchini 2009) results are not calculated in `gwasurvivr` internally, instead we use the INFO scores that are provided in a separate file after performing imputation (`.info` file). Users select SNPs from the `.info` file to remove based on preferred criterion (ie INFO <

.8) these are then used in the argument `exclude.snps` in `impute2CoxSurv` to filter out the SNPs prior to analysis.

Survival Analysis

Survival analyses are run using genetic data in either VCF or IMPUTE2 (B. N. Howie, Donnelly, and Marchini 2009) formats and a phenotype file, which contains survival time, survival status and additional covariates, both files are indexed by sample ID. In addition to genomic data, the VCF files contain both sample IDs and imputation quality metrics (INFO score or r^2), while IMPUTE2 (B. N. Howie, Donnelly, and Marchini 2009) come in separate files (.gen, .sample, and .info). Gwasurvivr functions for IMPUTE2 (impute2CoxSurv or gdsCoxSurv) and VCF (michiganCoxSurv or sangerCoxSurv) include arguments for the survival model (event of interest, time to event, and covariates) and arguments for quality control that filter on minor allele frequency (MAF) or imputation quality (michiganCoxSurv and sangerCoxSurv only). INFO score filtering using impute2CoxSurv can be performed by accessing the .info file from IMPUTE2 results and subsequently providing the list of SNPs to 'exclude.snps' argument to gwasurvivr. Users can also provide a list of sample IDs for gwasurvivr to internally subset the data. gwasurvivr outputs two files:

(1) `.snps_removed` file, listing all SNPs that failed QC parameters

(2) `.coxph` file with the results from the analyses, including parameter estimates, p-values, MAF, the number of events and total sample N for each SNP.

gwasurvivr also allows the number of cores used during computation on Windows and Linux to be specified. Users can keep compressed GDS files after the initial run by setting keepGDS argument to TRUE when analyzing IMPUTE2 data (B. N. Howie, Donnelly, and Marchini 2009). On successive runs, gdsCoxSurv can then be

used instead of impute2CoxSurv to avoid compressing the data on each GWAS run.

Computational Experiments

We used the University at Buffalo Computational Center for Research (UB CCR) academic cluster for our benchmarking analyses. Each analysis was run exclusively on node CPU-L5520 with the same system specifications, controlling the computational resources for each run. The UB CCR uses Simple Linux Utility for Resource Management (SLURM) scheduling for jobs. SLURM scripts to run the analyses were generated using shell scripts below. Benchmarking was performed using identical CPU constraints, 1 node (2.27 GHz Clock Rate) and 8 cores with 24 GB of RAM, on the University at Buffalo Center for Computational Research supercomputer. With the exception of the larger sample size tests, these were run using the same node but 12 CPUs. genipe (Lemieux Perreault et al. 2016), SurvivalGWAS_SV (Syed, Jorgensen, and Morris 2017), and GWASTools (S. M. Gogarten et al. 2012) were performed as specified by the authors on available online documentation.

We performed the following benchmarking runtime experiments either against existing software or against time with varying N and SNP numbers that were performed:

Simulation 1. Compare gwasurvivr against genipe, GWASTools and SurvivalGWAS_SV - varying sample sizes (n=100, n=1000, n=5000) and 100,000 SNPs (m=100000) and 3 non-genetic covariates

Simulation 2. Comparison of gwasurvivr, genipe, GWASTools and SurvivalGWAS_SV with N=5,000 and 100,000 SNPs (m=100,000) with 4 covariates (age, drug treatment, sex and 1 PC), 8 covariates (age, drug treatment, sex and 5 PCs) and 12 covariates (age, drug treatment, sex and 9 PCs)

Simulation 3. Increasingly larger sample sizes (N=15K, 20K and 25K) tested on Chromosome 22

Simulation 4. Full autosomal GWAS with varying sample sizes (N=3K, 6K and 9K)

Simulating Genotypes and Phenotypes

Genotypes

HAPGENv2 (Su, Marchini, and Donnelly 2011) was used to generate simulated genetic datasets from 1000 Genomes Project CEU data (NCBI Build 36) for all benchmarking experiments. To replicate simulations the 1000 Genomes Project CEU data should be downloaded in its entirety (only a subset is available on our GitHub repo). The code for all HAPGENv2 simulations are available on our GitHub.

Phenotypes

For each sample size tested, survival events (alive/dead) were simulated as two separate datasets. For the dead dataset, time to event and covariates were simulated using a normal distribution. For the alive dataset, time was simulated by randomly sampling weighted probabilities for times to simulate few samples being censored, covariates were simulated from a normal distribution. Principal components (PCs) were simulated using random normal distributions with decreasing variance for each additional PC. Furthermore, the `.sample` file from IMPUTE2 includes 4 columns (ID_1, ID_2, missing, and sex) which link individuals with their respective genotypes. For SurvivalGWAS_SV and GWASTools, the simulated phenotypes were appended to column 5 onward in the `.sample` file.

The following genotypes and phenotypes were simulated:

Simulations 1 and 2. Subset of chromosome 18 for 100,000 SNPs 1) varying N and 3 covariates done in triplicate and 2) with 4, 8 and 12 covariates

- genotype code
- phenotype code
- PCs phenotype code

Simulation 3. chromosome 22 (~117,000 SNPs) for larger sample sizes (N=15000-25000)

- genotype code

Simulation 4. Full GWAS for N=9000 (the smaller subsets were just parsed from the data during analyses)

- genotype code
- phenotype code
- simulate sample ids code

Benchmarking with other software capable of GWAS coxph survival analysis

We benchmarked `gwasurvivr` with GWAS survival analysis software, genipe, SurvivalGWAS_SV and GWASTools using simulated phenotype and genotype data. Genetic data were formatted as output from IMPUTE2 software (.GEN). Genipe, SurvivalGWAS_SV, and GWASTools do not directly take VCF data output from Sanger or Michigan imputation servers. SurvivalGWAS_SV does accept VCF files as an input but uncompressed and not explicity the same format that Sanger and Michigan imputation servers output, rendering additional steps to be taken. The benchmarking with IMPUTE2 was done for (1) varying sample sizes and (2) varying additional non-genetic covariates. Both are described here.

gwasurvivr

The following scripts were used to run gwasurvivr using `impute2CoxSurv`. These R scripts are run using a shell script (SLURM script) that pass the system variables into R (facilitated by the R package `batch`).

N=100, 1000 and 5000 with M=100K SNPs + 3 non-genetic covariates in triplicate:

- `run_gwasurvivr.R`

- `create_gwasurvivr_scripts.sh`

N=5,000 and M=100K with 4, 8 and 12 covariates:

- `run_gwasurvivr_covs.R`

- `gwasurvivr_cov4.sh`

- `gwasurvivr_cov8.sh`

- `gwasurvivr_cov12.sh`

genipe

For genipe, the shell scripts was used to generate SLURM scripts for genipe and each sample and SNP set. We used specific settings for OPENBLAS that are suggested on genipe's website to ensure that computational efficiency was maximized.

varying sample sizes + 3 non-genetic covariates:

- `create_genipe_scripts.sh`

additional covariates:

- `genipe_cov4.sh`

- `genipe_cov8.sh`

- `genipe_cov12.sh`

SurvivalGWAS_SV

To maximize the performance of SurvivalGWAS_SV, these jobs were run using "array" jobs as recommended by the authors. An example batch script, provided in the SurvivalGWAS_SV documentation, was converted from PBS to SLURM. 24GB of ram was not needed on all runs, however was used to ensure each run remained uniform. The jobs were split into array sets of 1000 SNPs for m=100,000, totaling 100 batched jobs in a single array. We define rate-limiting array as the array index that had the longest runtime. In the main manuscript, we report SurvivalGWAS_SV runtimes as the rate-limiting array runtime. This is an important caveat and bears consideration when using SurvivalGWAS_SV. Depending on availability on the computing cluster, the analyses could be completed as quickly as the longest individual array job (which is shown in Figure 1), or potentially the entire runtime could be equal to the summation runtime of all of the array indices if these cannot be run simultaneously (or if there are failures with any of the array indices). The shell script below was used to generate SLURM scripts for SurvivalGWAS_SV for each sample and SNP set.

N=100, 1000 and 5000 with M=100K SNPs + 3 non-genetic covariates in triplicate:

- `create_sv_scripts.sh`

N=5,000 and M=100K with 4, 8 and 12 covariates:

- `sv_cov4.sh`
- `sv_cov8.sh`
- `sv_cov12.sh`

GWASTools

For GWASTools, the files are converted to GDS format and survival is run using `GWASTools::assocCoxPH` within `gwastools_survival.R`. The R script was passed to the SLURM scripts using the script `create_gwastools_scripts.sh`. GWASTools does not run in parallel across multiple cores on a single computing processor internally, however experienced users could code this themselves.

N=100, 1000 and 5000 with M=100K SNPs + 3 non-genetic covariates in triplicate:

- `gwastools_survival.R`
- `create_gwastools_scripts.sh`

N=5,000 and M=100K with 4, 8 and 12 covariates:

- `gwastools_survival_covs.R`
- `gwastools_cov4.sh`
- `gwastools_cov8.sh`
- `gwastools_cov12.sh`

Runtime large N chromosomes to test size limitations

We tested chr22 with different sample sizes of N=15,000; N=20,000; N=25,000 using `gwasurvivr::impute2CoxSurv`. The code for all of the runs can be found here. The R script called from the shell scripts to run these analyses is labeled `run_bigNs.R`.

Runtime GWAS with different sample sizes

We performed three GWAS (chr1-chr22) with different sample sizes (n=3000; n=6000; n=9000) using `gwasurvivr::impute2CoxSurv`. The code to simulate the

GWAS is available on our repository. The R script used to run these analyses is `run_fullgwas.R`. The shell script run these scripts on SLURM can be found here.

Simulations and Benchmarking

Computational runtimes for gwasurvivr were benchmarked against existing software comparing varying sample sizes and SNP numbers, with 4, 8 or 12 covariates and for a single chromosome with 15,000-25,000 individuals. In addition, we evaluated time for gwasurvivr for a GWAS (~6 million SNPS) for 3000, 6000 and 9000 samples. All benchmarking experiments were performed using IMPUTE2 format (comparison packages do not take VCF from either imputation servers).

Descriptions of simulated genotype and phenotype data are in the Supplementary Data.

Developing an R package

Results

gwasurvivr was faster than genipe (Lemieux Perreault et al. 2016), SurvivalGWAS_SV (Syed, Jorgensen, and Morris 2017), and GWASTools (S. M. Gogarten et al. 2012) for 100,000 SNPs at N=100, and 5000, with the exception of SurvivalGWAS_SV at N=1000 (Figure 1A). Similarly, increasing the number of covariates for gwasurvivr has minimal effects on runtime versus other software (Figure 1B). Gwasurvivr computes for large sample sizes, however, compression time increases with increasing sample size, and likely will be limited by available RAM on a machine or cluster (Figure 1C). The keepGDS argument helps address this and results in reduced run times (Figures 1C and 1D), i.e. < 3 hours for a GWAS of N=9,000. A ~6 million SNP GWAS can be run in < 10 hours for 9000 samples when using

separately scheduled jobs on a supercomputer (Figure 1D). However, gwasurvivr overcomes memory limitations often attributed to R by processing subsets of the entire data, and thus it is possible to conduct genome-wide survival analyses on a typical laptop computer.

gwasurvivr is a fast, efficient, and flexible program well suited for multi-core processors and easily run in a computing cluster environment.

`gwasurvivr` is an R package that can be used to conduct survival analysis (Cox proportional hazards model) on imputed GWAS data from either IMPUTE2 (B. N. Howie, Donnelly, and Marchini 2009) or VCF files from the Michigan and/or Sanger imputation servers. `gwasurvivr` can also be used on directly typed data in plink format (`.bed`, `.bim` and `.fam` files).

Herein, we detail our implementation of the Cox model, generation of the simulated data and survival benchmarking and graphically report the correlation of `gwasurvivr` beta coefficient estimates, minor allele frequencies (MAF) and p-values with those produced from SurvivalGWAS_SV, genipe, and GWASTools.

To reproduce the data and create Figure 1 and Supplementary Figures 2-4, the data is available on the gwasurvivr manuscript repository. GitHub Large File Storage (LFS).

To clone the whole repository:

```
git lfs clone \
    https://github.com/suchestoncampbelllab/ \
        gwasurvivr_manuscript.git
```

Implementation of Survival Model in gwasurvivr

Modifying coxph

We decrease computation time by decreasing the number of Newton-Raphson iterations used to optimize the partial likelihood function in the Cox proportional hazard models. To do this, a survival model was fit using only non-genetic covariates (i.e. the SNP is not included and only covariates are fit); `survival::coxph` (Therneau and Grambsch 2013) is modified such that gwasurvivr manually creates the objects found in the helper function (`survival::coxph.fit`) that fits the Cox model.

These variables are then passed to `survival::coxph.fit`.

Benchmarking with survival package

To assess if providing initial estimates from covariates versus using the survival function as implemented in the survival package improves computational time, we tested a dataset of 500 individuals at 7255 SNPs with 1, 2, or 3 covariates. These data are a subset of the simulated data described in detail below.

The helper function `gwasurvivr:::coxParam`, adjusted for this Supplementary documentation is labeled `gcoxph`. In `gcoxph_model.R` we fit the model without the SNP and the parameter estimates are then used as initial points for all subsequent models and applied over all SNPs in the dataset. If there were no covariates, the initial estimates would be null. The function `coxph_model.R` implements a `survival` model (survival package, Therneau and Grambsch 2013) without using the optimization starting point obtained from including covariates in the model.

To test the package runtime over a pre-specified number of iterations and including 1, 2, or 3 covariates the `microbenchmark` package in R was used. The code

for Supplementary Figure 1) is available.

By leveraging an initalization point from the analyses with covariates `gwasurvivr` (gcoxph) is several seconds faster than the survival analyses function as implemented in `survival` (coxph, Therneau and Grambsch, 2000) in R (**Supplementary Figure 1**). While this is a small test dataset, in practice this would be an appreciable difference when testing across several thousands of samples and millions of SNPS. In the `gwasurvivr` package, we opted to use `parallel::parApply` instead of `base::apply` as shown above to compute across multiple cores.

Time Plots

Figure 1

To generate Figure 1 times from the computation runtime were pulled from SLURM log files and collected using the perl scripts, which can be found in each of the log folders on our manuscript GitHub repository, compiled and Figure 1 was generated using the R code shown here.

Diagnostic Plots

Supplementary Figures 2, 3 and 4 below show the correlation of the coefficient estimates, minor allele frequency and p-values, respectively between gwasurvivr and all other software assessed. The correlations show excellent agreement. The R code used to generate supplemental figures 2-4 can be found here.

Coefficient Estimates

Minor Allele Frequency (MAF)

P-value Estimates

Full GWAS Runtimes

Introduction

Does this say anything? Maybe this is the problem.

Methodology

Schematic of pipeline

Reimputation with HRC

Imputed data location

perl script

```
#! /usr/bin/perl
use warnings;
use strict;
use Cwd;
use Getopt::Long;

my $manifest;
my $walltime;
my $email;
my $rscript;
GetOptions ("m=s" => \$manifest, \
            "w=s" => \$walltime, \
            "e=s" => \$email, \
            "rscript=s" => \$rscript) or die$!;
my $wd = getcwd;
if (!-d "$wd/job_scripts") { mkdir "$wd/job_scripts"; }
if (!-d "$wd/log") { mkdir "$wd/log"; }
if (!-d "$wd/job_scripts/metal_scripts") \
    { mkdir "$wd/job_scripts/metal_scripts"; }
if (!-d "$wd/temp") { mkdir "$wd/temp"; }
```

```
if (!-d "$wd/out") { mkdir "$wd/out"; }
open MANIFEST, "$manifest" or die$!;
my $line = 1;
while(<MANIFEST>) {
    if ($line == 1) {
        $line++;
        next;
    }
    chomp($_);
    my @temparray = split("\t", $_);
    my $vcf = "$temparray[0]";
    my $out = $temparray[1];
    my $ptSubset = $temparray[2];
    my $genome = $temparray[3];
    my $outcome = $temparray[4];
    my $memory = $temparray[5];
    open JOB1, ">${out}_1.sh" or die$!;
    print JOB1 "\#!/bin/bash\n\
                #SBATCH --time=$walltime\n\
                #SBATCH --nodes=1\n\
                #SBATCH --mem=$memory\n\
                #SBATCH --mail-user=$email\n\
                #SBATCH --ntasks-per-node=4";
    print JOB1 "\n\#SBATCH --mail-type=END\n\
                #SBATCH --partition=general-compute \
                        --qos general-compute\n\
                #SBATCH --job-name=$out\_c1\n\
                #SBATCH --output=$wd/log/\%j\_$out.out\n\
                #SBATCH --error=$wd/log/\%j\_$out.err\n\n\n";
    print JOB1 "\#Get date and time\n\
                tstart=\$(date +\%s)\n\
                echo \"\#\#\#\#\#\# start time:\"`date`\n\n";
    print JOB1 "echo \"SLURM_JOB_ID\"=\$SLURM_JOB_ID\n\
                echo \"SLURM_JOB_NODELIST\"=\$SLURM_JOB_NODELIST\n\
                echo \"SLURM_NNODES\"=\$SLURM_NNODES\n\
                echo \"SLURM_NTASKS\"=\$SLURM_NTASKS\n";
    print JOB1 "echo \"SLURMTMPDIR\"=\$SLURMTMPDIR\n\
                echo \"working directory\"=\$SLURM_SUBMIT_DIR\n\
                echo \"************************\"\n";
    print JOB1 "NPROCs=`srun --nodes=\${SLURM_NNODES} \
                bash -c 'hostname' | wc -l`\n\
                echo NPROCS=\$NPROCS";
```

```
print JOB1 "\nmodule load R\n\
            \nR --file=$rscript -q \
            --args cohort 1 \
            vcf.file $vcf \
            out.file $wd/$out \
            ptSubset $ptSubset \
            genome $genome \
            outcome $outcome \
            ncores \$SLURM_NTASKS\n\
            \n\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\n";
close JOB1;
system "mv ${out}_1.sh $wd/job_scripts/\n";
my $job1 =`sbatch $wd/job_scripts/${out}_1.sh`;
chomp $job1;
my @jobid = split(" ", $job1);
$job1 = $jobid[3];
print "$out Cohort 1 job submitted $job1 \n";
open JOB2, ">${out}_2.sh" or die$!;

print JOB2 "\#!/bin/bash\n\
            #SBATCH --time=$walltime\n\
            #SBATCH --nodes=1\n\
            #SBATCH --mem=$memory\n\
            #SBATCH --mail-user=$email\n\
            #SBATCH --ntasks-per-node=4";
print JOB2 "\n\#SBATCH --mail-type=END\n\
           #SBATCH --partition=general-compute \
                   --qos=general-compute\n\
           #SBATCH --job-name=$out\_c2\n\
           #SBATCH --output=$wd/log/\%j\_$out.out\n\
           #SBATCH --error=$wd/log/\%j\_$out.err\n\n";
print JOB2 "\#Get date and time\n\
            tstart=\$(date +\%s)\n\
            echo \"\#\#\#\#\#\# start time:\"`date`\n\n";
print JOB2 "echo \"SLURM_JOB_ID\"=\$SLURM_JOB_ID\n\
            echo \"SLURM_JOB_NODELIST\"=\$SLURM_JOB_NODELIST\n\
            echo \"SLURM_NNODES\"=\$SLURM_NNODES\n\
            echo \"SLURM_NTASKS\"=\$SLURM_NTASKS\n";
print JOB2 "echo \"SLURMTMPDIR\"=\$SLURMTMPDIR\n\
            echo \"working directory\"=\$SLURM_SUBMIT_DIR\n\n\
            echo \"************************\"\n\n";
print JOB2 "NPROCs=`srun --nodes=\${SLURM_NNODES} bash -c \
```

```
            'hostname' | wc -l`\necho NPROCS=\$NPROCS\n";
print JOB2 "\nmodule load R\n\
            \nR --file=$rscript -q \
            --args cohort 2 \
            vcf.file $vcf \
            out.file $wd/$out
            \ptSubset $ptSubset \
            genome $genome \
            outcome $outcome \
            ncores \$SLURM_NTASKS\n\
            \n\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\n";
close JOB2;

system "mv ${out}_2.sh $wd/job_scripts/\n";
my $job2 =`sbatch $wd/job_scripts/${out}_2.sh`;
chomp $job2;
@jobid = split(" ", $job2);
$job2 = $jobid[3];
print "$out Cohort 2 job submitted $job2 \n";
open JOB3, ">${out}_3.sh" or die$!;
print JOB3 "\#!/bin/bash\n\#SBATCH --time=$walltime\n\
            #SBATCH --nodes=1\n\
            #SBATCH --mem=8000\n\
            #SBATCH --mail-user=$email";
print JOB3 "\n\#SBATCH --mail-type=END\n\
            #SBATCH --partition=general-compute \
                    --qos=general-compute\n\
            #SBATCH --job-name=$out\_meta\n\
            #SBATCH --output=$wd/log/\%j\_$out.out\n\
            #SBATCH --error=$wd/log/\%j\_$out.err\n\n\n";
print JOB3 "\#Get date and time\n\
            tstart=\$(date +\%s)\n
            echo \"\#\#\#\#\#\# start time:\"`date`\n\n";
print JOB3 "echo \"SLURM_JOB_ID\"=\$SLURM_JOB_ID\n\
            echo \"SLURM_JOB_NODELIST\"=\$SLURM_JOB_NODELIST\n\
            echo \"SLURM_NNODES\"=\$SLURM_NNODES\n";
print JOB3 "echo \"SLURMTMPDIR\"=\$SLURMTMPDIR\n\
            echo \"working directory\"=\$SLURM_SUBMIT_DIR\n\n\
            echo \"**********************\"\n\n";
print JOB3 "NPROCs=`srun --nodes=\${SLURM_NNODES} bash \
            -c 'hostname' | wc -l`\necho NPROCS=\$NPROCS\n";
print JOB3 "awk -F \"\\t\" 'OFS=\";\" { print \$1,\$2,\$3,\$4,\$5,\$6}' \
```

```
            $wd/$out\_c1.coxph > $wd/$out\_first6.txt\n";
print JOB3 "awk -F \"\\t\" 'OFS=\"\\t\" {for (i=5; i<NF; i++) printf \$i \
            \"\\t\"; print \$NF}' $wd/$out\_c1.coxph > $wd/$out\_last.txt\n";
print JOB3 "paste -d \"\\t\" $wd/$out\_first6.txt \
            $wd/$out\_last.txt > $wd/$out\_c1.formetal\n";
print JOB3 "awk -F \"\\t\" 'OFS=\";\" { print \$1,\$2,\$3,\$4,\$5,\$6}' \
            $wd/$out\_c2.coxph > $wd/$out\_first6.txt\n";
print JOB3 "awk -F \"\\t\" 'OFS=\"\\t\" {for (i=5; i<NF; i++) printf \$i \"\\t
            print \$NF}' $wd/$out\_c2.coxph > $wd/$out\_last.txt\n";
print JOB3 "paste -d \"\\t\" $wd/$out\_first6.txt $wd/$out\_last.txt > \
            $wd/$out\_c2.formetal\n";
#create the metal_arguments.txt file for this job - these are distinguished
# by the variable $out, so that needs to be unique for each run.
open METAL, ">$wd/job_scripts/metal_scripts/metal_arguments_$out.txt" or die$!
print METAL "\#Meta analysis for cohort 1 and cohort 2\n\
            SCHEME STDERR\n\
            SEPARATOR TAB\n\
            MARKER RSID;TYPED;CHR;POS;REF;ALT\n";
print METAL "ALLELE REF ALT\n\
            EFFECT COEF\n\
            STDERR SE.COEF\n\
            PVALUE PVALUE\n\
            WEIGHT N\n\
            PROCESS $wd/$out\_c1.formetal\n";
print METAL "PROCESS $wd/$out\_c2.formetal\n\
            OUTFILE $wd/$out\_metal_out. .tbl\n\
            MINWEIGHT 10\n\
            ANALYZE HETEROGENEITY\n\n\
            EXIT\n";
close METAL;
#lines to run metal for this $out
print JOB3 "module load metal\n\n";
print JOB3 "metal $wd/job_scripts/metal_scripts/ \
            metal_arguments_$out.txt\n";
#clean up the name of the metal outfile
print JOB3 "cp $wd/$out\_metal_out*tbl $wd/$out.tbl\n";

#end up with $out.tbl, $out_c1.formetal, and $out_c2.formetal - put these to
print JOB3 "module load R\n";
print JOB3 "R --file=/projects/rpci/lsuchest/lsuchest/DBMT_metaPipeline/spread
            -q --args metal_result $wd/$out.tbl \
            for_metal_cohort1 \$wd/$out\_c1.formetal \
```

```
                for_metal_cohort2 $wd/$out\_c2.formetal \
                full_output $wd/$out.res\n\n\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\n
    print JOB3 "rm $wd/$out\_c1_temp.formetal\n";
    print JOB3 "rm $wd/$out\_c2_temp.formetal\n";
    print JOB3 "rm $wd/$out\_first6.txt\n";
    print JOB3 "rm $wd/$out\_last.txt\n";
    print JOB3 "mv $wd/$out\_c1* $wd/temp\n";
    print JOB3 "mv $wd/$out\_c2* $wd/temp\n";
    print JOB3 "mv $wd/$out\_metal* $wd/temp\n";
    print JOB3 "mv $wd/$out.tbl $wd/temp\n";
    print JOB3 "mv $wd/$out.res $wd/out\n";
    close JOB3;
    system "mv ${out}_3.sh $wd/job_scripts/\n";
    my $job3 =`sbatch --dependency=afterok:${job1}:${job2} $wd/job_scripts/${out}_
    chomp $job3;
        @jobid = split(" ", $job3);
        $job3 = $jobid[3];
    print "$out METAL script for cohorts 1 and 2 submitted $job3 \n";
```

Execution of script

```
perl /projects/rpci/lsuchest/lsuchest/DBMT_metaPipeline/Meta_pipeline.pl
```

Required commands:

```
-m        path to the manifest file
-e        e-mail address for messaging
-w        walltime in format 00:00:00
-rscript  /projects/rpci/lsuchest/lsuchest/DBMT_metaPipeline/\
           run_gwasurvivr.R (or path to another user defined R script)
```

Manifest file

Output Folders

Utilization and purpose

Conclusion

CHAPTER 5: Acute Lymphobastic Leukemia (ALL) GWAS

ALL: Schematic of Pipeline

It doesn't say anything at the beginning. Could this be the problem

ALL stuff

Run ALL only and B-ALL only

Discovery

Interpretation

Discussion

CHAPTER 6: Conclusion and Future Work

+100 Days Schematic of Pipeline

Wow maybe the problem is that it didn't say anything. # Future Work

Conclusion

Stuff is written here …

REFERENCES

1000 Genomes Project Consortium. 2015. "A Global Reference for Human Genetic Variation." *Nature* 526 (7571). Nature Publishing Group: 68.

1000 Genomes Project Consortium, Gonçalo R Abecasis, David Altshuler, Adam Auton, Lisa D Brooks, Richard M Durbin, Richard A Gibbs, Matt E Hurles, and Gil A McVean. 2010. "A Map of Human Genome Variation from Population-Scale Sequencing." *Nature* 467 (7319): 1061–73. doi:10.1038/nature09534.

Arnold, Matthias, Johannes Raffler, Arne Pfeufer, Karsten Suhre, and Gabi Kastenmüller. 2015. "SNiPA: An Interactive, Genetic Variant-Centered Annotation Browser." *Bioinformatics* 31 (8): 1334–6. doi:10.1093/bioinformatics/btu779.

Arrieta-Bolaños, Esteban, Neema P Mayor, Steven G E Marsh, J Alejandro Madrigal, Jane F Apperley, Keiren Kirkland, Stephen Mackinnon, et al. 2016. "Polymorphism in Tgfb1 Is Associated with Worse Non-Relapse Mortality and Overall Survival After Stem Cell Transplantation with Unrelated Donors." *Haematologica* 101 (3): 382–90. doi:10.3324/haematol.2015.134999.

Bakker, Paul I W de, Manuel A R Ferreira, Xiaoming Jia, Benjamin M Neale, Soumya Raychaudhuri, and Benjamin F Voight. 2008. "Practical Aspects of Imputation-Driven Meta-Analysis of Genome-Wide Association Studies." *Hum Mol Genet* 17 (R2): R122–8. doi:10.1093/hmg/ddn288.

Bakker, Paul I W de, Roman Yelensky, Itsik Pe'er, Stacey B Gabriel, Mark J Daly, and David Altshuler. 2005. "Efficiency and Power in Genetic Association Studies." *Nat Genet* 37 (11): 1217–23. doi:10.1038/ng1669.

Balavarca, Y, K Pearce, J Norden, M Collin, G Jackson, E Holler, R Dressel, et al. 2015. "Predicting Survival Using Clinical Risk Scores and Non-Hla Immunogenetics." *Bone Marrow Transplant* 50 (11): 1445–52. doi:10.1038/bmt.2015.173.

Botstein, David, and Neil Risch. 2003. "Discovering Genotypes Underlying Human Phenotypes: Past Successes for Mendelian Disease, Future Approaches for

Complex Disease." *Nat Genet* 33 Suppl (March): 228–37. doi:10.1038/ng1090.

Boyle, Alan P, Eurie L Hong, Manoj Hariharan, Yong Cheng, Marc A Schaub, Maya Kasowski, Konrad J Karczewski, et al. 2012. "Annotation of Functional Variation in Personal Genomes Using Regulomedb." *Genome Research* 22 (9): 1790–7. doi:10.1101/gr.137323.112.

Breslow, Norman. 1974. "Covariance Analysis of Censored Survival Data." *Biometrics*. JSTOR, 89–99.

Browning, Brian L, and Sharon R Browning. 2016. "Genotype Imputation with Millions of Reference Samples." *Am J Hum Genet* 98 (1): 116–26. doi:10.1016/j.ajhg.2015.11.020.

Cavet, J, A M Dickinson, J Norden, P R Taylor, G H Jackson, and P G Middleton. 2001. "Interferon-Gamma and Interleukin-6 Gene Polymorphisms Associate with Graft-Versus-Host Disease in Hla-Matched Sibling Bone Marrow Transplantation." *Blood* 98 (5): 1594–1600.

Chang, Ya-Yi, Hildegard T Greinix, Anne M Dickinson, Daniel Wolff, Graham H Jackson, Reinhard Andreesen, Ernst Holler, and Gerhard C Hildebrandt. 2009. "G to c Transition at Position -173 of Mif Gene of the Recipient Is Associated with Reduced Relapse Rates After Allogeneic Stem Cell Transplantation." *Cytokine* 48 (3): 218–25. doi:10.1016/j.cyto.2009.07.012.

Chien, Jason W, Xinyi Cindy Zhang, Wenhong Fan, Hongwei Wang, Lue Ping Zhao, Paul J Martin, Barry E Storer, Michael Boeckh, Edus H Warren, and John A Hansen. 2012. "Evaluation of Published Single Nucleotide Polymorphisms Associated with Acute Gvhd." *Blood* 119 (22): 5311–9. doi:10.1182/blood-2011-09-371153.

Clay-Gilmour, Alyssa I., Theresa Hahn, Leah M. Preus, Kenan Onel, Andrew Skol, Eric Hungate, Qianqian Zhu, et al. 2017. "Genetic Association with B-Cell Acute Lymphoblastic Leukemia in Allogeneic Transplant Patients Differs by Age and Sex." *Blood Advances* 1 (20): 1717–28.

Colhoun, Helen M, Paul M McKeigue, and George Davey Smith. 2003. "Problems of Reporting Genetic Associations with Complex Outcomes." *The Lancet* 361 (9360). Elsevier: 865–72.

Copelan, Edward A. 2006. "Hematopoietic Stem-Cell Transplantation." *N Engl J Med* 354 (17): 1813–26. doi:10.1056/NEJMra052638.

Cox, David R. 1972. "Regression Models and Life-Tables." *Journal of the Royal Sta-*

tistical Society: Series B (Methodological) 34 (2). Wiley Online Library: 187–202.

Das, Sayantan, Lukas Forer, Sebastian Schönherr, Carlo Sidore, Adam E Locke, Alan Kwong, Scott I Vrieze, et al. 2016. "Next-Generation Genotype Imputation Service and Methods." *Nature Genetics* 48 (10). Nature Publishing Group: 1284.

Durbin, Richard. 2014. "Efficient Haplotype Matching and Storage Using the Positional Burrows-Wheeler Transform (Pbwt)." *Bioinformatics* 30 (9): 1266–72. doi:10.1093/bioinformatics/btu014.

D'Souza, Anita, Stephanie Lee, Xiaochun Zhu, and Marcelo Pasquini. 2017. "Current Use and Trends in Hematopoietic Cell Transplantation in the United States." *Biology of Blood and Marrow Transplantation* 23 (9): 1417–21. doi:10.1016/j.bbmt.2017.05.035.

Efron, Bradley. 1977. "The Efficiency of Cox's Likelihood Function for Censored Data." *Journal of the American Statistical Association* 72 (359). Taylor & Francis: 557–65.

Fine, Jason P, and Robert J Gray. 1999. "A Proportional Hazards Model for the Subdistribution of a Competing Risk." *Journal of the American Statistical Association* 94 (446). Taylor & Francis: 496–509.

Gallagher, Michael D, and Alice S Chen-Plotkin. 2018. "The Post-Gwas Era: From Association to Function." *Am J Hum Genet* 102 (5): 717–30. doi:10.1016/j.ajhg.2018.04.002.

Gogarten, Stephanie M, Tushar Bhangale, Matthew P Conomos, Cecelia A Laurie, Caitlin P McHugh, Ian Painter, Xiuwen Zheng, et al. 2012. "GWASTools: An R/Bioconductor Package for Quality Control and Analysis of Genome-Wide Association Studies." *Bioinformatics* 28 (24). Oxford University Press: 3329–31.

Grambsch, Patricia M, and Terry M Therneau. 1994. "Proportional Hazards Tests and Diagnostics Based on Weighted Residuals." *Biometrika* 81 (3). Oxford University Press: 515–26.

Green, Richard E, Benjamin P Lewis, R Tyler Hillman, Marco Blanchette, Liana F Lareau, Aaron T Garnett, Donald C Rio, and Steven E Brenner. 2003. "Widespread Predicted Nonsense-Mediated mRNA Decay of Alternatively-Spliced Transcripts of Human Normal and Disease Genes." *Bioinformatics* 19

Suppl 1: i118–21.

GTEx Consortium. 2013. "The Genotype-Tissue Expression (Gtex) Project." *Nat Genet* 45 (6): 580–5. doi:10.1038/ng.2653.

Hahn, Theresa, Lara E Sucheston-Campbell, Leah Preus, Xiaochun Zhu, John A Hansen, Paul J Martin, Li Yan, et al. 2015. "Establishment of Definitions and Review Process for Consistent Adjudication of Cause-Specific Mortality After Allogeneic Unrelated-Donor Hematopoietic Cell Transplantation." *Biology of Blood and Marrow Transplantation* 21 (9): 1679–86. doi:10.1016/j.bbmt.2015.05.019.

Hamilton, Betty K, and Edward A Copelan. 2012. "Concise Review: The Role of Hematopoietic Stem Cell Transplantation in the Treatment of Acute Myeloid Leukemia." *Stem Cells* 30 (8): 1581–6. doi:10.1002/stem.1140.

Henig, Israel, and Tsila Zuckerman. 2014. "Hematopoietic Stem Cell Transplantation-50 Years of Evolution and Future Perspectives." *Rambam Maimonides Medical Journal* 5 (4): e0028. doi:10.5041/RMMJ.10162.

Holler, E, G Rogler, J Brenmoehl, J Hahn, H Greinix, A M Dickinson, G Socie, et al. 2008. "The Role of Genetic Variants of Nod2/Card15, a Receptor of the Innate Immune System, in Gvhd and Complications Following Related and Unrelated Donor Haematopoietic Stem Cell Transplantation." *Int J Immunogenet* 35 (4-5): 381–4. doi:10.1111/j.1744-313X.2008.00795.x.

Holler, Ernst, Gerhard Rogler, Hans Herfarth, Julia Brenmoehl, Peter Johannes Wild, Joachim Hahn, Günther Eissner, Jürgen Schölmerich, and Reinhard Andreesen. 2004. "Both Donor and Recipient Nod2/Card15 Mutations Associate with Transplant-Related Mortality and Gvhd Following Allogeneic Stem Cell Transplantation." *Blood* 104 (3): 889–94. doi:10.1182/blood-2003-10-3543.

Howie, Bryan N, Peter Donnelly, and Jonathan Marchini. 2009. "A Flexible and Accurate Genotype Imputation Method for the Next Generation of Genome-Wide Association Studies." *PLoS Genetics* 5 (6): e1000529. doi:10.1371/journal.pgen.1000529.

Igl, Bernd-Wolfgang, Inke R Konig, and Andreas Ziegler. 2009. "What Do We Mean by 'Replication' and 'Validation' in Genome-Wide Association Studies?" *Human Heredity* 67 (1): 66–68. doi:10.1159/000164400.

International HapMap Consortium. 2005. "A Haplotype Map of the Human

Genome." *Nature* 437 (7063): 1299–1320. doi:10.1038/nature04226.

International HapMap Consortium, Kelly A Frazer, Dennis G Ballinger, David R Cox, David A Hinds, Laura L Stuve, Richard A Gibbs, et al. 2007. "A Second Generation Human Haplotype Map of over 3.1 Million Snps." *Nature* 449 (7164): 851–61. doi:10.1038/nature06258.

Ioannidis, J P, E E Ntzani, T A Trikalinos, and D G Contopoulos-Ioannidis. 2001. "Replication Validity of Genetic Association Studies." *Nature Genetics* 29 (3): 306–9. doi:10.1038/ng749.

Jindra, Peter T, Susan E Conway, Stacy M Ricklefs, Stephen F Porcella, Sarah L Anzick, Mike Haagenson, Tao Wang, et al. 2016. "Analysis of a Genetic Polymorphism in the Costimulatory Molecule Tnfsf4 with Hematopoietic Stem Cell Transplant Outcomes." *Biol Blood Marrow Transplant* 22 (1): 27–36. doi:10.1016/j.bbmt.2015.08.037.

Jorde, L B. 2000. "Linkage Disequilibrium and the Search for Complex Disease Genes." *Genome Res* 10 (10): 1435–44.

Karaesmen, Ezgi, Abbas A. Rizvi, Leah M. Preus, Philip L. McCarthy, Marcelo C. Pasquini, Kenan Onel, Xiaochun Zhu, et al. 2017. "Replication and Validation of Genetic Polymorphisms Associated with Survival After Allogeneic Blood or Marrow Transplant." *Blood* 130 (13). American Society of Hematology: 1585–96. doi:10.1182/blood-2017-05-784637.

Kornblit, Brian, Tania Masmas, Søren L Petersen, Hans O Madsen, Carsten Heilmann, Lone Schejbel, Henrik Sengeløv, Klaus Müller, Peter Garred, and Lars Vindeløv. 2010. "Association of Hmgb1 Polymorphisms with Outcome After Allogeneic Hematopoietic Cell Transplantation." *Biol Blood Marrow Transplant* 16 (2): 239–52. doi:10.1016/j.bbmt.2009.10.002.

Lander, E S, L M Linton, B Birren, C Nusbaum, M C Zody, J Baldwin, K Devon, et al. 2001. "Initial Sequencing and Analysis of the Human Genome." *Nature* 409 (6822): 860–921. doi:10.1038/35057062.

Laurie, Cathy C, Kimberly F Doheny, Daniel B Mirel, Elizabeth W Pugh, Laura J Bierut, Tushar Bhangale, Frederick Boehm, et al. 2010. "Quality Control and Quality Assurance in Genotypic Data for Genome-Wide Association Studies." *Genet Epidemiol* 34 (6): 591–602. doi:10.1002/gepi.20516.

Lemieux Perreault, Louis-Philippe, Marc-André Legault, Géraldine Asselin, and Marie-Pierre Dube. 2016. "Genipe: An Automated Genome-Wide Imputation Pipeline with Automatic Reporting and Statistical Tools." *Bioinformatics*

32 (23). Oxford University Press: 3661–3.

Lewis, Cathryn M, and Jo Knight. 2012. "Introduction to Genetic Association Studies." *Cold Spring Harb Protoc* 2012 (3): 297–306. doi:10.1101/pdb.top068163.

Li, Na, and Matthew Stephens. 2003. "Modeling Linkage Disequilibrium and Identifying Recombination Hotspots Using Single-Nucleotide Polymorphism Data." *Genetics* 165 (4): 2213–33.

Li, Yun, Cristen J Willer, Jun Ding, Paul Scheet, and Gonçalo R Abecasis. 2010. "MaCH: Using Sequence and Genotype Data to Estimate Haplotypes and Unobserved Genotypes." *Genet Epidemiol* 34 (8): 816–34. doi:10.1002/gepi.20533.

Marchini, Jonathan, and Bryan Howie. 2010. "Genotype Imputation for Genome-Wide Association Studies." *Nat Rev Genet* 11 (7): 499–511. doi:10.1038/nrg2796.

Marchini, Jonathan, Bryan Howie, Simon Myers, Gil McVean, and Peter Donnelly. 2007. "A New Multipoint Method for Genome-Wide Association Studies by Imputation of Genotypes." *Nature Genetics* 39 (7). Nature Publishing Group: 906.

Martin, Paul J, Wenhong Fan, Barry E Storer, David M Levine, Lue Ping Zhao, Edus H Warren, Mary E D Flowers, et al. 2016. "Replication of Associations Between Genetic Polymorphisms and Chronic Graft-Versus-Host Disease." *Blood* 128 (20): 2450–6. doi:10.1182/blood-2016-07-728063.

McCarthy, Shane, Sayantan Das, Warren Kretzschmar, Olivier Delaneau, Andrew R Wood, Alexander Teumer, Hyun Min Kang, et al. 2016. "A Reference Panel of 64,976 Haplotypes for Genotype Imputation." *Nature Genetics* 48 (10). Nature Publishing Group: 1279.

McDermott, David H, Susan E Conway, Tao Wang, Stacy M Ricklefs, Manza A Agovi, Stephen F Porcella, Huong Thi Bich Tran, Edgar Milford, Stephen Spellman, and Reza Abdi. 2010. "Donor and Recipient Chemokine Receptor Ccr5 Genotype Is Associated with Survival After Bone Marrow Transplantation." *Blood* 115 (11): 2311–8. doi:10.1182/blood-2009-08-237768.

McLaren, William, Bethan Pritchard, Daniel Rios, Yuan Chen, Paul Flicek, and Fiona Cunningham. 2010. "Deriving the Consequences of Genomic Variants with the Ensembl Api and Snp Effect Predictor." *Bioinformatics* 26 (16):

2069–70. doi:10.1093/bioinformatics/btq330.

Mishra, Aniket, and Stuart Macgregor. 2015. "VEGAS2: Software for More Flexible Gene-Based Testing." *Twin Research and Humans Genetics* 18 (1): 86–91. doi:10.1017/thg.2014.79.

Obenchain, Valerie, Michael Lawrence, Vincent Carey, Stephanie Gogarten, Paul Shannon, and Martin Morgan. 2014. "VariantAnnotation: A Bioconductor Package for Exploration and Annotation of Genetic Variants." *Bioinformatics* 30 (14). Oxford University Press: 2076.

Pasquini, Marcelo, Zhiwei Wang, Mary M Horowitz, and Robert Peter Gale. 2013. "2013 Report from the Center for International Blood and Marrow Transplant Research (Cibmtr): Current Uses and Outcomes of Hematopoietic Cell Transplants for Blood and Bone Marrow Disorders." *Clinical Transplants*, 187–97.

Pérez-García, Arianne, Rafael De la Cámara, Jose Román-Gómez, Antonio Jiménez-Velasco, Maite Encuentra, Jose B Nieto, Javier de la Rubia, et al. 2007. "CTLA-4 Polymorphisms and Clinical Outcome After Allogeneic Stem Cell Transplantation from Hla-Identical Sibling Donors." *Blood* 110 (1): 461–7. doi:10.1182/blood-2007-01-069781.

Price, Alkes L, Nick J Patterson, Robert M Plenge, Michael E Weinblatt, Nancy A Shadick, and David Reich. 2006. "Principal Components Analysis Corrects for Stratification in Genome-Wide Association Studies." *Nature Genetics* 38 (8): 904–9. doi:10.1038/ng1847.

Pritchard, J K, and M Przeworski. 2001. "Linkage Disequilibrium in Humans: Models and Data." *Am J Hum Genet* 69 (1): 1–14. doi:10.1086/321275.

Purcell, Shaun, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel AR Ferreira, David Bender, Julian Maller, et al. 2007. "PLINK: A Tool Set for Whole-Genome Association and Population-Based Linkage Analyses." *The American Journal of Human Genetics* 81 (3). Elsevier: 559–75.

R Core Team. 2018. "R: A Language and Environment for Statistical Computing." Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Reich, D E, M Cargill, S Bolk, J Ireland, P C Sabeti, D J Richter, T Lavery, et al. 2001. "Linkage Disequilibrium in the Human Genome." *Nature* 411 (6834): 199–204. doi:10.1038/35075590.

Rizvi, Abbas A, Ezgi Karaesmen, Martin Morgan, Leah Preus, Junke Wang, Michael

Sovic, Theresa Hahn, and Lara E Sucheston-Campbell. 2018. "Gwasurvivr: An R Package for Genome Wide Survival Analysis." *Bioinformatics*, November. doi:10.1093/bioinformatics/bty920.

Schoenfeld, David. 1982. "Partial Residuals for the Proportional Hazards Regression Model." *Biometrika* 69 (1). Oxford University Press: 239–41.

Schwarzer, Guido. 2007. "Meta: An R Package for Meta-Analysis." *R News* 7 (3): 40–45.

Shi, Zhen, and John Moult. 2011. "Structural and Functional Impact of Cancer-Related Missense Somatic Mutations." *J Mol Biol* 413 (2): 495–512. doi:10.1016/j.jmb.2011.06.046.

Su, Zhan, Jonathan Marchini, and Peter Donnelly. 2011. "HAPGEN2: Simulation of Multiple Disease Snps." *Bioinformatics* 27 (16). Oxford University Press: 2304–5.

Sucheston-Campbell, Lara E, Alyssa Clay, Philip L McCarthy, Qianqian Zhu, Leah Preus, Marcelo Pasquini, Kenan Onel, and Theresa Hahn. 2015. "Identification and Utilization of Donor and Recipient Genetic Variants to Predict Survival After Hct: Are We Ready for Primetime?" *Current Hematologic Malignancy Reports* 10 (1): 45–58. doi:10.1007/s11899-014-0246-x.

Sudmant, Peter H, Tobias Rausch, Eugene J Gardner, Robert E Handsaker, Alexej Abyzov, John Huddleston, Yan Zhang, et al. 2015. "An Integrated Map of Structural Variation in 2,504 Human Genomes." *Nature* 526 (7571): 75–81. doi:10.1038/nature15394.

Syed, Hamzah, Andrea L Jorgensen, and Andrew P Morris. 2017. "Survival-GWAS_SV: Software for the Analysis of Genome-Wide Association Studies of Imputed Genotypes with 'Time-to-Event' Outcomes." *BMC Bioinformatics* 18 (1). BioMed Central: 265.

Therneau, Terry M, and Patricia M Grambsch. 2013. *Modeling Survival Data: Extending the Cox Model*. Springer Science & Business Media.

Timpson, Nicholas J, Celia M T Greenwood, Nicole Soranzo, Daniel J Lawson, and J Brent Richards. 2018. "Genetic Architecture: The Shape of the Genetic Contribution to Human Traits and Disease." *Nat Rev Genet* 19 (2): 110–24. doi:10.1038/nrg.2017.101.

Venables, W. N., and B. D. Ripley. 2002. *Modern Applied Statistics with S*. Fourth.

New York: Springer. http://www.stats.ox.ac.uk/pub/MASS4.

Visscher, Peter M, Matthew A Brown, Mark I McCarthy, and Jian Yang. 2012. "Five Years of Gwas Discovery." *Am J Hum Genet* 90 (1): 7–24. doi:10.1016/j.ajhg.2011.11.029.

Westra, Harm-Jan, Marjolein J Peters, Tõnu Esko, Hanieh Yaghootkar, Claudia Schurmann, Johannes Kettunen, Mark W Christiansen, et al. 2013. "Systematic Identification of Trans eQTLs as Putative Drivers of Known Disease Associations." *Nat Genet* 45 (10): 1238–43. doi:10.1038/ng.2756.

Willer, Cristen J, Yun Li, and Gonçalo R Abecasis. 2010. "METAL: Fast and Efficient Meta-Analysis of Genomewide Association Scans." *Bioinformatics* 26 (17): 2190–1. doi:10.1093/bioinformatics/btq340.

Yan, Li, Changxing Ma, Dan Wang, Qiang Hu, Maochun Qin, Jeffrey M Conroy, Lara E Sucheston, et al. 2012. "OSAT: A Tool for Sample-to-Batch Allocations in Genomics Experiments." *BMC Genomics* 13 (December): 689. doi:10.1186/1471-2164-13-689.

Yee, L J, K Im, B Borg, H Yang, T J Liang, and Virahep-C Study. 2009. "Interleukin-6 Haplotypes and the Response to Therapy of Chronic Hepatitis c Virus Infection." *Genes Immun* 10 (4): 365–72. doi:10.1038/gene.2009.26.

APPENDICES

Chapter 2

Candidate Gene Analyses

We conducted an exhaustive literature search looking at papers from 71 candidate gene studies that were found via Pubmed search. The candidate gene studies focused on any blood disorders as long as one of the disorders was either ALL or AML. The candidate gene studies typically looked at several candidate SNPs in candidate genes and determined if there was an association with the SNP and survival outcomes (DRM, OS, PFS, or TRM). We gathered all of the SNPs that were studied in these candidate gene studies (even if they were not significant in the original study).

The final table can be found on UB CCR at:
`/projects/rpci/lsuchest/abbasriz/candidate_gene/ \`
`result_files_cg/final_table/cg_snptable.txt`.

The ultimate goal of this study was to test all of the candidate SNPs in DISCOVeRY-BMT to replicate or validate the original studies' findings. Replication means that the phenotype and population are the same. Validation means that the phenotype is the same but the study population is not necessarily the same (i.e. different ethnic group). And also, because these studies essentially had gene based hypotheses, we sought to test DISCOVeRY-BMT survival data

using gene-based statistical approaches (VEGAS2 software) to test whether the aggregation of the entire gene locus was significant or not.

Here we present our data analysis.

```
# grab gene list in unix shell from final table and unique it
[abbasriz@rush:]$ </projects/rpci/lsuchest/abbasriz/candidate_gene/ \
    result_files_cg/final_table/cg_snptable.txt \
    tr -s ' ' ' ' | \
    awk '{print $1}' | \
    uniq > gene_list.txt
```

```
# load gene list skip column header 'gene'
# this list has genes from chrX already
# removed
genes.w.locs <- read.table("/projects/rpci/lsuchest/lsuchest/
                           CandidateGeneReplication/
                          final_gene_list/
                          gene_locations_20170222.txt",
    header = TRUE, stringsAsFactors = FALSE)
# order by chromosome and sort by numeric part
# of the character vector so they are ordered
# correctly
vals <- as.numeric(gsub("chr", "", genes.w.locs$seqnames))
genes.w.locs <- genes.w.locs[order(vals), ]
head(genes.w.locs)
```

Final Candidate Gene List

```
gene.list <- as.character(sort(genes.w.locs$gene_symbol))
write.table(gene.list, file = "final_cg_genelist.txt",
    quote = FALSE, sep = "\t", row.names = FALSE,
    col.names = FALSE)
```

We need to grab the typed and imputed SNPs from DISCOVeRY-BMT. These are located in:

`/projects/rpci/lsuchest/lsuchest/Rserve/ImputeData/var/db/gwas/ \`

`imputed_data/BMT093013_forImpute/Impute2_summary/Impute2.INFO`

file.

```
[abbasriz@rush:]$ head /projects/rpci/lsuchest/lsuchest/Rserve/ImputeData/var/db/gwas/ \
imputed_data/BMT093013_forImpute/Impute2_summary/Impute2.INFO
region snp_id rs_id position exp_freq_a1 info certainty type info_type0 concord_type0 r2_type0
```

```
chr10-0-5000000 --- rs148087467 60523 0.002 0.686 0.998 0 -1 -1 -1
chr10-0-5000000 --- rs187110906 60969 0.092 0.499 0.891 0 -1 -1 -1
chr10-0-5000000 --- rs192025213 61005 0.003 0.371 0.996 0 -1 -1 -1
chr10-0-5000000 --- rs115033199 61020 0.000 0.341 0.999 0 -1 -1 -1
chr10-0-5000000 --- rs183305313 61334 0.005 0.302 0.990 0 -1 -1 -1
chr10-0-5000000 --- rs186558141 65978 0.000 0.114 1.000 0 -1 -1 -1
chr10-0-5000000 --- rs190079063 66269 0.000 0.137 1.000 0 -1 -1 -1
chr10-0-5000000 --- rs12260013 66326 0.030 0.649 0.972 0 -1 -1 -1
chr10-0-5000000 --- chr10:66627:D 66627 0.548 0.590 0.734 0 -1 -1 -1
```

Now that we have the gene locations, we are going to create a shell script that will be able to grab the `Impute2.INFO` file regions and let us collect all of the SNPs that we have from DISCOVeRY-BMT that are typed and imputed. The shell script can be found at:

`/projects/rpci/lsuchest/lsuchest/CandidateGeneReplication/ \`
`parse_Impute2INFO/awk_Impute2.INFO_by_geneLoc_cg.sh`

This shell script rearranges the columns to the following: `chr`, `position`, `gene`, `snp_id`, `rs_id`, `exp_freq_q1`, `info`, `certainty`, `type`, `info_type0`, `concord_type0`, `r2_type0`.

```
capture.output(file = "/projects/rpci/lsuchest/lsuchest/CandidateGeneReplication/
              parse_Impute2INFO/awk_Impute2.INFO_by_geneLoc_cg.sh",
    {
        for (i in 1:nrow(genes.w.locs)) {
            cat("awk '{if ($1 ~ /", genes.w.locs$seqnames[i],
                "-/ && $4 > ", genes.w.locs$start_position[i],
                " && $4 < ", genes.w.locs$end_position[i],
                ") print ", "\"", genes.w.locs$seqnames[i],
                "\" \"\\t\" $4 \"\\t\" \"", genes.w.locs$gene_symbol[i],
                "\"", " \"\\t\" $2", " \"\\t\" $3",
                " \"\\t\" $5", " \"\\t\" $6",
                " \"\\t\" $7", " \"\\t\" $8",
                " \"\\t\" $9", " \"\\t\" $10",
                " \"\\t\" $11}' ", "/projects/rpci/lsuchest/lsuchest/Rserve/ImputeData/var/db/
            gwas/imputed_data/BMT093013_forImpute/Impute2_summary/Impute2.INFO >>
            /projects/rpci/lsuchest/abbasriz/candidate_gene/info.cg.txt",
                "\n", sep = "")
        }
    })
```

We edited this to make this into a SLURM script would and after running we have a file `info.cg.txt`.

```
[abbasriz@rush:/projects/rpci/lsuchest/abbasriz/candidate_gene]$ head -3 info.cg.txt
chr1    70866967    CTH --- rs77482262  0.069   0.984   0.997   0   -1  -1  -1
chr1    70866988    CTH --- rs187366946 0.002   0.548   0.997   0   -1  -1  -1
chr1    70867130    CTH --- rs190937437 0.001   0.208   0.999   0   -1  -1  -1
```

`info.cg.txt` does not have a header so are going to add the header in R and re-write the file (alternatively, we could have just assigned this in each of the survival parsing R scripts that we are about to make.)

```
library(data.table)
info <- fread("/projects/rpci/lsuchest/abbasriz/candidate_gene/info.cg.txt")
colnames(info) <- c("chr", "position", "gene",
    "snp_id", "rs_id", "exp_freq_a1", "info",
    "certainty", "type", "info_type0", "concord_type0",
    "r2_type0")
write.table(info, file = "/projects/rpci/lsuchest/abbasriz/candidate_gene/info.cg.txt",
    quote = FALSE, col.names = TRUE, row.names = FALSE,
    sep = "\t")
```

Survival Results Directories

Individual (donor/recipient genotyping cohorts 1 and 2) and shared (mismatch between donor-recipient pairs from cohorts 1 and 2), done for 4 different outcomes (DRM, PFS, OS, TRM) and 4 different disease groups (AMLonly, ALLonly, mixed, noALL), and their corresponding meta analyses (combining cohort 1 and 2 using a fixed effect model from METAL software) are located in different directories on UB CCR. 3 genomes (donor, recipient, shared) x 3 cohorts (c1, c2, meta) x 4 outcomes (DRM, PFS, OS, TRM) x 4 diseases (AML, ALL, mixed, noALL) = 144 analyses.

Individual directory:

`/projects/rpci/lsuchest/lsuchest/Rserve/ImputeData/var/db/gwas/ \`

`imputed_data/BMT093013_forImpute/analyses/`

Meta-individual directory:

/projects/rpci/lsuchest/lsuchest/Rserve/ImputeData/var/db/gwas/ \
imputed_data/BMT093013_forImpute/analyses/METAL.results/

Shared directory:

/projects/rpci/lsuchest/lsuchest/Rserve/ImputeData/var/db/gwas/ \
imputed_data/SHARED/analyses/METAL.RESULTS.SHARED/

Meta-shared directory:

/projects/rpci/lsuchest/lsuchest/Rserve/ImputeData/var/db/gwas/ \
imputed_data/BMT093013_forImpute/analyses/METAL.results/

As a collective, these directories contain files of 144 analyses. However, the result files are not so well organized. Using some unix commands to capture a clean amount of directories and files, as well as *ad hoc* manual curation, final directory lists (/projects/rpci/lsuchest/lsuchest/CandidateGeneReplication/ \ survival_results_directories/) can be found in the following files: ind.directories.txt, shared.directories.txt, meta.ind.directories.txt, and meta.shared.directories.txt.

Parsing Result Files for High Quality Candidate Gene SNPs

We have survival results located in these directories, now we want to subset each of these survival results for just all of the high quality SNPs in our candidate genes. Here I will show only 1 of 4 (independent_results.R) parsing results. The others are meta_independent.results.R, shared_results.R, and meta_results.R. I split these up to decrease computational time and to have some safety checks at a smaller scale.

The result files go into the directory: /projects/rpci/lsuchest/abbasriz/\

`candidate_gene/result_files_cg/impute.results.w.typedsnps`

The result files will be in the following format:

`genome_cohort_outcome_disease.txt`, e.g. (`D_c1_DRM_ALLonly.txt`, for donor, cohort 1, death to due to disease, ALL only subset)

```
## INDEPENDENT RESULTS
library(data.table)
# read in candidate gene
info <- fread("/projects/rpci/lsuchest/abbasriz/candidate_gene/info.cg.txt",
    header = "auto", sep = "\t")
# colnames(info) <- c('chr', 'position',
# 'gene', 'snp_id', 'rs_id', 'exp_freq_a1',
# 'info', 'certainty', 'type', 'info_type0',
# 'concord_type0', 'r2_type0')
setkey(info, "gene", "snp_id", "rs_id")
info <- info[, c("gene", "snp_id", "rs_id")]
setkey(info, "snp_id", "rs_id")
ind <- scan("/projects/rpci/lsuchest/abbasriz/candidate_gene/
            result_files_cg/res.directories/ind.directories.txt",
    what = character())
for (i in 1:length(ind)) {
    ind.res <- fread(ind[i], header = "auto",
        sep = "\t", verbose = TRUE)
    colnames(ind.res)[1:2] <- c("snp_id", "rs_id")
    setkey(ind.res, "snp_id", "rs_id")
    ind.res <- ind.res[info]
    ind.res <- na.omit(ind.res)
    ind.res[, `:=`(c("z", "loglik0", "loglik"),
        NULL)]
    ind.res[, `:=`(`95%-CI`, NA)]
    setcolorder(ind.res, c("gene", "rs_id", "CHR",
        "BP", "ALLELE1", "ALLELE2", "n", "coef",
        "se(coef)", "exp(coef)", "95%-CI", "snp_id",
        "Pr(>|z|)"))
    colnames(ind.res) <- c("gene", "rsID", "chr",
        "BP", "allele1", "allele2", "N", "coef",
        "se.coef", "exp.coef", "95%-CI", "impute",
        "Pvalue")
    file.name <- strsplit(ind[i], split = "/")[[1]][14]
    file.name <- gsub("[.]", "_", file.name)
    file.name <- strsplit(file.name, "_")[[1]]
    if (length(file.name) < 6) {
        file.name[6] <- "mixed"
    } else {
        file.name <- file.name
    }
    write.table(ind.res, file = paste0("/projects/rpci/lsuchest/abbasriz/candidate_gene/
\t                                  results_files_cg/impute.results.w.typedsnps/",
        paste(file.name[1], file.name[2], file.name[4],
            file.name[6], sep = "_"), ".txt"),
        quote = FALSE, sep = "\t", col.names = TRUE,
        row.names = FALSE)
    rm(ind.res)
}
```

Note: Don't run the following shell script without editing the R script above (and

similar ones) on UB CCR, as the directory may not be correct. The following shell script (`/projects/rpci/lsuchest/abbasriz/candidate_gene/ \ result_files_cg/res.directories/independent_results.sh`) was written to run this command using SLURM:

```
#!/bin/bash
#SBATCH --time=24:00:00
#SBATCH --nodes=1
#SBATCH --mem-per-cpu=5000
#SBATCH --job-name=myjobname
#SBATCH --output=myjob.out
#SBATCH --error=myjoberr.err
#SBATCH --partition=general-compute
#SBATCH --mail-user=abbasriz@buffalo.edu
#SBATCH --mail-type=ALL
#SBATCH --mail-type=END

#Get date and time
tstart=`date`
echo "###### start time:"$tstart
cd /projects/rpci/lsuchest/abbasriz/candidate_gene/\
    result_files_cg/res.directories/
echo "Run program"
module load R
R CMD BATCH independent_results.R
echo "program finished"
echo "All Done!"
tend=`date`
echo "###### end time: $tend"
```

Again, the 144 result files can be found in: `/projects/rpci/lsuchest/abbasriz/\ candidate_gene/result_files_cg/impute.results.w.typedsnps/`

Remove Duplicates

METAL and VEGAS2 ignore duplicates rsids. In DISCOVeRY-BMT often has duplicates rsids due to some SNPs being typed AND imputed. We filtered these files to removed the typed SNPs, as the imputed SNP may be more reliable as it is calculated using the reference genome. Again, we separated this into 3 jobs, divvying up the results by donor, recipient, and shared. Instead of using two columns to remove duplicates, we removed duplicates by searching for duplicates

and removing the one with lower base pair (i.e. position 111110 would be typed and position 111111 would be imputed, so we would remove position 111110 in this case). Here we demonstrate this using only the donor files (donor_remove_dups.R and donor_remove_dups.sh)

```
# donor_remove_dups.R
library(data.table)
donor.files <- list.files(pattern = "^D_")
for (i in 1:length(donor.files)) {
    donor <- fread(donor.files[i], header = "auto",
        sep = "\t")
    setkey(donor, chr, BP, rsID)
    donor <- donor[!which(duplicated(donor$rsID)) -
        1]
    write.table(donor, file = paste0("/projects/rpci/lsuchest/abbasriz/
\t                                candidate_gene/result_files_cg/",
        donor.files[i]), quote = F, sep = "\t",
        col.names = T, row.names = F)
    rm(donor)
}
```

```
#!/bin/bash
#SBATCH --time=24:00:00
#SBATCH --nodes=1
#SBATCH --mem=5000
#SBATCH --job-name=myjobname
#SBATCH --output=myjob.out
#SBATCH --error=myjoberr.err
#SBATCH --partition=general-compute
#SBATCH --mail-user=abbasriz@buffalo.edu
#SBATCH --mail-type=ALL
#SBATCH --mail-type=END

#Get date and time
tstart=`date`
echo "###### start time:"$tstart
cd /projects/rpci/lsuchest/abbasriz/candidate_gene/\
    result_files_cg/impute.results.w.typedsnps/
echo "Run program"
module load R
R CMD BATCH donor_remove_dups.R
echo "program finished"
echo "All Done!"
tend=`date`
echo "###### end time: $tend"
```

We did this for recipient (`recipient_remove_dups.R` and `recipient_remove_dups.sh`) and shared (`shared_remove_dups.R` and `shared_remove_dups.sh`) genomes as well. Again, the donor, recipient, and shared result files and their corresponding shell scripts are in `/projects/rpci/lsuchest/ \`

abbasriz/candidate_gene/result_files_cg/impute.results.w.typedsnps. The output for this analysis can be found in: /projects/rpci/lsuchest/ \ abbasriz/candidate_gene/result_files_cg/, however, the files in these directories are the most current files, and they have been updated from this point in the analysis and include meta results, hazard ratios, and 95% confidence intervals for the HRs.

Meta-Analysis

So since the meta analysis files didn't have hazard ratio coefficients and confidence intervals, we had to re-run the metal analysis using STDERR option to get these results. The files from the meta-analysis can be found in /projects/rpci/lsuchest/abbasriz/candidate_gene/\ result_files_cg/metal_results/ These analyses are run pairwise (cohort 1 and cohort 2), e.g. /projects/rpci/lsuchest/ \ abbasriz/candidate_gene/result_files_cg/D_c1_DRM_ALLonly.txt and /projects/rpci/ \ lsuchest/abbasriz/candidate_gene/result_files_cg/D_c2_DRM_ALLonly.txt.

A metal script can be run using METAL software:

Example of METAL script (e.g. metal.txt)

```
module load metal
cat metal.txt
# META ANALYSIS FOR COHORT 1 and COHORT 2 FROM DISCOVERY-BMT
# CANDIDATE GENE REPLICATION/VALIDATION SUBSET
# THE RESULTS ARE STORED IN FILES metal_D_M_DRM_ALLonly.tbl
# and metal_D_M_DRM_ALLonly.tbl.info
SCHEME STDERR
```

```
# LOAD COHORT 1 and COHORT 2 FILES
# === DESCRIBE AND PROCESS THE FIRST INPUT FILE ===
MARKER rsID
ALLELE allele1 allele2
EFFECT coef
STDERR se.coef
PVALUE Pvalue
WEIGHT N
PROCESS /projects/rpci/lsuchest/abbasriz/candidate_gene/\
    result_files_cg/D_c1_DRM_ALLonly.txt
# === THE SECOND INPUT FILE HAS THE SAME FORMAT AND CAN BE PROCESSED IMMEDIATELY ===
PROCESS /projects/rpci/lsuchest/abbasriz/candidate_gene/\
    result_files_cg/D_c2_DRM_ALLonly.txt

OUTFILE /projects/rpci/lsuchest/abbasriz/candidate_gene/\
    result_files_cg/metal_results/metal_D_M_DRM_ALLonly .tbl
MINWEIGHT 10
```

Python script to generate metal files

To automate this for our cohorts we wrote a Python script to generate all of these different pairwise metal scripts.

```
#!/usr/bin/python
import glob
import io
# grab file names
files_c1=sorted(glob.glob("/projects/rpci/lsuchest/abbasriz/candidate_gene/result_files_cg/*c1*.txt"))
files_c2=sorted(glob.glob("/projects/rpci/lsuchest/abbasriz/candidate_gene/result_files_cg/*c2*.txt"))
# set up file names
meta_file_names = []
for i in range(len(files_c1)):
    last_item = files_c1[i].split("/")[-1]
    last_item = last_item.replace("c1", "M")
    last_item = last_item.replace(".txt", "")
    meta_file_names.append(last_item)
# generate metal scripts
#for i in meta_file_names:
for i in range(len(meta_file_names)):
    with open("run_metal_{}.txt".format(meta_file_names[i]), "w") as f:
        f.write("\nSCHEME STDERR")
        f.write("\n# LOAD COHORT 1 and COHORT 2 FILES")
        f.write("\n# === DESCRIBE AND PROCESS THE FIRST INPUT FILE ===")
    f.write("\nMARKER rsID")
    f.write("\nALLELE allele1 allele2")
    f.write("\nEFFECT coef")
    f.write("\nSTDERR se.coef")
    f.write("\nPVALUE Pvalue")
    f.write("\nWEIGHT N")
    f.write("\nPROCESS "+ files_c1[i])
    f.write("\n# === THE SECOND INPUT FILE HAS THE SAME FORMAT AND CAN BE PROCESSED IMMEDIATELY ===")
    f.write("\nPROCESS " + files_c2[i])
    f.write("\n")
    f.write("\nOUTFILE /projects/rpci/lsuchest/abbasriz/",
    "candidate_gene/result_files_cg/metal_results/metal_"+meta_file_names[i] + " .tbl")
```

```
    f.write("\nMINWEIGHT 10")
    f.write("\nANALYZE")
```

Another python script was written to create the shell script to run these METAL scripts

```
#!/usr/bin/python
import glob
import io
# grab file names
files = sorted(glob.glob("/projects/rpci/lsuchest/abbasriz/",
            "candidate_gene/result_files_cg/metal_results/run_*.txt"))
f = open('run_metal.sh', 'w')
f.write("#!/bin/bash")
f.write("\n#SBATCH --time=24:00:00")
f.write("\n#SBATCH --nodes=14")
f.write("\n#SBATCH --ntasks-per-node=4")
f.write("\n#SBATCH --mem-per-cpu=5000")
f.write("\n#SBATCH --job-name=myjobname")
f.write("\n#SBATCH --output=myjob.out")
f.write("\n#SBATCH --error=myjoberr.err")
f.write("\n#SBATCH --partition=general-compute")
f.write("\n#SBATCH --mail-user=abbariz@buffalo.edu")
f.write("\n#SBATCH --mail-type=ALL")
f.write("\n#SBATCH --mail-type=END")
f.write("\n")
f.write("\n#Get date and time")
f.write("\ntstart=`date`")
f.write("\necho \"###### start time:\"$tstart")
f.write("\nmodule load python")
f.write('\nmodule load metal')
f.write("\n")
for i in range(len(files)):
    f.write("metal " + files[i] + "\n")
f.write("\n")
f.write("program finished")
f.write("All done!")
f.write("tend=`date`")
f.write("echo \"###### end time:\"$tend")
f.write("exit")
f.close()
```

The results are stored in `.tbl` files in `/projects/rpci/lsuchest/abbasriz/ \ candidate_gene/metal_results/`.

Merge meta-analys results back into original files

Now that we had the meta-analysis results, we merged the beta coefficients and standard errors, back into the `._M_$'` files so that we could have the same format. The script that does this can be found at:

`/projects/rpci/lsuchest/abbasriz/candidate_gene/ \`
`metal_results/merge_metal_res_w_impute_res.R`

```
# merge_metal_res_w_impute_res.R
library(data.table)
path <- "/projects/rpci/lsuchest/abbasriz/candidate_gene/metal_results/"
metal.res.files <- list.files(path = path, pattern = ".tbl$")
metal.files <- lapply(metal.res.files, fread)
cnames <- c("rsID", "Allele1", "Allele2", "coef",
    "se.coef", "Pvalue", "Direction")
metal.files <- lapply(metal.files, setNames, cnames)
metal.res.files <- list.files(pattern = ".tbl$")
res.files <- list.files(pattern = "._M_", path = "/projects/rpci/lsuchest/abbasriz/candidate_gene/
                        result_files_cg/")
res.files <- paste0("/projects/rpci/lsuchest/abbasriz/
                    candidate_gene/result_files_cg/",
    res.files)
impute.res <- lapply(res.files, fread)
metal.files <- lapply(metal.files, function(x) setkey(x,
    rsID))
metal.files <- lapply(metal.files, function(x) x[,
    `:=`(c("Allele1", "Allele2", "Direction"),
        NULL), ])
metal.files <- lapply(metal.files, function(x) x[,
    `:=`(exp.coef, exp(coef)), ])
impute.res <- lapply(impute.res, function(x) setkey(x,
    rsID))
impute.res <- lapply(impute.res, function(x) x[,
    `:=`(c("coef", "se.coef", "exp.coef", "Pvalue"),
        NULL), ])

for (i in 1:length(impute.res)) {
    impute.res[[i]] <- impute.res[[i]][metal.files[[i]]]
}
impute.res <- lapply(impute.res, setcolorder,
    c("gene", "rsID", "chr", "BP", "allele1",
        "allele2", "N", "coef", "se.coef", "exp.coef",
        "95%-CI", "impute", "Pvalue"))
res.files <- list.files(pattern = "._M_", path = "/projects/rpci/lsuchest/
                        abbasriz/candidate_gene/result_files_cg/")
names(impute.res) <- res.files
res.files <- paste0("/projects/rpci/lsuchest/abbasriz/
                    candidate_gene/result_files_cg/",
    res.files)
for (i in 1:length(res.files)) {
    write.table(impute.res[[i]], file = res.files[[i]],
        sep = "\t", row.names = F, col.names = T,
        quote = F)
}
```

Calculate Hazard Ratios and 95% CI

Now that all the files are in the same format, we can automate the way that we calculate the hazard ratios and HR 95 CI, as all files should be the same for all analyses. Here we used the R script: `/projects/rpci/lsuchest/abbasriz/ \ candidate_gene/result_files_cg/calc_hazardratios.R`. This R script also ensures both allele1 and allele2 are uppercase and reassigns the rows in the `impute` column to `typed` (repeated rsid) and `imputed` (—).

```
library(data.table)
path <- "/projects/rpci/lsuchest/abbasriz/candidate_gene/result_files_cg/"
files <- list.files(path = path, pattern = ".txt")
res <- lapply(files, fread)
# 95% CI of hazard ratios
hr.ci <- function(coef.est, se) {
    lb <- round(exp(coef.est - 1.96 * se), 5)
    ub <- round(exp(coef.est + 1.96 * se), 5)
    paste0("[", lb, ", ", ub, "]")
}
# make new column of HR in lists
res <- lapply(res, function(x) x <- x[, `:=`("95%-CI",
    hr.ci(x$coef, x$se.coef))])
# change the imputation notation to 'imputed'
# or 'typed'
impute <- function(x) {
    setkey(x, impute)
    x[impute != "---", `:=`(impute, "typed")]
    x[impute == "---", `:=`(impute, "imputed")]
}
res <- lapply(res, impute)
# lets make alleles all upper case too
res <- lapply(res, function(x) x <- x[, `:=`("allele1",
    toupper(allele1)), ])
res <- lapply(res, function(x) x <- x[, `:=`("allele2",
    toupper(allele2)), ])
res <- lapply(res, setkey, gene)
names(res) <- files
res.files <- paste0("/projects/rpci/lsuchest/abbasriz/candidate_gene/result_files_cg/",
    files)
# sort by gene
res <- lapply(res, setkey, gene)
# write to file
for (i in 1:length(res.files)) {
    write.table(res[[i]], file = res.files[[i]],
        sep = "\t", row.names = F, col.names = T,
        quote = F)
}
```

Create genome, cohort, outcome, disease columns

The script can be found at `/projects/rpci/lsuchest/abbasriz/ \ candidate_gene/result_files_cg/descript_columns.R`

```
library(data.table)
# subsetted impute results w/ individual
# cohort and meta beta estimates/hr/CIs merged
path.cg <- "/projects/rpci/lsuchest/abbasriz/candidate_gene/result_files_cg/"
files.cg <- list.files(path = path.cg, pattern = ".txt")[-1]
impute.res <- lapply(paste0(path.cg, files.cg),
    fread)

# add the columns that we want, a column for
# genome, outcome, and cohort grab that
# information from the file names
genome <- gsub("_.*$", "", files.cg)
disease <- sub("(.*\\_)([^.]+)(\\.[[:alnum:]]+$)",
    "\\2", files.cg)
cohort <- sapply(strsplit(files.cg, "_", fixed = T),
    "[", 2)
outcome <- sapply(strsplit(files.cg, "_", fixed = T),
    "[", 3)

# creating columns with that info
mapply(function(x, cat) x <- x[, `:=`("genome",
    cat)], impute.res, genome)
mapply(function(x, cat) x <- x[, `:=`("cohort",
    cat)], impute.res, cohort)
mapply(function(x, cat) x <- x[, `:=`("outcome",
    cat)], impute.res, outcome)
mapply(function(x, cat) x <- x[, `:=`("disease",
    cat)], impute.res, disease)

impute.res <- lapply(impute.res, setkey, gene)
impute.res <- do.call(rbind, impute.res)
impute.res$gene[which(impute.res$gene == "GSTM1")] <- "GSTM"
impute.res <- na.omit(impute.res)

write.table(impute.res, file = "/projects/rpci/lsuchest/abbasriz/
            candidate_gene/result_files_cg/final_table/
            cg_snptable.txt",
    col.names = T, row.names = F, quote = F, sep = "\t")
```

VEGAS2

To run VEGAS2 you need two unlabeled columns: 1.) RSID and 2.) P-values. We created a shell script through R to do this, which grabs the 2nd and 13th column from our phenotype subsets.

```
path <- "/projects/rpci/lsuchest/abbasriz/candidate_gene/result_files_cg/"
x <- list.files(path = path, pattern = ".txt")
capture.output(file = "v2_parse.sh", {
    for (i in 1:length(x)) {
        cat("awk '{if (NR!=1) {print $2 \"\\t\" $13}}'",
            x[i], ">", paste0("/projects/rpci/lsuchest/abbasriz/candidate_gene/
\t           result_files_cg/vegas2/v2_",
               x[i]), "\n", sep = " ")
    }
})
```

The shell script can be found at: `/projects/rpci/lsuchest/abbasriz/ \ candidate_gene/result_files_cg/vegas2/v2_parse.sh` The results look like the following:

```
[abbasriz@rush:/projects/rpci/lsuchest/abbasriz/\
candidate_gene/result_files_cg/vegas2]$ head v2_D_c1_DRM_ALLonly.txt
rs6696752   0.827182998293298
rs72638700  0.874653327370856
rs6659287   0.526921155628948
rs141567582 0.874653327370856
rs6699669   0.262962565390368
rs6659541   0.627228150206333
rs12132479  0.890715829741203
rs6699881   0.829409938995696
rs41275458  0.874758581607631
rs7538516   0.756659623949985
```

The vegas2 results were split into 3 shell scripts (`run_vegas2_donor.sh`, `run_vegas2_recipient.sh`, `run_vegas2_shared.sh`). To create these files we used perl.

```
ls *.txt | grep '_D_' | perl \
    -lane '$new=$_; $new =~ s/\.txt$//; print "vegas2 $_ \
    -pop 1000GEURO \
    -subpop EURO \
    -genesize 10kbloc \
    -top 100 \
    -sex BothMnF -max 1000000 \
    -out ./results/$new.V2out "'  > run_vegas2.donor.sh
ls *.txt | grep '_R_' | perl \
    -lane '$new=$_; $new =~ s/\.txt$//; print "vegas2 $_ \
    -pop 1000GEURO \
    -subpop EURO \
    -genesize 10kbloc \
    -top 100 \
    -sex BothMnF \
    -max 1000000 \
    -out ./results/$new.V2out "' > run_vegas2_recipient.sh
ls *.txt | grep '_S_' | perl \
    -lane '$new=$_; $new =~ s/\.txt$//; print "vegas2 $_ \
    -pop 1000GEURO \
```

```
    -subpop EURO \
    -genesize 10kbloc \
    -top 100 \
    -sex BothMnF \
    -max 1000000 \
    -out ./results/$new.V2out "' > run_vegas2_shared.sh
```

These were put into SLURM scripts and the following lines were run in the programs

```
module load R
module load plink
module load vegas2
```

An example of the arguments that were used for vegas2 produced from the perl script can be seen here:

```
[abbasriz@rush:/projects/rpci/lsuchest/abbasriz/candidate_gene/vegas2]$ vegas2 v2_D_DRM_ALLonly.txt /
    -pop 1000GEURO /
    -subpop EURO /
    -genesize 10kbloc /
    -top 100 /
    -sex BothMnF /
    -max 1000000 /
    -out /projects/rpci/lsuchest/abbasriz/candidate_gene/vegas2/results/v2_D_DRM_ALLonly.V2out
```

VEGAS2 results are stored in: `/projects/rpci/lsuchest/abbasriz/ \ candidate_gene/result_files_cg/vegas2/results` with the extension `.V2out`.

Finalizing VEGAS2 results

```
# candidate gene list
gene.list <- scan("gene_list.txt", what = character())
gene.list <- c(gene.list, "GSTM1", "GSTM2", "GSTM3",
    "GSTM4", "GSTM5")
gene.list[gene.list == "GSTM"] <- NA
gene.list <- gene.list[complete.cases(gene.list)]
# vegas2 results
path.v2 <- "/projects/rpci/lsuchest/abbasriz/candidate_gene/result_files_cg/vegas2/results/"
files.v2 <- list.files(path = path.v2, pattern = ".V2out")
v2.res <- lapply(paste0(path.v2, files.v2), fread)
v2.res <- lapply(v2.res, setkey, Gene)
v2.res <- lapply(v2.res, function(x) x[gene.list])
v2.res <- lapply(v2.res, setNames, c("chr", "gene",
```

```
    "nSNPs", "nSims", "start", "stop", "Test",
    "geneBased.pval", "topSNP", "topSNP.pval"))
v2.res <- lapply(v2.res, setkey, gene)
# parse the file names to get the info that we
# want for the columns
genome <- sapply(strsplit(files.v2, "_", fixed = T),
    "[", 2)
outcome <- sapply(strsplit(files.v2, "_", fixed = T),
    "[", 4)
cohort <- sapply(strsplit(files.v2, "_", fixed = T),
    "[", 3)
disease <- sapply(strsplit(files.v2, "_", fixed = T),
    "[", 5)
disease <- sapply(strsplit(disease, ".", fixed = T),
    "[", 1)

# create descriptive columns
for (i in 1:length(v2.res)) {
    v2.res[[i]][, `:=`("genome", genome[i])]
    v2.res[[i]][, `:=`("cohort", cohort[i])]
    v2.res[[i]][, `:=`("outcome", outcome[i])]
    v2.res[[i]][, `:=`("disease", disease[i])]
}

v2.res <- do.call(rbind, v2.res)
head(v2.res)
write.table(v2.res, file = "/projects/rpci/lsuchest/abbasriz/candidate_gene/
            result_files_cg/final_table/cg_v2table.txt",
    col.names = T, row.names = F, quote = F, sep = "\t")
```

Top SNP and gene Based Pvalue adjustment

Grab number of typed SNPs from just cohort 1 (because it's the largest cohort). The script to count the typed SNPs is called `typed_snps.R` and `typed_snps.sh`.

```
# typed snps
library(data.table)
impute.res <- fread("/projects/rpci/lsuchest/abbasriz/candidate_gene/
                    result_files_cg/final_table/cg_snptable.txt")

## normalize top snp pvalue by number of typed
## snps in cohort 1
p <- c()
x <- c()
typed.snps <- unique(impute.res[, c("gene", "genome",
    "disease", "outcome"), with = F])

# my code
for (i in seq(nrow(typed.snps))) {
    x <- impute.res[gene == rr$gene[i] & disease ==
        rr$disease[i] & genome == rr$genome[i] &
        outcome == rr$outcome[i] & cohort == "c1"]
    setkey(x, "impute")
    p[i] <- nrow(x[impute == "typed"])
```

```
}

typed.snps <- typed.snps[, `:=`(no.typed.snps,
    p)]
head(typed.snps)
write.table(typed.snps, file = "typed.snps.txt",
    col.names = T, row.names = F, sep = "\t",
    quote = F)
```

We adjusted the topSNP pvalues by the number of typed SNPs in that gene (in cohort 1). We also adjusted the gene-based pvalue by the total number of genes that we tested in the candidate gene replication study.

```
library(data.table)
# vegas 2 results
v2.res <- fread("/projects/rpci/lsuchest/abbasriz/candidate_gene/
                result_files_cg/final_table/cg_v2table.txt")
# number of typed snps in each gene
typed.snps <- fread("/projects/rpci/lsuchest/abbasriz/candidate_gene/
                    result_files_cg/final_table/typed.snps.txt")

setkey(typed.snps, gene, genome, disease, outcome)
setkey(v2.res, gene, genome, disease, outcome)
v2.typed.snp <- v2.res[typed.snps]
# adjust for number of typed snps
v2.typed.snp <- v2.typed.snp[, `:=`("adj.topSNP.pval",
    topSNP.pval * no.typed.snps)]
# adjust for number of genes
v2.typed.snps <- v2.typed.snp[, `:=`("adj.geneBased.pval",
    geneBased.pval * 174)]
setcolorder(v2.typed.snp, c("chr", "gene", "nSNPs",
    "nSims", "start", "stop", "Test", "geneBased.pval",
    "adj.geneBased.pval", "topSNP", "topSNP.pval",
    "adj.topSNP.pval", "genome", "cohort", "outcome",
    "disease", "no.typed.snps"))

write.table(v2.typed.snps, file = "cg_v2table_adj.txt",
    col.names = T, row.names = F, quote = F, sep = "\t")
```

NOD2 analysis

NOD2 is a gene that has been frequently been studied in candidate gene studies looking at genetic variants from patients who have been treated with blood and marrow transplants (BMT). Oftentimes when NOD2 is studied, three SNPs frequently appear which have been labeled as SNP8 (rs2066844), SNP12 (rs2066845),

and SNP13 (rs2066847). SNP8 and SNP12 were genotyped in in the DISCOVeRY-BMT GWAS and SNP13 was not genotyped in DISCOVeRY-BMT. In order to still consider this SNP, we searched for a SNP that may be in LD with rs2066847, and the best LD pair found in DISCOVeRY-BMT was rs146528649 ($r^2 = 0.7422$). All of the groups that studied NOD2, also chose the same design in the way they analyzed this SNP, such that only the patient-donor pairs that were homozygous wildtype for all 3 SNPs were deemed "wild type", and on the contrary, if any one variant was present in any of the 3 SNPs in either of the donor-recipient pair, they were deemed to have a variant. Survival analysis was conducted using this aforementioned classification method.

Here we will discuss how we pre-processed the DISCOVeRY-BMT data to grab these SNPs of interest and how we implemented a similar composite scoring design and subsequent survival analysis. In order to replicate these studies we will subset rs2066844, rs2066845 and rs146528649 from the imputed data. Each of these SNPs lie on chr16 between the ranges of 50000000-55000000. The imputed data was found on the UB supercomputer at the following location:

```
## File location:
"/projects/rpci/lsuchest/lsuchest/Rserve/
ImputeData/var/db/gwas/imputed_data/
BMT093013_forImpute/BMT093013_forImpute.chr16-50000000-55000000.impute2"
```

The imputed data was extracted and loaded into R so that the SNPs of interest and their corresponding genotype probabilities for each sample can be pulled. The impute2 data was coverted into a VCF file. Also we need a file listing the SNPs in an unlabeled column vector (`nod2_snps.txt`). We will use QCTOOL to convert `.impute2` to `.vcf`.

```
module load qctool
qctool \
```

```
    -g BMT093013_forImpute.chr16-50000000-55000000.impute2 \
    -og /projects/rpci/lsuchest/abbasriz/\
    candidate_gene/nod2_rep/nod2_rep_dosages.vcf \
    -incl-rsids /projects/rpci/lsuchest/abbasriz/\
    candidate_gene/nod2_rep/nod2_snps.txt
```

Now that we have the vcf file, we can use `VariantAnnotation` library to easily pull this data into R.

```
library(VariantAnnotation)
## read in vcf file
vcf <- readVcf("/projects/rpci/lsuchest/abbasriz/
               candidate_gene/cg_haplotypes/nod2_rep/
               nod2_rep_dosages.vcf",
    genome = "hg19")
```

Now we will grab the donor and recipients FID and IID. These correspond to the indices for the samples that we will subset.

```
# grab unique identifiers of patients in
# different cohorts
library(data.table)
patients <- c("/projects/rpci/lsuchest/lsuchest/Rserve/BMT/
              genetic_data/R_c1_EA_FID_IID.txt",
    "/projects/rpci/lsuchest/lsuchest/Rserve/BMT/
              genetic_data/R_c2_EA_FID_IID.txt",
    "/projects/rpci/lsuchest/lsuchest/Rserve/BMT/
              genetic_data/D_c1_EA_FID_IID.txt",
    "/projects/rpci/lsuchest/lsuchest/Rserve/BMT/
              genetic_data/D_c2_EA_FID_IID.txt")
cohorts <- lapply(patients, fread)
files <- c()
for (i in 1:length(patients)) files[i] <- strsplit(patients,
    "/")[[i]][9]
names(cohorts) <- files
head(cohorts)
rec.ids <- list(cohorts[[1]], cohorts[[2]])
names(rec.ids) <- files[1:2]
donor.ids <- list(cohorts[[3]], cohorts[[4]])
names(donor.ids) <- files[3:4]
## read in phenotype files files
r.pheno <- fread("/projects/rpci/lsuchest/lsuchest/Rserve/
                 TheData/Plink_Recipient.pheno")
d.pheno <- fread("/projects/rpci/lsuchest/lsuchest/Rserve/
                 TheData/Plink_Donor.pheno")
r.cov <- fread("/projects/rpci/lsuchest/lsuchest/Rserve/
               TheData/Plink_Recipient.cov")
## SUBSET EUROPEAN AMERICAN PATIENTS BASED OFF
## OF IID
id.subset <- function(x, pheno) {
    pheno[pheno$IID %in% x$IID]
}
# recipients
rec.pheno.ea <- lapply(rec.ids, id.subset, r.pheno)
# donor
don.pheno.ea <- lapply(donor.ids, id.subset, d.pheno)
# bring down to just FID, IID, pair_id
```

```
don.pheno.ea <- lapply(don.pheno.ea, function(x) x[,
    c("FID", "IID", "pair_id"), with = F])
don.pheno.ea <- lapply(don.pheno.ea, setnames,
    c("D_FID", "D_IID", "pair_id"))
# merge to recipient file
merge.pair_id <- function(recipient, donor) {
    setkey(recipient, pair_id)
    setkey(donor, pair_id)
    recipient[donor]
}
merged.ea <- mapply(merge.pair_id, rec.pheno.ea,
    don.pheno.ea, SIMPLIFY = FALSE)
merged.ea <- data.table(do.call(rbind, merged.ea))
setnames(merged.ea, c("R_FID", "R_IID", colnames(merged.ea)[3:ncol(merged.ea)]))
# I think there are duplicated NAs because of
# the almost 2806 match and now 2815
merged.ea <- merged.ea[!duplicated(merged.ea$R_IID)]
ids <- merged.ea[, c("pair_id", "R_IID", "D_IID"),
    with = F]
# impute sample names
info.samples <- fread("/projects/rpci/lsuchest/lsuchest/Rserve/ImputeData/var/db/
                      gwas/imputed_data/BMT093013_forImpute/
                      BMT093013_forImpute.chr16-50000000-55000000.impute2_samples")
info.samples <- info.samples[-1]
nrow(info.samples)
# 6805 samples
nrow(gt)
# same number of samples...6805 relabel the
# IDs so they just correspond to the sample
# order because when we converted the impute2
# file to vcf, the order remains the same as
# seen info the impute2_samples file so we can
# just relabel the IDs as 1:6805 samples and
# go from there
info.samples$ID_1 <- seq(1, nrow(info.samples))
# okay now we will grab the indices from the
# impute_sample file and append them as
# columns to the ids data.table
ids$r_sample_index <- info.samples$ID_1[match(ids$R_IID,
    info.samples$ID_2)]
ids$d_sample_index <- info.samples$ID_1[match(ids$D_IID,
    info.samples$ID_2)]
# okay so now 22 donors didnt map back to that
# file
missing_samples <- ids[is.na(ids$d_sample_index)]
missing_samples
ids <- na.omit(ids)
head(ids)
# now we have 2783 samples that we can go and
# grab all the genotype info on
```

Now we we will grab the genotype threshold values

```
## threshold genotypes grab genotypes of 3 SNPs
## for all patients
gt <- geno(vcf)$GT
gt[1:nrow(gt), 1:5]
```

We can see that we have both typed and imputed for rs2066844. We will keep only

the imputed ones (which is the second rs2066844)

```
# only we have both typed and imputed of
# rs2066844 so we remove the first rs2066844
# because it is typed and we are keeping just
# the imputed ones
gt <- gt[-2, ]
gt[1:nrow(gt), 1:5]
```

Now we are going to transpose this data frame so we can have the samples as the rows and the SNPs as the columns

```
# transpose the data.frame so we can patients
# as rows
gt <- data.frame(t(gt))
gt[1:5, 1:ncol(gt)]
```

Now we code the alleles in a dominant model. So if there is any 1 alleles, we will just relabel it as 1 and if there are homozygous 0, we will relabel as 0. We will also relabel the . as NA and then remove the NAs

```
## coding dominant model over df columns
gt <- apply(gt, 2, FUN = function(x) gsub("[.]",
    NA, x))
gt <- apply(gt, 2, FUN = function(x) gsub("0/0",
    0, x))
gt <- apply(gt, 2, FUN = function(x) gsub("0/1",
    1, x))
gt <- apply(gt, 2, FUN = function(x) gsub("1/1",
    1, x))
## create a data.frame that changes the factors
## into numeric and keeps the row.names ...
gt <- data.frame(apply(gt, 2, function(x) as.numeric(as.character(x))),
    check.names = F, row.names = row.names(gt))
head(gt)
```

Now we will subset based off of the genome and cohort we are interested and create variables to do the composite SNP testing. So if SNP 8 and SNP 12 had 0 for both SNPs, they will be recoded as 0, and if any had 1 in it, it will be recoded as a 1. This will also done for the 3 SNPs (SNP8/SNP12/SNP13).

```
## parse the unique IDs (genome/cohort) from
## the main genotype df into 4 dfs specific to
## genome/cohort
recs <- gt[ids$r_sample_index, ]
```

```
donors <- gt[ids$d_sample_index, ]
## recode with composite coding -- 0 if snps
## are all 0, 1 if any of the snps are 1 snp 8
## = rs2066844 x[,2] snp 12 = rs2066845 x[,3]
## snp 13 = rs146528649 x[,1]
recode <- function(x) {
    x$snp8snp12 <- ifelse((x[, 2] == 0 & x[, 3] ==
        0), x$snp8snp12 <- 0, x$snp8snp12 <- 1)
    x$compSNPs <- ifelse((x[, 1] == 0 & x[, 2] ==
        0 & x[, 3] == 0), x$compSNPs <- 0, x$compSNPs <- 1)
    data.table(x, keep.rownames = T)
}
recs <- recode(recs)
donors <- recode(donors)
# remove 'sample' from sample columns so we
# just have indices that we can map back to
# phenotype file
rm.sample <- function(x) {
    x$rn <- gsub("sample_", "", x$rn)
    class(x$rn) <- "numeric"
    x
}
recs <- rm.sample(recs)
donors <- rm.sample(donors)
# match column names to ids column names
colnames(recs)[1] <- "r_sample_index"
colnames(donors)[1] <- "d_sample_index"
# grab pair ids so we can compare donor and
# recipient pairs
recs$pair_id <- ids$pair_id
donors$pair_id <- ids$pair_id
# function to recode 'score', again, if there
# is 0 in both D-R pairs the score 0,
# otherwise its 1
comp_score <- function(x) {
    x$NOD2_score <- ifelse((x[, "recipient"] ==
        0 & x[, "donor"] == 0), x$NOD2_score <- 0,
        x$NOD2_score <- 1)
    x
}
# merge snp8/snp12 donor-recipient pair
# data.frame
snp8snp12.r <- recs[, c("snp8snp12", "pair_id")]
setkey(snp8snp12.r, pair_id)
snp8snp12.d <- donors[, c("snp8snp12", "pair_id")]
setkey(snp8snp12.d, pair_id)
snp8snp12 <- snp8snp12.d[snp8snp12.r]
colnames(snp8snp12) <- c("donor", "pair_id", "recipient")
snp8snp12 <- snp8snp12[, c("pair_id", "donor",
    "recipient")]
snp8snp12 <- comp_score(snp8snp12)
# merge snp8/snp12/snp13 donor-recipient pair
# data.frame
all3.r <- recs[, c("compSNPs", "pair_id")]
setkey(all3.r, pair_id)
all3.d <- donors[, c("compSNPs", "pair_id")]
setkey(all3.d, pair_id)
all3 <- all3.d[all3.r]
colnames(all3) <- c("donor", "pair_id", "recipient")
all3 <- all3[, c("pair_id", "donor", "recipient")]
all3 <- comp_score(all3)
colnames(snp8snp12)[ncol(snp8snp12)] <- "snp8snp12_score"
colnames(all3)[ncol(all3)] <- "all3_score"
snp8snp12 <- snp8snp12[, c("pair_id", "snp8snp12_score")]
```

```
all3 <- all3[, c("pair_id", "all3_score")]
```

Now we have the recoded SNP8 and SNP12. We need to append these back to the phenotype files so we can run survival analyses on these. Here we grab the recipient and donor phenotype files and merge them based off of pair_id column.

```
# merge back to phenotype file .. which is
# called merged.ea
setkey(merged.ea, pair_id)
setkey(snp8snp12, pair_id)
setkey(all3, pair_id)
merged.ea <- merged.ea[snp8snp12]
merged.ea <- merged.ea[all3]
# lets quickly see how many people had NA
# genotype scores
table(is.na(merged.ea$all3_score))
# 486 deemed NA for all 3
table(is.na(merged.ea$snp8snp12_score))
# 208 deemed NA for snp8snp12
# write.table(merged.ea,
# file='pheno_merged_nod2haps.txt', sep='\t',
# row.names=F, col.names=T, quote=F)
```

Survival analysis

```
######################## SURVIVAL ANALYSIS w do survival
library(survival)
# grab covariate files
r.cov <- fread("/projects/rpci/lsuchest/lsuchest/Rserve/TheData/
               Plink_Recipient.cov")

r.cov <- r.cov[r.cov$pair_id %in% ids$pair_id]
setkey(r.cov, pair_id)
r.cov$cohort <- merged.ea$cohort
r.cov.c1 <- r.cov[cohort == "c1"]
r.cov.c2 <- r.cov[cohort == "c2"]
################################### DRM covariates are: age, distatD cohort 1
################################### snp 8/snp12
DRM.c1 <- Surv(time = merged.ea.c1$intxsurv_1Y,
    event = merged.ea.c1$disease_death_1Y == 1)
DRM.snp8snp12.c1 <- coxph(DRM.c1 ~ merged.ea.c1$snp8snp12_score +
    r.cov.c1$distatD)
DRM.snp8snp12.c1.coef <- summary(DRM.snp8snp12.c1)$coef[1,
    1]
DRM.snp8snp12.c1.se.coef <- summary(DRM.snp8snp12.c1)$coef[1,
    3]
DRM.snp8snp12.c1.hr <- summary(DRM.snp8snp12.c1)$coef[1,
    2]
DRM.snp8snp12.c1.pval <- summary(DRM.snp8snp12.c1)$coef[1,
    5]
# snp8/snp12/snp13
DRM.all3.c1 <- coxph(DRM.c1 ~ merged.ea.c1$all3_score +
    r.cov.c1$distatD)
## cohort 2 snp 8/snp12
DRM.c2 <- Surv(time = merged.ea.c2$intxsurv_1Y,
```

```
    event = merged.ea.c2$disease_death_1Y == 1)
DRM.snp8snp12.c2 <- coxph(DRM.c2 ~ merged.ea.c2$snp8snp12_score +
    r.cov.c2$distatD)
# snp8/snp12/snp13
DRM.all3.c2 <- coxph(DRM.c2 ~ merged.ea.c2$all3_score +
    r.cov.c2$distatD)
DRM.all3.c2 <- coxph(DRM.c2 ~ merged.ea.c2$all3_score +
    r.cov.c2$distatD)
############################################ OS covariates are: age, distatD, Pblood OS
############################################ #### cohort 1 OS covariates are: age,
############################################ distatD, Pblood
OS.c1 <- Surv(time = merged.ea.c1$intxsurv_1Y,
    event = merged.ea.c1$dead_1Y == 1)
# OS SNP8/SNP12
os.snp8snp12.c1 <- coxph(OS.c1 ~ merged.ea.c1$snp8snp12_score +
    r.cov.c1$age + r.cov.c1$distatD + r.cov.c1$PBlood)
summary(os.snp8snp12.c1)$coef[1, 5]
# OS SNP8/SNP12/SNP13
os.all3.c1 <- coxph(OS.c1 ~ merged.ea.c1$all3_score +
    r.cov.c1$age + r.cov.c1$distatD + r.cov.c1$PBlood)
summary(os.all3.c1)$coef[1, 5]
## cohort 2
OS.c2 <- Surv(time = merged.ea.c2$intxsurv_1Y,
    event = merged.ea.c2$dead_1Y == 1)
# OS SNP8/SNP12
os.snp8snp12.c2 <- coxph(OS.c2 ~ merged.ea.c2$snp8snp12_score +
    r.cov.c2$age + r.cov.c2$distatD + r.cov.c2$PBlood)
summary(os.snp8snp12.c2)$coef[1, 5]

# OS SNP8/SNP12/SNP13
os.all3.c2 <- coxph(OS.c2 ~ merged.ea.c2$all3_score +
    r.cov.c2$age + r.cov.c2$distatD + r.cov.c2$PBlood)
summary(os.all3.c2)$coef[1, 5]
################################### PFS covariates: age, distatD cohort 1
PFS.c1 <- Surv(time = merged.ea.c1$intxsurv_1Y,
    event = merged.ea.c1$lfs_1Y == 1)
# PFS SNP8 / SNP12
pfs.snp8snp12.c1 <- coxph(PFS.c1 ~ merged.ea.c1$snp8snp12_score +
    r.cov.c1$distatD + r.cov.c1$age)
summary(pfs.snp8snp12.c1)$coefficients[1, 5]
# PFS SNP 8 / SNP 12 / SNP 13
pfs.all3.c1 <- coxph(PFS.c1 ~ as.numeric(merged.ea.c1$all3_score) +
    r.cov.c1$distatD + as.numeric(r.cov.c1$age))
summary(pfs.all3.c1)$coefficients[1, 5]
# cohort 2
PFS.c2 <- Surv(time = merged.ea.c2$intxsurv_1Y,
    event = merged.ea.c2$lfs_1Y == 1)
# PFS SNP8 / SNP12
pfs.snp8snp12.c2 <- coxph(PFS.c2 ~ merged.ea.c2$snp8snp12_score +
    r.cov.c2$distatD + r.cov.c2$age)
summary(pfs.snp8snp12.c2)$coefficients[1, 5]
# PFS SNP 8 / SNP 12 / SNP 13
pfs.all3.c2 <- coxph(PFS.c2 ~ as.numeric(merged.ea.c2$all3_score) +
    r.cov.c2$distatD + as.numeric(r.cov.c2$age))
summary(pfs.all3.c2)$coefficients[1, 5]
################################################ TRM covariates: age, bmiOBS, bmiOVWT, PBlood
################################################ cohort 1
TRM.c1 <- Surv(time = merged.ea.c1$intxsurv_1Y,
    event = merged.ea.c1$TRM_1Y == 1)
# TRM SNP 8 / SNP 12
trm.snp8snp12.c1 <- coxph(TRM.c1 ~ as.numeric(merged.ea.c1$snp8snp12_score) +
    as.numeric(r.cov.c1$age) + r.cov.c1$PBlood +
    r.cov.c1$bmiOBS + r.cov.c1$bmiOVWT)
summary(trm.snp8snp12.c1)$coefficients[1, 5]
```

```
# TRM SNP 8 / SNP 12 / SNP 13
trm.all3.c1 <- coxph(TRM.c1 ~ as.numeric(merged.ea.c1$all3_score) +
    as.numeric(r.cov.c1$age) + r.cov.c1$PBlood +
    as.numeric(r.cov.c1$bmiOBS) + as.numeric(r.cov.c1$bmiOVWT))
summary(trm.all3.c1)$coefficients[1, 5]
# cohort 2
TRM.c2 <- Surv(time = merged.ea.c2$intxsurv_1Y,
    event = merged.ea.c2$TRM_1Y == 1)
# TRM SNP 8 / SNP 12
trm.snp8snp12.c2 <- coxph(TRM.c2 ~ as.numeric(merged.ea.c2$snp8snp12_score) +
    as.numeric(r.cov.c2$age) + r.cov.c2$PBlood +
    r.cov.c2$bmiOBS + r.cov.c2$bmiOVWT)
summary(trm.snp8snp12.c2)$coefficients[1, 5]
# TRM SNP 8 / SNP 12 / SNP 13
trm.all3.c2 <- coxph(TRM.c2 ~ as.numeric(merged.ea.c2$all3_score) +
    as.numeric(r.cov.c2$age) + r.cov.c2$PBlood +
    as.numeric(r.cov.c2$bmiOBS) + as.numeric(r.cov.c2$bmiOVWT))
summary(trm.all3.c2)$coefficients[1, 5]


## WRITE TO TABLE NEED TO SPLIT INTO TWO TABLES
## FOR META-ANALYSIS will calculate confidence
## interval of hazard ratios AFTER set up
## columns so they are the same as our
## candidate gene table
cols <- c("gene", "rsID", "chr", "BP", "N", "allele1",
    "allele2", "coef", "se.coef", "exp.coef",
    "95%-CI", "Pvalue", "impute", "genome", "cohort",
    "outcome", "disease")
## Cohort 1
make.tbl <- function(outcome.vector, gene, rsID,
    chr, BP, N, allele1, allele2, genome, cohort,
    outcome, disease) {
    coef <- outcome.vector[1, 1]
    se.coef <- outcome.vector[1, 3]
    hr <- outcome.vector[1, 2]
    pval <- outcome.vector[1, 5]
    res <- c(gene, rsID, chr, BP, N, allele1,
        allele2, coef, se.coef, hr, NA, pval,
        "imputed", genome, cohort, outcome, disease)
    res
}

# cohort 1
nod2.c1 <- data.table(rbind(make.tbl(summary(DRM.snp8snp12.c1)$coef,
    "NOD2", "2 SNPs* DRM", "chr16", "*", 2033,
    "*", "*", "S", "c1", "DRM", "mixed"), make.tbl(summary(os.snp8snp12.c1)$coef,
    "NOD2", "2 SNPs* OS", "chr16", "*", 2033,
    "*", "*", "S", "c1", "OS", "mixed"), make.tbl(summary(trm.snp8snp12.c1)$coef,
    "NOD2", "2 SNPs* TRM", "chr16", "*", 2033,
    "*", "*", "S", "c1", "TRM", "mixed"), make.tbl(summary(pfs.snp8snp12.c1)$coef,
    "NOD2", "2 SNPs* PFS", "chr16", "*", 2033,
    "*", "*", "S", "c1", "PFS", "mixed"), make.tbl(summary(DRM.all3.c1)$coef,
    "NOD2", "3 SNPs* DRM", "chr16", "*", 2033,
    "*", "*", "S", "c1", "DRM", "mixed"), make.tbl(summary(os.all3.c1)$coef,
    "NOD2", "3 SNPs* OS", "chr16", "*", 2033,
    "*", "*", "S", "c1", "OS", "mixed"), make.tbl(summary(pfs.all3.c1)$coef,
    "NOD2", "3 SNPs* PFS", "chr16", "*", 2033,
    "*", "*", "S", "c1", "PFS", "mixed"), make.tbl(summary(trm.all3.c1)$coef,
    "NOD2", "3 SNPs* TRM", "chr16", "*", 2033,
    "*", "*", "S", "c1", "TRM", "mixed")))
colnames(nod2.c1) <- cols
write.table(nod2.c1, file = "nod2.c1.txt", col.names = T,
    row.names = F, quote = F, sep = "\t")
```

```
## Cohort 2
nod2.c2 <- data.table(rbind(make.tbl(summary(DRM.snp8snp12.c2)$coef,
    "NOD2", "2 SNPs* DRM", "chr16", "*", 757,
    "*", "*", "S", "c2", "DRM", "mixed"), make.tbl(summary(os.snp8snp12.c2)$coef,
    "NOD2", "2 SNPs* OS", "chr16", "*", 757, "*",
    "*", "S", "c2", "OS", "mixed"), make.tbl(summary(pfs.snp8snp12.c2)$coef,
    "NOD2", "2 SNPs* PFS", "chr16", "*", 757,
    "*", "*", "S", "c2", "PFS", "mixed"), make.tbl(summary(trm.snp8snp12.c2)$coef,
    "NOD2", "2 SNPs* TRM", "chr16", "*", 757,
    "*", "*", "S", "c2", "TRM", "mixed"), make.tbl(summary(DRM.all3.c2)$coef,
    "NOD2", "3 SNPs* DRM", "chr16", "*", 757,
    "*", "*", "S", "c2", "DRM", "mixed"), make.tbl(summary(os.all3.c2)$coef,
    "NOD2", "3 SNPs* OS", "chr16", "*", 757, "*",
    "*", "S", "c2", "OS", "mixed"), make.tbl(summary(pfs.all3.c2)$coef,
    "NOD2", "3 SNPs* PFS", "chr16", "*", 757,
    "*", "*", "S", "c2", "PFS", "mixed"), make.tbl(summary(trm.all3.c2)$coef,
    "NOD2", "3 SNPs* TRM", "chr16", "*", 757,
    "*", "*", "S", "c2", "TRM", "mixed")))
colnames(nod2.c2) <- cols
write.table(nod2.c2, file = "nod2.c2.txt", col.names = T,
    row.names = F, quote = F, sep = "\t")
```

Meta-analysis was performed using METAL software. The hazard ratios and 95% confidence intervals were computed using the standard errors from the METAL output.

```
# RUN METAL append meta results
nod2.meta <- fread("metal_S_M_mixed_nod21.tbl")
# remove direction column
nod2.meta <- nod2.meta[, -7]
# build up columns to match how our final
# table is set up
nod2.meta$gene <- "NOD2"
colnames(nod2.meta)[1] <- "rsID"
nod2.meta$chr <- "chr16"
nod2.meta$exp.coef <- exp(nod2.meta$Effect)
nod2.meta$genome <- "S"
nod2.meta$cohort <- "M"
nod2.meta$disease <- "mixed"
nod2.meta$N <- 2790
nod2.meta$BP <- "*"
nod2.meta$impute <- "imputed"
nod2.meta$`95%-CI` <- NA
nod2.meta$outcome <- NA
colnames(nod2.meta) <- c("rsID", "allele1", "allele2",
    "coef", "se.coef", "Pvalue", "gene", "chr",
    "exp.coef", "genome", "cohort", "disease",
    "N", "BP", "impute", "95%-CI", "outcome")

nod2.meta$outcome[c(1, 8)] <- "OS"
nod2.meta$outcome[2:3] <- "DRM"
nod2.meta$outcome[c(4, 6)] <- "PFS"
nod2.meta$outcome[c(5, 7)] <- "TRM"
setcolorder(nod2.meta, cols)
# join all of the nod2 results (c1, c2, meta
# analyiss)
nod2.final <- do.call(rbind, list(nod2.c1, nod2.c2,
    nod2.meta))
```

```
# remove extra names in rsID column
nod2.final$rsID <- substr(nod2.final$rsID, 1,
    7)
## now calculate 95% confidence interval for
## hazard ratio
hr.ci <- function(coef.est, se) {
    lb <- round(exp(coef.est - 1.96 * se), 5)
    ub <- round(exp(coef.est + 1.96 * se), 5)
    paste0("[", lb, ", ", ub, "]")
}
for (i in 1:nrow(nod2.final)) {
    nod2.final[i, "95%-CI"] <- hr.ci(as.numeric(nod2.final$coef)[i],
        as.numeric(nod2.final$se.coef)[i])
}
write.table(nod2.final, file = "nod2_survival_pvals.txt",
    col.names = T, row.names = F, sep = "\t",
    quote = F)
```

CCR5 analysis

Creating a file with the CCR SNPs in it:

```
printf "rs1799987\nrs333\nrs1800023\nrs1800024\nrs113341849" > ccr5_snps_tt.txt
```

```
library(VariantAnnotation)
library(data.table)
library(survival)
library(dplyr)
## read in vcf file
vcf <- readVcf("/projects/rpci/lsuchest/abbasriz/
           candidate_gene/cg_haplotypes/ccr5_rep/
           ccr5_rep_dosages.vcf",
    genome = "hg19")
## threshold genotypes grab genotypes of 3 SNPs
## for all patients
gt <- geno(vcf)$GT
gt[1:nrow(gt), 1:5]
## no rs333 in our data proxy which rs113341849
## transpose the data.frame so we can patients
## as rows
gt <- data.frame(t(gt))
gt[1:5, 1:ncol(gt)]
## CCR5 rs113341849
table(gt$rs113341849)
h1h1 <- gt
# recode to homozygous major allele
h1h1 <- apply(h1h1, 2, FUN = function(x) gsub("[.]",
    NA, x))
h1h1 <- apply(h1h1, 2, FUN = function(x) gsub("0/0",
    1, x))
h1h1 <- apply(h1h1, 2, FUN = function(x) gsub("0/1",
    0, x))
h1h1 <- apply(h1h1, 2, FUN = function(x) gsub("1/1",
    0, x))

dosages <- fread("ccr5_dosages.impute2")
dosages <- data.table(t(dosages))
```

```
colnames(dosages) <- as.character(dosages[3, ])
dosages <- dosages[-c(1:6), ]
dosages <- data.frame(apply(dosages, 2, as.numeric))
dosages$ccr5 <- rowSums(dosages)
dosages <- dosages %>% mutate(h1h1 = ifelse(ccr5 <
    0.5, 1, 0)) %>% data.table(keep.rownames = T)
################################################ LOAD UP PATIENT ID AND COVARIATE FILES grab
################################################ unique identifiers of patients in different
################################################ cohorts
library(data.table)
patients <- c("/projects/rpci/lsuchest/lsuchest/
Rserve/BMT/genetic_data/R_c1_EA_FID_IID.txt",
    "/projects/rpci/lsuchest/lsuchest
/Rserve/BMT/genetic_data/R_c2_EA_FID_IID.txt",
    "/projects/rpci/lsuchest/lsuchest/
Rserve/BMT/genetic_data/D_c1_EA_FID_IID.txt",
    "/projects/rpci/lsuchest/lsuchest/
Rserve/BMT/genetic_data/D_c2_EA_FID_IID.txt")
cohorts <- lapply(patients, fread)
files <- c()
for (i in 1:length(patients)) files[i] <- strsplit(patients,
    "/")[[i]][9]
names(cohorts) <- files
head(cohorts)
rec.ids <- list(cohorts[[1]], cohorts[[2]])
names(rec.ids) <- files[1:2]
rec.ids[[1]]$cohort <- "c1"
rec.ids[[2]]$cohort <- "c2"
donor.ids <- list(cohorts[[3]], cohorts[[4]])
names(donor.ids) <- files[3:4]
donor.ids[[1]]$cohort <- "c1"
donor.ids[[2]]$cohort <- "c2"
## read in phenotype files files
r.pheno <- fread("/projects/rpci/lsuchest/lsuchest/Rserve/
                TheData/Plink_Recipient.pheno")
d.pheno <- fread("/projects/rpci/lsuchest/lsuchest/Rserve/
                TheData/Plink_Donor.pheno")
r.cov <- fread("/projects/rpci/lsuchest/lsuchest/Rserve/
              TheData/Plink_Recipient.cov")
# parse covariant/pheno files to just EA
# SUBSET EUROPEAN AMERICAN PATIENTS BASED OFF
# OF IID
id.subset <- function(x, pheno) {
    pheno[pheno$IID %in% x$IID]
}
# recipients
rec.pheno.ea <- lapply(rec.ids, id.subset, r.pheno)
# donors
don.pheno.ea <- lapply(donor.ids, id.subset, d.pheno)
# bring down to just FID, IID, pair_id
don.pheno.ea <- lapply(don.pheno.ea, function(x) x[,
    c("FID", "IID", "pair_id", "population"),
    with = F])
don.pheno.ea <- lapply(don.pheno.ea, setnames,
    c("D_FID", "D_IID", "pair_id", "population"))
# merge to recipient file
merge.pair_id <- function(recipient, donor) {
    setkey(recipient, pair_id)
    setkey(donor, pair_id)
    recipient[donor]
}
merged.ea <- mapply(merge.pair_id, rec.pheno.ea,
    don.pheno.ea, SIMPLIFY = FALSE)
merged.ea <- data.table(do.call(rbind, merged.ea))
```

```
setnames(merged.ea, c("R_FID", "R_IID", colnames(merged.ea)[3:ncol(merged.ea)]))
# I think there are duplicated NAs because of
# the almost 2806 match and now 2815
merged.ea <- merged.ea[!duplicated(merged.ea$R_IID)]
ids <- merged.ea[, c("pair_id", "R_IID", "D_IID"),
    with = F]
# impute sample names
info.samples <- fread("/projects/rpci/
                      lsuchest/lsuchest/Rserve/ImputeData/
                      var/db/gwas/imputed_data/
                      BMT093013_forImpute/
                      BMT093013_forImpute.chr16-
                      50000000-55000000.impute2_samples")
# top row is useless
info.samples <- info.samples[-1]
nrow(info.samples)
# 6805 samples
nrow(gt)
# nice! same number of samples...6805 relabel
# the IDs so they just correspond to the
# sample order because when we converted the
# impute2 file to vcf, the order remains the
# same as seen info the impute2_samples file
# so we can just relabel the IDs as 1:6805
# samples and go from there
info.samples$ID_1 <- seq(1, nrow(info.samples))
# okay now we will grab the indices from the
# impute_sample file and append them as
# columns to the ids data.table
ids$r_sample_index <- info.samples$ID_1[match(ids$R_IID,
    info.samples$ID_2)]
ids$d_sample_index <- info.samples$ID_1[match(ids$D_IID,
    info.samples$ID_2)]
# okay so now 22 donors didnt map back to that
# file
missing_samples <- ids[is.na(ids$d_sample_index)]
missing_samples
ids <- na.omit(ids)
# now we have 2783 samples that we can go and
# grab all the genotype info on NOW PARSE
# GENOTYPE FILE BY SAMPLE ID INDICES parse the
# unique IDs (genome/cohort) from the main
# genotype df into 4 dfs specific to
# genome/cohort
recs <- data.table(dosages[ids$r_sample_index,
    ], keep.rownames = T)
donors <- data.table(dosages[ids$d_sample_index,
    ], keep.rownames = T)
# remove 'sample' from sample columns so we
# just have indices that we can map back to
# phenotype file
rm.sample <- function(x) {
    x$rn <- gsub("sample_", "", x$rn)
    class(x$rn) <- "numeric"
    x
}
recs <- rm.sample(recs)
donors <- rm.sample(donors)
# match column names to ids column names
colnames(recs)[1] <- "r_sample_index"
colnames(donors)[1] <- "d_sample_index"
# grab pair ids so we can compare donor and
# recipient pairs
recs$pair_id <- ids$pair_id
```

```
donors$pair_id <- ids$pair_id
recs <- recs %>% na.omit()
donors <- donors %>% na.omit()
recs <- recs %>% rename(Rh1h1 = h1h1)
donors <- donors %>% rename(Dh1h1 = h1h1)
donors.h1h1 <- donors %>% select(pair_id, Dh1h1)
recs.h1h1 <- recs %>% select(pair_id, Rh1h1)
dr.h1h1 <- donors.h1h1 %>% right_join(recs.h1h1,
    "pair_id") %>% data.table()
dr.h1h1$grp2vsgrp1 <- with(dr.h1h1, ifelse(Dh1h1 ==
    1 & Rh1h1 == 0, 1, ifelse(Dh1h1 == 0 & Rh1h1 ==
    0, 0, NA)))
dr.h1h1$grp3vsgrp1 <- with(dr.h1h1, ifelse(Dh1h1 ==
    0 & Rh1h1 == 1, 1, ifelse(Dh1h1 == 0 & Rh1h1 ==
    0, 0, NA)))
dr.h1h1$grp3vsgrp2 <- with(dr.h1h1, ifelse(Dh1h1 ==
    0 & Rh1h1 == 1, 1, ifelse(Dh1h1 == 1 & Rh1h1 ==
    0, 0, NA)))
r.cov <- r.cov %>% right_join(dr.h1h1, "pair_id") %>%
    data.table()
## split into cohorts
r.cov.c1 <- r.cov[cohort1 == 1]
r.cov.c2 <- r.cov[cohort1 == 0]
merged.ea <- r.cov %>% dplyr::select(pair_id,
    age, distatD, PBlood, bmiOBS, bmiOVWT, MDSdummy,
    AMLdummy, ALLdummy, Dh1h1, Rh1h1, grp2vsgrp1,
    grp3vsgrp1, grp3vsgrp2) %>% inner_join(merged.ea,
    by = "pair_id") %>% data.table()
merged.ea$population <- gsub("EA.", "", merged.ea$population)
OScov1Y <- c("intxsurv_1Y", "dead_1Y", "age",
    "distatD", "PBlood")
PFScov1Y <- c("intxrel_1Y", "lfs_1Y", "age", "distatD")
OScov.full <- c("intxsurv_1Y", "dead_1Y", "age",
    "distatD", "PBlood")
PFScov.full <- c("intxrel_1Y", "lfs_1Y", "age",
    "distatD")
################################################ SURVIVAL ANALYSIS EVENTS ######## DRM -
################################################ disease_death_1Y OS - dead_1Y PFS - lfs_1Y
################################################ TRM - TRM_1Y COVARIATES
################################################ ###################### DRM covariates are:
################################################ age, distatD OS covariates are: age,
################################################ distatD, Pblood PFS covariates: age, distatD
################################################ TRM covariates: age, bmiOBS, bmiOVWT, PBlood
DRMcov <- c("intxsurv_1Y", "disease_death_1Y",
    "age", "distatD")
OScov <- c("intxsurv_1Y", "dead_1Y", "age", "distatD",
    "PBlood")
PFScov <- c("intxrel_1Y", "lfs_1Y", "age", "distatD")
TRMcov <- c("intxsurv_1Y", "TRM_1Y", "age", "bmiOBS",
    "bmiOVWT", "PBlood")
# adjusts for 2 covariates + genotype of
# interest
surv_fit_cov2 <- function(geno, cov, covFile) {
    outcome <- Surv(time = covFile[, cov[1]],
        event = covFile[, cov[2]])
    res <- coxph(outcome ~ covFile[, geno] + covFile[,
        cov[3]] + covFile[, cov[4]])
    summary(res)$coefficients[1, c(1, 3, 2, 5)]
}
# adjusts for 3 covariates + genotype of
# interest
surv_fit_cov3 <- function(geno, cov, covFile) {
    outcome <- Surv(time = covFile[, cov[1]],
        event = covFile[, cov[2]])
```

```
    res <- coxph(outcome ~ covFile[, geno] + covFile[,
        cov[3]] + covFile[, cov[4]] + covFile[,
        cov[5]])
    summary(res)$coefficients[1, c(1, 3, 2, 5)]
}
# adjusts for 4 covariates + genotype of
# interest
surv_fit_cov4 <- function(geno, cov, covFile) {
    outcome <- Surv(time = covFile[, cov[1]],
        event = covFile[, cov[2]])
    res <- coxph(outcome ~ covFile[, geno] + covFile[,
        cov[3]] + covFile[, cov[4]] + covFile[,
        cov[5]] + covFile[, cov[6]])
    summary(res)$coefficients[1, c(1, 3, 2, 5)]
}
## WRITE TO TABLE NEED TO SPLIT INTO TWO TABLES
## FOR META-ANALYSIS will calculate confidence
## interval of HRs AFTER set up columns so they
## are the same as our CG table
cols <- c("gene", "rsID", "chr", "BP", "N", "allele1",
    "allele2", "coef", "se.coef", "exp.coef",
    "95%-CI", "Pvalue", "impute", "genome", "cohort",
    "outcome", "disease")

make.tbl <- function(outcome.vector, gene, rsID,
    chr, BP, N, allele1, allele2, imputed, genome,
    cohort, outcome, disease) {
    coef <- outcome.vector[1]
    se.coef <- outcome.vector[2]
    hr <- outcome.vector[3]
    pval <- outcome.vector[4]
    res <- c(gene, rsID, chr, BP, N, allele1,
        allele2, coef, se.coef, hr, NA, pval,
        imputed, genome, cohort, outcome, disease)
    res
}
################################ MASTER FILE CREATION #####
compsnps <- c("R_h1h1", "grp2vsgrp1", "grp3vsgrp1",
    "grp3vsgrp2")
cov.list <- c("DRMcov", "PFScov", "OScov", "TRMcov")
outcomes <- c("DRM", "PFS", "OS", "TRM")
cohorts <- c("c1", "c2")
diseases <- c("mixed", "AMLonly", "ALLonly", "noMDS",
    "noALL")
survival.functions <- c("surv_fit_cov2", "surv_fit_cov2",
    "surv_fit_cov3", "surv_fit_cov4")
create.master <- function(genotype, genome, outcomes,
    cohorts, diseases, survival.functions) {
    master <- data.frame(matrix(nrow = 3, ncol = 2))
    # master <- list()
    for (i in 1:length(diseases)) {
        master[i, ] <- c(genotype, diseases[i])

    }
    colnames(master) <- c("genotype", "disease")
    master.list <- list()
    for (i in 1:length(outcomes)) {
        master$outcome <- outcomes[i]
        master$covList <- cov.list[i]
        master$survivalFunc <- survival.functions[i]
        master$genome <- genome
        master.list[[i]] <- master
    }
    master.merge <- data.table(do.call(rbind,
```

```
        master.list))
    master.list <- list()
    for (i in 1:length(cohorts)) {
        master.merge$cohort <- cohorts[i]
        master.list[[i]] <- master.merge
    }
    do.call(rbind, master.list)
}
# donors
master.dh1h1 <- create.master("Dh1h1", "D", outcomes,
    cohorts, diseases, survival.functions)
# recipients h1h1
master.rh1h1 <- create.master("Rh1h1", "R", outcomes,
    cohorts, diseases, survival.functions)
# shared
master.grp2vsgrp1 <- create.master("grp2vsgrp1",
    "S", outcomes, cohorts, diseases, survival.functions)
master.grp3vsgrp1 <- create.master("grp3vsgrp1",
    "S", outcomes, cohorts, diseases, survival.functions)
master.grp3vsgrp2 <- create.master("grp3vsgrp2",
    "S", outcomes, cohorts, diseases, survival.functions)
master.list <- data.table(do.call(rbind, list(master.dh1h1,
    master.rh1h1, master.grp2vsgrp1, master.grp3vsgrp1,
    master.grp3vsgrp2)))
outcome.order <- c("DRM", "PFS", "OS", "TRM")
# order by outcome so we can have DRM and PFS
# as \ top two outcomes b/c they both have 2
# covariates
master.list.2cov <- master.list[order(match(master.list$outcome,
    outcome.order))][1:100]
# OS has 3 covariates so we will grab those
master.list.3cov <- master.list[order(match(master.list$outcome,
    outcome.order))][101:150]
# trm has 4 covs
master.list.4cov <- master.list[order(match(master.list$outcome,
    outcome.order))][151:200]
## subset by cohort and disease
cohort.disease <- function(pheno, master.list) {
    if (master.list$cohort == "c1") {
        pheno <- pheno[cohort1 == 1]
    } else {
        pheno <- pheno[cohort1 == 0]
    }
    if (master.list$disease == "mixed") {
        data.frame(pheno)
    } else if (master.list$disease == "AMLonly") {
        data.frame(pheno[AMLdummy == 1])
    } else if (master.list$disease == "ALLonly") {
        data.frame(pheno[ALLdummy == 1])
    } else if (master.list$disease == "noALL") {
        data.frame(pheno[ALLdummy == 0])
    } else if (master.list$disease == "noMDS") {
        data.frame(pheno[MDSdummy == 0])
    }

}
# PFS
surv.res.DRMpfs <- data.frame(matrix(nrow = 100,
    ncol = 17))
colnames(surv.res.DRMpfs) <- cols
for (i in 1:nrow(surv.res.DRMpfs)) {
    surv.res.DRMpfs[i, ] <- make.tbl(surv_fit_cov2(master.list.2cov$genotype[i],
        eval(as.name(paste(master.list.2cov$covList[i]))),
        cohort.disease(merged.ea, master.list.2cov[i,
```

```
            ])), "CCR5", paste(master.list.2cov$genotype[i],
        master.list.2cov$outcome[i], master.list.2cov$disease[i],
        master.list.2cov$genome[i], sep = "_"),
        "chr3", "*", cohort.disease(merged.ea,
            master.list.2cov[i, ]) %>% select(cohort,
            eval(as.name(paste(master.list.2cov$genotype[i])))) %>%
            filter(cohort == master.list.2cov$cohort[i]) %>%
            na.omit %>% nrow, "*", "*", "imputed",
        master.list.2cov$genome[i], master.list.2cov$cohort[i],
        master.list.2cov$outcome[i], master.list.2cov$disease[i])
    surv.res.DRMpfs
}
# OVERALL SURVIVAL
surv.res.os <- data.frame(matrix(nrow = 50, ncol = 17))
colnames(surv.res.os) <- cols
for (i in 1:nrow(surv.res.os)) {
    surv.res.os[i, ] <- make.tbl(surv_fit_cov2(master.list.3cov$genotype[i],
        eval(as.name(paste(master.list.3cov$covList[i]))),
        cohort.disease(merged.ea, master.list.3cov[i,
            ])), "CCR5", paste(master.list.3cov$genotype[i],
        master.list.3cov$outcome[i], master.list.3cov$disease[i],
        master.list.3cov$genome[i], sep = "_"),
        "chr3", "*", cohort.disease(merged.ea,
            master.list.3cov[i, ]) %>% select(cohort,
            eval(as.name(paste(master.list.3cov$genotype[i])))) %>%
            filter(cohort == master.list.3cov$cohort[i]) %>%
            na.omit %>% nrow, "*", "*", "imputed",
        master.list.3cov$genome[i], master.list.3cov$cohort[i],
        master.list.3cov$outcome[i], master.list.3cov$disease[i])
    surv.res.os
}
# TRM
surv.res.trm <- data.frame(matrix(nrow = 50, ncol = 17))
colnames(surv.res.trm) <- cols
for (i in 1:nrow(surv.res.trm)) {
    surv.res.trm[i, ] <- make.tbl(surv_fit_cov2(master.list.4cov$genotype[i],
        eval(as.name(paste(master.list.4cov$covList[i]))),
        cohort.disease(merged.ea, master.list.4cov[i,
            ])), "CCR5", paste(master.list.4cov$genotype[i],
        master.list.4cov$outcome[i], master.list.4cov$disease[i],
        master.list.4cov$genome[i], sep = "_"),
        "chr3", "*", cohort.disease(merged.ea,
            master.list.4cov[i, ]) %>% select(cohort,
            eval(as.name(paste(master.list.4cov$genotype[i])))) %>%
            filter(cohort == master.list.4cov$cohort[i]) %>%
            na.omit %>% nrow, "*", "*", "imputed",
        master.list.4cov$genome[i], master.list.4cov$cohort[i],
        master.list.4cov$outcome[i], master.list.4cov$disease[i])
    surv.res.trm
}
## save to file
surv.res <- data.table(do.call(rbind, list(surv.res.DRMpfs,
    surv.res.os, surv.res.trm)))
# split into cohorts
surv.res.c1 <- surv.res[cohort == "c1"]
surv.res.c2 <- surv.res[cohort == "c2"]
write.table(surv.res.c1, file = "h1h1_nometa_c1.txt",
    sep = "\t", quote = F, row.names = F, col.names = T)
write.table(surv.res.c2, file = "h1h1_nometa_c2.txt",
    sep = "\t", quote = F, row.names = F, col.names = T)
## RAN META ANALYSIS ###
```

CCR5 meta analysis was run and the results are loaded in below:

```
## LOAD META RESULTS
surv.res.meta <- fread("metal_ccr5_full1.tbl")
surv.res.meta <- surv.res.meta[, -7]
meta.res <- function(ccr5.meta, col.order) {
    # remove direction column build up columns to
    # match how our final table is set up
    ccr5.meta$gene <- "CCR5"
    colnames(ccr5.meta)[1] <- "rsID"
    ccr5.meta$chr <- "chr3"
    ccr5.meta$exp.coef <- exp(ccr5.meta$Effect)
    ccr5.meta$genome <- sapply(strsplit(surv.res.meta$MarkerName,
        "_", fixed = T), "[", 4)
    ccr5.meta$cohort <- "M"
    ccr5.meta$disease <- sapply(strsplit(surv.res.meta$MarkerName,
        "_", fixed = T), "[", 3)
    ccr5.meta$N <- NA
    ccr5.meta$BP <- "*"
    ccr5.meta$impute <- "imputed"
    ccr5.meta$`95%-CI` <- NA
    ccr5.meta$outcome <- sapply(strsplit(surv.res.meta$MarkerName,
        "_", fixed = T), "[", 2)
    colnames(ccr5.meta) <- c("rsID", "allele1",
        "allele2", "coef", "se.coef", "Pvalue",
        "gene", "chr", "exp.coef", "genome", "cohort",
        "disease", "N", "BP", "impute", "95%-CI",
        "outcome")
    setcolorder(ccr5.meta, cols)
    data.table(ccr5.meta)
}

surv.res.meta <- meta.res(surv.res.meta, cols)
# need to make sure in right order for N
setkey(surv.res.c1, rsID)
setkey(surv.res.c2, rsID)
setkey(surv.res.meta, rsID)
surv.res.meta$N <- as.numeric(surv.res.c1$N) +
    as.numeric(surv.res.c2$N)
surv.res.full <- do.call(rbind, list(surv.res.c1,
    surv.res.c2, surv.res.meta))
surv.res.full$rsID <- sapply(strsplit(surv.res.full$rsID,
    "_", fixed = T), "[", 1)
ccr5.final <- surv.res.full
# calculate confidence interval
hr.ci <- function(coef.est, se) {
    lb <- round(exp(coef.est - 1.96 * se), 5)
    ub <- round(exp(coef.est + 1.96 * se), 5)
    paste0("[", lb, ", ", ub, "]")
}
for (i in 1:nrow(ccr5.final)) {
    ccr5.final[i, "95%-CI"] <- hr.ci(as.numeric(ccr5.final$coef)[i],
        as.numeric(ccr5.final$se.coef)[i])
}
write.table(ccr5.final, file = "CCR5_H1H1_FINAL_wMETA.txt",
    sep = "\t", quote = F, row.names = F, col.names = T)
```

```
library(stringr)
library(grid)
library(gtable)
library(gridExtra)

CG_DBMT <- read.table("~/Google Drive/OSU_PHD/dissertation/code/chapter2/SNP_CGwDBMT_20170522.txt",
    sep = "\t", header = T, stringsAsFactors = F)
CG_DBMT.sig <- CG_DBMT %>% filter(Significance ==
    "Significant")
DBMT <- CG_DBMT.sig %>% select(Gene:Graft, Outcome,
    Genome, SNP, Pvalue_D.BMT, N_D.BMT) %>% mutate(Group = "DISCOVeRY-BMT") %>%
    rename(numPvalue = Pvalue_D.BMT, N = N_D.BMT)
fig.tbl <- CG_DBMT.sig %>% select(Gene:Graft,
    Outcome, Genome, SNP, numPvalue, N, Group) %>%
    bind_rows(DBMT) %>% mutate(log_pval = -log(numPvalue,
    base = 10))
mac <- read.table("~/Google Drive/OSU_PHD/dissertation/code/chapter2/MacMillan2003.txt",
    header = T, sep = "\t", stringsAsFactors = F)
mac <- mac %>% mutate(log_pval = -log10(numPvalue),
    Group = ifelse(Group == "Replication in DISCOVeRY-BMT",
        "DISCOVeRY-BMT", "Replication"))
fig.tbl <- rbind(fig.tbl, mac)
plotFun <- function(repval) {
    sub.genes <- fig.tbl %>% filter(Group == repval) %>%
        pull(Gene)
    fig.tbl <- fig.tbl %>% filter(Gene %in% c(sub.genes,
        "CCL2", "MIF"), Group %in% c(repval, "DISCOVeRY-BMT"),
        SNP != "rs2066842")
    fig.tbl$Group <- as.factor(fig.tbl$Group)
    levels(fig.tbl$Group) <- c(paste0(repval,
        " in DISCOVeRY-BMT"), "Literature")
    if (repval == "Replication") {
        group.facet.labs <- c(`Replication in DISCOVeRY-BMT` = "Replication in \n DISCOVeRY-BMT",
            Literature = "Literature")
    } else if (repval == "Validation") {
        group.facet.labs <- c(Literature = "Literature",
            `Validation in DISCOVeRY-BMT` = "Validation in \n DISCOVeRY-BMT")
    }
    pR <- fig.tbl %>% mutate(SNP = str_replace(SNP,
        "D-R group 3 vs D-R group 1", "D-R group 3 vs\n D-R group 1"),
        SNP = str_replace(SNP, "D-R group 3 vs D-R group 2",
            "D-R group 3 vs\n D-R group 2"), Outcome = str_replace(Outcome,
            "DD", "DRM"), Genome = str_replace(Genome,
            "^R$", "Recipient"), Genome = str_replace(Genome,
            "^D$", "Donor"), Genome = str_replace(Genome,
            "^S$", "Mismatch"), Gene = str_replace(Gene,
            "NOD2", "NOD2/\nCARD15")) %>% ggplot(aes(SNP,
        log_pval)) + geom_hline(yintercept = -log10(0.05),
        color = "red") + geom_point(aes(color = Genome,
        shape = Outcome, size = N), stroke = 1.5) +
        facet_grid(Group ~ Gene, scales = "free_x",
            space = "free", labeller = labeller(Group = group.facet.labs)) +
        guides(size = FALSE) + theme(rect = element_rect(fill = "white"),
        legend.key = element_rect(fill = "white"),
        panel.border = element_rect(colour = "gray85",
            fill = NA), panel.background = element_rect(fill = "white"),
        panel.grid.major = element_line(colour = "gray85"),
        plot.background = element_rect(fill = "white"),
        axis.text.x = element_text(hjust = 1,
            size = 18, angle = -90), axis.text.y = element_text(hjust = 1,
            size = 18, angle = -90), strip.text.x = element_text(size = 18,
```

```
            face = "italic", angle = -90), strip.text.y = element_text(size = 18,
            angle = -90), axis.title.y = element_text(size = 18,
            angle = -90), axis.title.x = element_text(size = 18,
            angle = -90), panel.spacing.x = unit(0.1,
            "lines"), panel.spacing.y = unit(0.1,
            "lines"), legend.position = c(0.05,
            0.565)) + scale_y_reverse() + ylab("-log10(Pvalue)") +
        scale_shape_discrete(solid = FALSE, guide = guide_legend(direction = "horizontal",
            title.position = "right", title.theme = element_text(angle = -90,
                size = 18), label.position = "bottom",
            label.hjust = 0.5, label.vjust = 0.5,
            label.theme = element_text(angle = -90,
                size = 18))) + scale_color_discrete(guide = guide_legend(direction = "horizontal",
        title.position = "right", title.theme = element_text(angle = -90,
            size = 18), label.position = "bottom",
        label.hjust = 0.5, label.vjust = 0.5,
        label.theme = element_text(angle = -90,
            size = 18)))


    xr <- ggplotGrob(pR)
    # labelR <- 'Pvalues'
    labelT <- "Genes"
    posT <- subset(xr$layout, grepl("strip-t",
        name), select = t:r)
    height <- xr$heights[min(posT$t)]
    xr <- gtable_add_rows(xr, height, min(posT$t) -
        1)
    # Construct the new strip grobs
    stripT <- gTree(name = "Strip_top", children = gList(rectGrob(gp = gpar(col = NA,
        fill = "grey85")), textGrob(labelT, rot = -90,
        gp = gpar(fontsize = 18, col = "grey10"))))
    # Position the grobs in the gtable
    xr <- gtable_add_grob(xr, stripT, t = min(posT$t),
        l = min(posT$l), r = max(posT$r), name = "strip-top")
    # Add small gaps between strips
    xr <- gtable_add_rows(xr, unit(1/5, "line"),
        min(posT$t))
    # Draw it
    grid.newpage()
    grid.draw(xr)
}

plotFun("Replication")
plotFun("Validation")
```

RegulomeDB of Candidate SNPs

```
library(tidyverse)
x <- readxl::read_xlsx("~/Google Drive/OSU_PHD/dissertation/code/chapter2/repval_supps.xlsx",
    sheet = 1, na = "NS")  ## this is supp table 1 from CG paper
x <- x %>% mutate(P.Value = as.numeric(P.Value),
    pval_cat = case_when(is.na(P.Value) ~ "Not Significant",
        P.Value >= 0.05 ~ "Not Significant", P.Value <
            0.05 ~ "Significant")) %>% select(SNP,
    pval_cat) %>% distinct()
regdb <- read_delim("~/Downloads/RegulomeDB.dbSNP141.lessCol.txt",
    delim = " ", col_names = FALSE)
colnames(regdb) <- c("chr", "pos", "SNP", "RegDB_Score")
```

```
# big file takes up too much memory
rm(regdb)
cg.regDB <- regdb %>% right_join(x) %>% distinct() %>%
    group_by(RegDB_Score, pval_cat) %>% summarise(Count = n()) %>%
    ungroup() %>% rename(RegulomeDB_Score = RegDB_Score) %>%
    mutate(Freq = Count/sum(Count))
write.table(cg.regDB, file = "~/Google Drive/OSU_PHD/dissertation/code/chapter2/cg_regulomedb.txt",
    sep = "\t", quote = FALSE, row.names = FALSE,
    col.names = TRUE)
cg.regDB <- read.table("~/Google Drive/OSU_PHD/dissertation/code/chapter2/cg_regulomedb.txt",
    header = TRUE, sep = "\t")
p <- cg.regDB %>% na.omit() %>% ggplot(aes(x = fct_rev(RegulomeDB_Score),
    y = Count, fill = pval_cat)) + geom_col() +
    scale_fill_manual(values = c("black", "gray70")) +
    guides(fill = guide_legend(title = NULL),
        cex = 1) + xlab("RegulomeDB Score") +
    ylab("Number of previously reported SNPs") +
    labs(fill = "Reported Association to Survival") +
    coord_flip() + theme_bw() + theme(legend.position = c(0.7,
    0.9), legend.text = element_text(size = 14),
    axis.text = element_text(size = 14), axis.title = element_text(size = 16))
p
```