POST GWAS ANALYSIS: FRAMEWORK, METHODS AND APPLICATIONS TO BLOOD PRESSURE SENSITIVITY STUDY ON WEIGHT AND SODIUM CHANGE

BY JIE LIU

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Statistics and Biostatistics

Written under the direction of

Javier Cabrera

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

January, 2015

ABSTRACT OF THE DISSERTATION

POST GWAS ANALYSIS: FRAMEWORK, METHODS AND APPLICATIONS TO BLOOD PRESSURE SENSITIVITY STUDY ON WEIGHT AND SODIUM CHANGE

by Jie Liu

Dissertation Director: Javier Cabrera

Genome-wide association studies (GWAS) have gained popularity in the past few years. Researchers have made findings on identifying genetic variants as risk factors for biological traits. A further question is how to apply the results to real-life applications. In this dissertation, a framework is proposed to conduct a post stage GWAS analysis, connect GWAS findings with existing clinical trials, and provide useful information to doctors and practitioners to help patients. The key part of this framework is to incorporate GWAS results with clinical information and perform appropriate analysis with the combined data. We illustrate the application to the Trial of Non-pharmacologic Interventions in the Elderly (TONE). TONE is a clinical trial for elderly with high blood pressure, in which patients were randomized to receive intensive intervention in weight if they were obese, sodium intake reduction, both weight and sodium control, or placebo. We investigate the relationship of 21 polymorphisms, which are reported to have association with hypertension, diabetes or obesity, with the change in systolic blood pressure at the end of the trial. The objective is to find the people who would significantly benefit from such interventions. For the analysis of data, we propose two

approaches under the Post GWAS framework: recursive partitioning tree and exhaustive search. The recursive partitioning algorithm is a binary tree based algorithm that assigns different functionalities to SNP data and clinical data in the tree construction. We fit the tree to the data and examine the sensitivity of blood pressure drop given weight loss or sodium reduction. We compare classical regression tree with our modification to emphasize the differences in their structures. Tree methods are easy to interpret and compute, but only investigate a subset of the feature spaces. Exhaustive search is proposed to overcome this disadvantage. We look at all possible combinations of genotypes with sufficient sample and compute the sensitivities. We control multiplicity by the permutation version of false discovery rate method. Multidimensional scaling is used to determine the maximum number of polymorphisms to consider. Finally, we report and interpret the results from recursive tree and exhaustive search models.

Acknowledgements

I would like to express the sincere gratitude to my advisor Professor Javier Cabrera for continuous support during my Ph.D. study. He provided me invaluable advice and guided me through the research. His motivation, optimism and hard working attitude encouraged me to overcome all difficulties in the dissertation. Without him, this dissertation would be impossible. His help is even far beyond my research. He shared his social connections with me, encouraged me to enhance communication skills, introduced industrial opportunities, gave advices on career directions and much more. I am truly fortunate to have him as my Ph.D. advisor.

I would like to thank Professor John Kolassa, our graduate advisor. He devoted time and effort to help all graduate students to solve all kinds of issues, including course selection, financial sources, TA/GA appointments and so on. He is very nice and always ready to help us. I deeply appreciate him.

I would like to thank all other faculty members, staffs, and fellow Ph.D. students. Professor David Tyler's courses trained me in statistical foundation, multivariate and regression analysis. Professor William Strawderman and Harold Sackrowitz enhanced my background in estimation and inference. Assistant Professor Han Xiao gave us excellent lectures in Data Mining, from which I benefited a lot. There are so many courses offered in the department that I can not list all lecturers. It is my honor to have the chance to learn from them. Friendship with other students brought much joy and happiness into my life.

I would like to thank Professor John Kostis, Director of Cardiovascular Institution, Robert Wood Johnson Medical School. He provided the dataset and computing resource for the project. With other colleges, he used his insight to help us understand the biomedical background of the data and propose the questions more appropriately.

Finally, I would like to thank Jerry Q Cheng, Assistant Professor of Cardiovascular

Institution, Robert Wood Johnson Medical School. He provided numerous help when we worked together toward projects. His knowledge and patients helped me and encouraged me to overcome difficulties in research. It is my pleasure to work with him.

Dedication

To my father and mother

Table of Contents

Abstract ii

Acknowledgements iv

Dedication vi

List of Tables ix

List of Figures x

1. Introduction 1

2. Post GWAS and TONE 5

2.1. Post Genome Wide Association Study 5

2.1.1. Motivation 5

2.1.2. Framework 7

2.2. Trial of Nonpharmacologic Interventions in the Elderly 9

2.2.1. Clinical Experiment 9

2.2.2. Data Collection 10

2.2.3. Variable Selection 13

2.3. Application of Post GWAS Framework 15

3. Recursive Tree Approach 18

3.1. Review of Early Stage Analysis on TONE 18

3.2. Recursive Tree Model 20

3.2.1. Review of Classical Recursive Regression Trees 20

3.2.2. Algorithm of Classical Recursive Regression Trees 23

3.3. Modified Recursive Regression Trees 32

3.3.1. Choice of Splitters 33
3.3.2. Split Variable Selection 34
3.3.3. Within Node Regression 36
3.3.4. Control Parameters 36
3.3.5. Test Procedures 37
3.3.6. Comparison with Traditional Regression Trees 41
3.3.7. Results 43

4. Exhaustive Search Approach 47
4.1. Motivation 47
4.2. Methods 48
4.2.1. Binary Representation for a Single Snip 49
4.2.2. Binary Representation for a General Genotype 50
4.2.3. Redundant Paths 53
4.2.4. Practical Issues 54
4.2.5. Tetrasexagesimal Encoding Scheme 55
4.2.6. Ranking of Paths 58
4.2.7. Termination Levels: Multidimensional Scaling 60
4.2.8. Results of MDS 64
4.3. Results and Discussions 70
References 78

List of Tables

2.1. The distributions of 21 Polymorphisms in All Treatment Groups 11

2.2. The distributions of 21 Polymorphisms in Weight Loss Group 12

2.3. List of Selected Input Variables 15

3.1. Controlling Thresholds of Nodes' Minimal Sizes 37

4.1. Binary Representation of a Single SNP 50

4.2. Matrix Representation of A Sample with 5 Obervations 51

4.3. Parameters of Nodes Sizes 54

4.4. Mappings of Base64 Characters 56

4.5. Top 3 Positive Paths with Last 3 Digits in Names Displayed 71

4.6. Distributions in Weight Loss Group for Overlapped SNPs 73

List of Figures

2.1. Post GWAS Data Streams 8

3.1. A Simple Example of Regression Tree 23

3.2. Recursive Tree Result for Weight Loss Group 45

3.3. Recursive Tree Result for Sodium Reduction Group 46

4.1. An Example of Base64 Compression 57

4.2. Change to Storage Space of Processed SNP Data 58

4.3. Histogram of P-values before FDR Adjustment 60

4.4. Example of Permutations for FDR 61

4.5. Number of Paths v.s. Number of Polymorphisms 61

4.6. 2 SNPs MDS Result for Weight Loss Group 65

4.7. 2 SNPs MDS Result for Weight Loss Group 66

4.8. MDS Result for Top Paths of Length 2&3 67

4.9. MDS Result for Top 100 Paths of Length 4 69

4.10. MDS Result for Top 300 Paths of Length 4 70

4.11. MDS Result for Top 400 Paths of Length 4 71

4.12. MDS Results for Top 500, 1000, 1500 and 2000 Paths of Length 4 72

4.13. MDS Result for Top 1500 Paths of Length 5 73

4.14. MDS Result for Top 3000 Paths of Length 5 74

4.15. MDS Result for Top 3000 Paths of Length 6 75

4.16. Adjusted P-values for Top 30 Paths of Length 4 75

4.17. Top Significant Set in Paths of Length 4 76

4.18. The Second Top Significant set in Paths of Length 4 76

4.19. Distribution of SNPs in Top 200 Paths of Length 4 by Clusters 77

Chapter 1

Introduction

Genome-wide association studies (GWAS) have gained popularity in the past few years. Decades ago, high cost in conducting gene-level studies and limited biological technology restricted the research in genetic field; only a small fraction of the genome regions were explored. As a consequence, without genome wide information of human cells, GWAS was impossible to conduct. Things changed with the exploration in human cells. In early 2000s, the Human Genome Project (HGP) was declared to be finished. The primary goals of this project were to find all the genes in human DNA, which was estimated to be 20,000 - 25,000; and also sequence the human DNA, which consists of approximately 3.3s billion chemical base pairs as the component units. This project was suggested in late 1980s. Department of Energy (DOE) and National Institutes of Health (NIH) have coordinated and controlled project activities since 1990. In the initial plan, the project would last for 15 years, partitioned into 3 stages evenly in time. The first stage was done in 1993, earlier than anticipated. The final draft was published in 2003. In terms of the goal on DNA sequencing, with 99.99% accuracy, more than 90% of gene-containing part of human sequence had finished. At the mean time, there were other objectives, including developing faster sequencing technologies, finding efficient tools for data analysis and solving potential ethical, legal and social concerns raised by the project [Schmutz et al., 2004].

The completion of the Human Genome Project is a great feat of exploration in history. Each individual hereditary characteristics is determined by the sequence of nucleotides, and HGP determined the order in our genome's DNA. The results of the

project made maps which display the locations of genes in the major part of the chromosomes, which provides investigators with the resource of information about the structure, components and function of human genes. HGP allows the possibility of Genome Wide Association Studies (GWASs). It is known that single nucleotide polymorphism (SNP) is the unit of genetic variation that occurs in the DNA sequence on different individuals of a biological species. GWAS examines such common genetic variants among people and try to detect if any variant is related to certain trait. The interested trait can be common diseases as type II diabetes or rare ones such as sickle cell anemia. One primary objective of GWAS studies is to identify some genetic variants as risk factors for the traits. The results can be used in multiple ways: making predictions on a person whether he is at more risk for a disease, or providing biological foundations for finding solutions to prevent or treat the disease. Human Diseases are commonly studied traits, see, for example, [Zeggini et al., 2008], [Samani et al., 2007]. But there are many more of interests, and some are for other species: [McCarthy et al., 2008], [Goddard and Hayes, 2009] and [Heffner et al., 2009]. Klein's study is considered to be the first published GWAS article [Klein et al., 2005]. They identified a variant from over 11,000 SNPs having a strong association with age-related macular degeneration. Later the number of GWASs increased rapidly and researchers looked into larger pool of SNPs with more sample size. Wellcome Trust Case Control Consortium (WTCCC) conducted a study in 2007 covering 14,000 cases and 3000 controls [Burton et al., 2007], which is a significant lift of size comparing to 96 cases and 50 controls in Klein's study. As of July 20, 2014, in NIH's GWAS catalog, 1928 publications and 13432 SNPs are included.

While GWASs were becoming popular, it also gained opponents, see, for example, [Wray et al., 2007] and [Visscher et al., 2012]. One type of criticisms are about the functionality of GWASs' findings. A majority of reported SNPs in GWAS studies does not have practical utility for diseases treatment or prognosis, according to some researchers [McClellan and King, 2010]. In this article, we propose a solution to utilize the findings from GWASs and illustrate the application with an clinical example. There are criticisms in other aspects as well. In early stages, researchers use some efficient

algorithm to scan the SNPs and find those who are statistical significant. For instance, one underlying assumption is that all SNPs function independently and make important impact to the diseases' risk. Another assumption is that the top significant SNP explains a large amount of variations [Yang et al., 2011]. It is also questioned that the finalized results only contain a handful variants which may be misleading given the large number of genes [Latham, 2011]. Discussion on these criticisms is beyond the scope of this article. In further chapters, we will accept as a fact that the methodology of GWAS is well established and the results are valid.

Efforts have been devoted to face some criticisms and more advanced methods have been used. A typical GWAS data contains millions of SNPs with much smaller number of observations, in the order of thousands, which leads to a flat data matrix with more predictor variables than the number of subjects. For this type of $p >> n$ problems, statisticians have proposed many methods to solve them. Penalized regressions, such as LASSO [Tibshirani, 1996], Group LASSO [Yuan and Lin, 2006], elastic net [Zou and Hastie, 2005], all are popular choices. This class of methods has been reviewed in the context of GWAS [Szymczak et al., 2009]. In a simulation study [Waldmann et al., 2013], it has been showed that elastic net provided the best compromise between the false positive rates and power of detecting influential SNPs. Random Forest is another popular method to rank the impermanence of features (SNPs). It has been applied to GWAS with the objective of predicting rheumatoid arthritis with top ranked SNPs [Jiang et al., 2007]. Jiang also pointed out the limitation of the method within training-testing framework. Boosting Trees is another ensemble tree-based methods [Friedman et al., 2000], [Friedman, 2001]. Boosting tree classifiers have been used in a hierarchical way to make SNPs compete with each other and selected the winner as valuable SNPs as the result [Wan et al., 2009]. Wan also used this method to consider the interactions between SNPs instead of taking a univariate ranking approach. Network-based algorithms are used in GWASs, such as Neural Networks [Tomita et al., 2004] and Bayesian Networks [Sebastiani et al., 2005]. With the substantial number of published GWAS studies, a question is brought to us: how to utilize the information from GWASs.

One approach is using meta analysis [Glass, 1976]. Meta analysis seeks to combine the results from multiple independent studies [Chan and Arvey, 2012], and it has been applied to GWASs [Cantor et al., 2010], [Yesupriya et al., 2008]. In 2007, Lewinger proposed a hierarchical regression model as a further investigation to GWASs' results although the framework and simulation was still at the same stage of GWAS [Lewinger et al., 2007]. Bayesian meta analysis was performed [Newcombe et al., 2009], [Verzilli et al., 2008]. Other approaches include effects based, studies' weights based, p-value or q-value based and more. A comprehensive review of these approaches has been provided [Evangelou and Ioannidis, 2013]. A closer look of those studies reveals that they are still preforming the same task as GWAS in early stages [Stahl et al., 2010], [Speliotes et al., 2010]. To perform a real post stage of GWAS analysis, we propose a new framework and apply it to a specific data set. The motivation of our framework is to incorporate the reported SNPs from similar GWAS studies and a clinical trail which contains the measurement of corresponding genes from participants. The framework does not set any requirement on the methods to use. We will introduce two methods: the first one is a modified regression tree and the second is an exhaustive search approach. It can be viewed that the tree model is a reduced version of the exhaustive search. The data was obtained from Cardiovascular Institution, Robert Wood Johnson Medical School, directed by Dr. John Kostis, who conducted a comprehensive clinical trial on old people with hypertension with the objective of observing the blood pressure drop on the change of sodium and/or fat intake. The trial does not involve any pharmacologic intervention but weight loss training and sodium control programs. It was designed specifically for people of age greater than sixty. The ultimate goal of the research is to identify multiple genetic factors that influence the impact of weight and sodium change on the blood pressure and provide personalized medication or no-medication recommendation. The rest of the thesis will be organized as follows. In Charter 2, I will introduce our framework and the background of the data. Charter 3 and Charter 4 will cover the details of the tree model and exhaustive search model.

Chapter 2

Post GWAS and TONE

2.1 Post Genome Wide Association Study

2.1.1 Motivation

Researched who completed GWAS studies successfully are still facing a challenge: how to apply the results to real life applications. It might be expected that the findings provide guidance for disease diagnostics, treatment discover and drug development. Some SNPs have been used to improve the accuracy of prognosis. Other applications include improving the prediction accuracy of the mortality rate on patients after receiving isolated primary coronary artery bypass graft (CABG) surgery [Muehlschlegel et al., 2010]. But they only looked into a single genetic variant. On the other hand, some other post GWAS analysis are essentially at the same GWAS stage [Lewinger et al., 2007]. The results from GWAS are typically restricted to a limited number of genetic variants, usually less than 10. Therefore we can perform more refined analysis instead of being constraint by computation complexity. We also would like to collect significant SNP candidates from multiple GWAS studies with the same or similar traits.

One underlying reason is that GWAS studies assume the independence of all genetic variants, which may not hold in reality. It has been pointed out that there are three major disadvantages of single SNP GWAS [Li et al., 2011]. Firstly, human's biological traits are determined by polygenes, so the analysis on a single SNP may not explain a small proportion of variants. Secondly, multiple genes may have some structure to function as a unit rather than independently, which is not considered in a single SNP screening method. Finally, quite some GWASs were performed for different environments by some categorial variable such as gender, and then make a comparison of two

subgroups, which leads to a less powerful analysis. As mentioned in Charter 1, some researchers have developed various approaches for simultaneously analyzing multiple SNPs for GWASs [Baranzini et al., 2009].

It is know that there are strong association among some diseases, so when we consider which GWAS to include for a post stage analysis, we prefer to include similar biological traits. This promotes us to consider a border range of GWASs, with the idea of considering similar such studies, where similarity is defined to be commonly related pheno-traits. Our Post-GWAS approach will be built for single-SNP based analysis, but it is compatible to clustered SNPs results, where clustered SNPs refers to a set of interacted SNPs which affect certain disease together.

In our study, we are interested in high systolic blood pressure. We looked into associated traits including obesity and diabetes mellitus since they are commonly accompanied. What we did was searching for these three traits individually within GWAS research articles and collected all reported SNPs. In our methodology, the number of genetic variants is more than that of the result of a typical GWAS study, but not as substantial as the whole data in GWAS studies. As a consequence, we do not need to request the algorithm to deliver results efficiently. We do not test on each SNP separately; instead, we are interested in the combination of multiple SNPs, which may have a stronger effect to the response. It differers from some multi-SNP GWAS in the way that we only take the reported SNPs into account, while those studies may use some technology of reducing the dimension, putting penalty on SNPs and finding the remained SNPs.

It is known that there are limitations of such regularized approach. For instance, LASSO would select one predictor from several correlated variables, which may not be the appropriated one. Group LASSO [Friedman et al., 2010a] was proposed to overcome this issue by adding penalized coefficients by groups, which requires a pre-step to identify such groups. It is a challenge to partition the SNP properly in the context of GWAS due to the extremely high dimension.

Therefore, we take an alternative approach by constructing a hierarchical structure. All details will be presented in the next charter. Another motivation of our approach

is to provided personalized guidance to disease prevention, control and treatment. Patients carrying different variants exhibit different treatment effects in clinical trails will be a data source of such investigation. Therefore a clinical trail with genetic information is required. The underlying models have to control for other aspects of the samples as well, in order to give more precise prediction on the genetic impact. Our implementations also meet this criterion and will be explained how we fit the requirements. With our methodology, we are able to look combinations of genotypes to investigate the impact on the sodium and weight sensitivities. Our methods also allow us to look at a large number of polymorphisms as the components of the genotypes.

2.1.2 Framework

We have explained the motivation of proposing our Post GWAS thoughts. In principle, there are three major components of this framework: SNP results collection, clinical data collection, statistical analysis on the integrated information. We summarize our framework below:

Post-GWAS Analysis Framework:

- Incorporate SNP data to clinical trial data
  - Merge the list of significant SNPs from GWAS studies for certain trait
  - Identify a suitable clinical trial and collect SNP data for participants
  - Model the data
- Apply SNP data to recognize genotype subgroups to be analyzed

Any GWAS research plan will engage certain phenotype traits. According to the article engine by National Human Genome Research Institute, there are thousands diseases to be searched for, in a database of about 300 journals. A typical study will only engage one disease. In our study, we are interested in finding genetic variants for high systolic blood pressure patients. Due to the motivation of including associated traits, we also looked into obesity and diabetes mellitus. A literature search was performed [Kostis et al., 2013] and selected 21 polymorphisms from the Wellcome Trust Case Control Consortium.

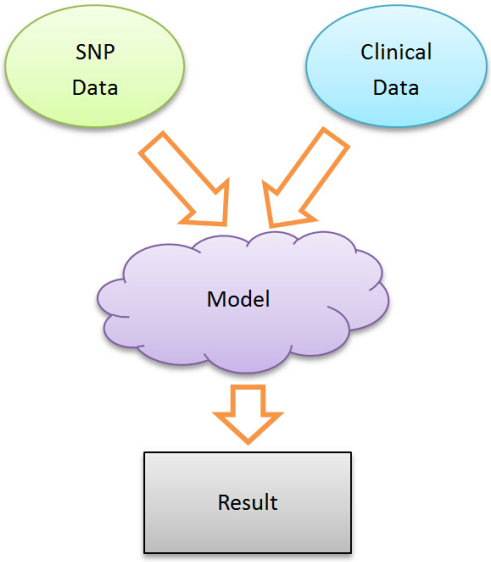

Figure 2.1: Post GWAS Data Streams

After identifying the key genetic variants, the framework also requires the collection of SNPs' information of participants in a suitable clinical trial. In a early stage analysis [Kostis et al., 2002], the impact of angiotensin-converting enzyme insertion-deletion polymorphism on the change of blood pressure given weight loss. Genetic Data for 7 polymorphisms out of 21 were analyzed by the blood samples in this study after informed consent. The rest of the SNPs' information was collected in 2011 for the same participants. In principle, many clinical trials would keep the blood samples such that they would be available for further investigation, which allows the second part of our framework to be feasible. We have implied the clinical trial that functions as a data source.

In principle, researchers may look into recently finished trails which may or may not focus on any polymorphism in the gene set from the first step of post GWAS. As long as a trail share the same or similar objective, and have the blood sample or DNA well kept, it fits the requirement to enter the second stage. The motivation of selecting a clinical with similar objective is: we need both the genetic data as well as the biological profiles of observations. Although analysts tend to collect phenotype information as more as possible, some key information might be missing if the original trial is highly unrelated

to current objective while the some of rest input variables may be not informative to explain the response. At this step, researchers are encouraged to use the own judgement to see if a clinical can be well-incorporate to the post stage GWAS analysis. It would be ideal if the trail focused on any single or multiple SNPs of the candidate list since some of the genetic information would be available, without requiring a detection on existing SNPs, which will accelerate the research period and save some budget. Even the selected trial does not contain the required SNP information, with rapidly developed biotechnology, it is still feasible to perform the experiment for a reasonable number of polymorphisms. Although the way we performed this analysis does not follow the framework precisely by having the trail identified first and then proposed the post stage analysis, we still do not have all the SNP data from patients which promoted us to retest the blood sample and got the 14 variants analyzed.

2.2 Trial of Nonpharmacologic Interventions in the Elderly

The clinical trail that we included in the study was for testing whether the reduction in sodium intake and/or the training of weight loss will significantly help the obese patients to lower the blood pressure. If the non-pharmacologic interventions in sodium and weight were proven to be effective, then the need for antihypertensive drugs from the patients would be reduced. Another motivation for this trail is that in past research studies, most were focused on the effect of salt sensitivity extensively while weight effect had not been investigated sufficiently [Whelton et al., 1998]. The trail intended to learn how old people with obesity would benefit from those interventions, therefore all interventions did not involve any pharmacologic treatment.

2.2.1 Clinical Experiment

TONE focused on the sensitivity of old people and admitted 975 participants aged between 60 and 80. Other conditions require the participants being on one or two antihypertensive medications, systolic blood pressure less than 145 $mmHg$ and diastolic blood pressure less than 85 $mmHg$. People with serious cardiovascular issue,

insulin-dependent diabetes mellitus, mental or physical illness were excluded. Qualified participants were distributed in four medical centers. The overweight criterion is male with the body mass index greater than 27.8 kg/m^2 and female with BMI greater than 27.3 kg/m^2. There were 585 participants falling into this region while 390 were considered to be not over weighted.

The interventions were a weight loss program, a sodium reduction program, or a combination of them. The goal of weight program was losing more than 4.5 kg and the goal of sodium program was a daily dietary salt intake less than 1800 mg measured by 24-hour level in urine. The intervention included necessary education to enhance the understanding of behavior change to achieve those goals under the guidance of social action theory. Each intervention consisted of 3 stages of period. The first intensive stage provided participants with the courses and trainings such that participants were educated with required knowledge for achieving goals. The knowledge contained details such as what to eat, how to cook, how much exercises recommended, etc.

It was a four-month stage and drug withdrawal was attempted after three months of the non-pharmacologic intervention. The next four month was the extended phase at which participants devoted to the prevention of relapse and solve some issues from the first stage. The last maintenance was mainly for the retention of remained participants. Doctors measured the blood pressure and sodium level before and during the drug withdrawal period. The final measurement minus that at the first drug withdrawal was recorded as the changes in blood pressure or sodium as our responses for data analysis. Overweight group were assigned randomly to weight loss only, sodium reduction only, the combined treatment, or placebo. The non overweight group was assigned to sodium control or placebo. Participants were removed when the drug withdrawal caused high blood pressure and medication was required or some other cardiovascular problem occurred.

2.2.2 Data Collection

Qualified participants' blood samples were collected for genotyping after informed consent and DNAs were extracted from the sample with a extraction kit and stored at

Polymorphism	Heterozygous	Homozygous mutant	Wildtype
rs5186	56% (405)	9% (65)	35% (252)
rs4646994	51% (366)	17% (122)	32% (234)
rs7961152	46% (333)	21% (151)	33% (238)
rs1800896	46% (332)	20% (145)	34% (245)
rs19822073	45% (328)	37% (265)	18% (129)
rs187238	44% (319)	7% (52)	49% (351)
rs699	48% (347)	23% (165)	29% (210)
rs1800795	36% (263)	11% (80)	52% (379)
rs1799983	39% (279)	6% (45)	55% (398)
rs6997709	38% (273)	6% (43)	56% (406)
rs5443	38% (275)	22% (160)	40% (287)
rs1937506	35% (253)	6% (45)	59% (424)
rs1800872	38% (277)	8% (59)	53% (386)
rs5370	32% (230)	4% (32)	64% (460)
rs1800629	24% (172)	2% (18)	74% (532)
rs2820037	32% (229)	5% (38)	63% (455)
rs5744292	28% (202)	3% (25)	69% (495)
rs4684847	19% (135)	1% (7)	80% (580)
rs11110912	25% (179)	2% (16)	73% (527)
rs4961	26% (190)	4% (28)	70% (504)
rs1800796	12% (90)	0% (2)	87% (630)

Table 2.1: The distributions of 21 Polymorphisms in All Treatment Groups

a proper temperature. GeneScan software applied a automated fluorescent scanning detector to obtain DNA fragments. Finally we obtained SNPs' information of 722 observations' as well as their clinical information. The distribution of 21 snips among all observations and weight loss subgroup are shown in Table 2.1 and 2.2 respectively. We have 722 observations in total and 112 in weigh loss group.

The clinical raw data contains over 200 input variables, covering a wide range of participants' profile, such as age, race, smoke history, cardiovascular disease history, other diseases, family factors, and so on. We will not use all of the covariates due to our limited sample size. Preprocessing step was taken to select most influential variables. We performed Elastic Net to reduced the size of data from clinical centers for further analysis.

Polymorphism	Heterozygous	Homozygous mutant	Wildtype
rs5186	58% (65)	9% (10)	33% (37)
rs4646994	52% (58)	16% (18)	32% (36)
rs7961152	50% (56)	21% (23)	29% (33)
rs1800896	49% (55)	21% (24)	29% (33)
rs19822073	47% (53)	35% (39)	18% (20)
rs187238	46% (52)	3% (3)	51% (57)
rs699	41% (46)	23% (26)	36% (40)
rs1800795	39% (44)	12% (13)	49% (55)
rs1799983	38% (42)	1% (1)	62% (69)
rs6997709	38% (43)	3% (3)	59% (66)
rs5443	37% (41)	27% (30)	37% (41)
rs1937506	35% (39)	8% (9)	57% (64)
rs1800872	33% (37)	10% (11)	57% (64)
rs5370	31% (35)	4% (4)	65% (73)
rs1800629	30% (34)	3% (3)	67% (75)
rs2820037	29% (33)	4% (4)	67% (75)
rs5744292	29% (32)	2% (2)	70% (78)
rs4684847	19% (21)	1% (1)	80% (90)
rs11110912	19% (21)	1% (1)	80% (90)
rs4961	16% (18)	4% (4)	80% (90)
rs1800796	13% (15)	0% (0)	87% (97)

Table 2.2: The distributions of 21 Polymorphisms in Weight Loss Group

2.2.3 Variable Selection

Elastic Net [Zou and Hastie, 2005] was proposed as an improvement to Least Absolute Shrinkage and Selection Operator (LASSO)[Tibshirani, 1996]. Linear Regression had been widely used in the history of statistics and gained popularity. It connects the response variable through a linear relationship to the input variables. It is easy to understand and interpret, can be used for both explaining quantitative response and predicting the potential observed values given a new input observation. It is closely related to hypothesis testing and inference. It incorporates the most common Gaussian Distribution and can be viewed as an optimization question. Its solution can be obtained flexibly by iterative method or least square method. It has easy-to-see geometric interpretation and is related to linear algebra and matrix theories. Under the normal assumption, the estimator is identical to maximal likelihood estimator and has asymptotic properties.

With all features above, linear regression are being used exclusively even it has some disadvantages. Tibshirani pointed out two issues: prediction accuracy and interpretation of a large number of predictors. The prediction issue is caused by the fact of using all explanatory variables. Even there is no overfitting, a handful number of vectors can span the space well in terms of the dimension, as a consequence, the bias of the model is small. Considering the trade off between bias and variance, linear regression tends to have less bias but higher variance. Subset selection and ridge regression are two standard methods to improve the least square estimates. But subset selection is very sensitive to the data, while ridge regression shrinks the coefficients continuously and does not select a nested model. Tibshirani replaced the L^2 penalty term by L^1 norm, therefore imposed the discrete variable selection process and reduced the variance by sacrificed some bias. It is a combination of continuous shrinkage and automated variable selection. A more detailed comparison and an alternative Bayes point of view can be found in [Friedman et al., 2010b].

Three scenarios were discussed where LASSO has limits [Zou and Hastie, 2005]. If

there are more variables than observations, due to the natural of linear regression, LASSO can select at most the same number of variables as that of the observations; suppose the penalty parameter for selecting such a sub model is λ_0, when the penalty is greater than λ_0, the model is over parameterized and LASSO is not properly defined. The second situation is related to collinearity. If there are a collection of highly correlated variables, LASSO tens to choose any one from them. In this situation, it was also illustrated that ridge regression outperforms LASSO. [Zou and Hastie, 2005] proposed a naive elastic net first, in which the penalty is the sum of L^1 and L^2 norms of the parameters with a tuning parameter respectively. It solved the first two problems mentioned above. But it shrinks the parameter more than necessary by introducing more bias, so the non-naive version of elastic net was to correct back this bias by relaxing the penalty a little.

We have more than 200 predictors in the clinical data set, which requires a variable selection. The reason is that although we have 722 participants in total, they were assigned into four different groups. More details will be included later but some of the subgroups contained less than 200 observations. With prior knowledge of the analysis, those variables might be classified into two pools: some have to be included as the key features, others are to be selected by the shrinkage method. The reason to force several features to stay in the minimal model is that we planned to look into the sensitivity of blood pressure change versus weight or body mass index change, therefore the difference of BMI before and after the drug withdraw has to be included. Here we used the before measurement minus the after, such that this quantity is positive for most people and we did the same to the blood pressure change.

Other key variables for the weight loss group are baseline BMI, age and gender. For sodium involved group, the sodium change, baseline sodium, age and gender were listed in the first pool of variables. We only included those contain important information of participants's profile and limited it to a collection as small as possible. For instance, the cardiovascular disease history appears to be also importance by common sense but we would leave them for the elastic net to determine automatically. Ten variables were selected from the second pool and they are listed in the table as additional covariates.

Raw Name	Variable Type	Description
BMI	basic	baseline Body Mass Index
BMICHNG	basic	change of BMI from the baseline to last visit
RV_UNA	basic	baseline sodium level
CHUNA	basic	change of sodium level to the last visit
AGE	basic	age when participants entered the trail
MALE	basic	male of female Indicator
RV_DIABETES	additional	diabetes requiring insulin
RV_SMOKECUR	additional	smoker or non-smoker
RV_STRK	additional	experienced a stroke or not before the trail
RV_BLOCKS	additional	estimated number of blocks walked per day
RV_EXYEAR	additional	physical activity rate in a year
RV_MOD	additional	moderate activity hours per week
RV_LIGHT	additional	light activity hours per week
RV_DIAB_KID	additional	having a child with diabetes mellitus
RV_KDNY_POP	additional	father had renal failure or not

Table 2.3: List of Selected Input Variables

2.3 Application of Post GWAS Framework

In Section 2.1, we introduced the general Post-GWAS Analysis Framework. It is summarized below:

1. Specify the clinical trait to investigate
2. Search within related GWASs and summarize all reported SNPs
3. Identify a suitable clinical trial satisfying following properties:
   (a) Involved the trait as primary or secondary response
   (b) Blood samples are available for DNA detection
   (c) Desired to include some of the SNPs that were found in GWASs but not required
4. Analyze the participants' blood sample and obtain their genotypes of interested SNPs
5. Perform statistical analysis on the incorporated clinical and genetic data sets

We may be aware that the details of the last part is still not presented. For our TONE experiment, the way we performed analysis is listed below:

1. Using SNP information to generate subgroups of patients
2. Find the sensitivity of blood pressure change vs BMI/Sodium change for a particular genotype and all participants
3. Make comparison of the geno-specified subgroup with the overall level
4. Identify the most/leaset sensitive subgroups

Since we have the genetic information of all participants, we were able to partition them into subgroups based on different genotypes. It is also a natural thought to compare the sensitivity across genotypes. Our post GWAS shared common procedures as the GWAS. After comparing the genotyped subgroup, they would be ranked by the adjusted statistics. One difference is that our approach is not limited to the single SNP comparison and can be extended to a combination of several SNPs. In next two chapters, we would introduce two solutions for such incorporated framework. The way of generating genotypes with multiple SNPs can be viewed as hierarchical steps: starting from one SNP, which is corresponding to certain subgroup of the sample, and add another SNP, which may eliminate some from the subgroup, and keep adding more. We would discuss how to determine the total number of SNPs in a genotype combination later. This hierarchical adding procedures lead us to consider models with a similar structure. It gives us the motivation to consider hierarchical tree models.

Based on the type of response variables, a single tree model may be a classification tree or a classification tree. No matter which type tree it is, it starts with a top root node containing all sample points. And then the algorithm selects a variable with a region to split the node into two children nodes. Keep partitioning the children iteratively and a tree is obtained finally. It is widely used in supervised learning field. The binary splitting is simplified but works with both categorial or continuous inputs. The structure of the tree is clear and easy to interpret. There are details about how to determine the splitting variable, when to terminate and some of them will be introduced when the model is reviewed in next chapter.

Researchers also proposed ensemble ways to improve the performance of a single tree, such as bagging, random forest, boosting, etc. Please be aware that all these versions of

tree models were designed for supervised training problems. Such problems commonly have a training data including the input objects and a output variable. The output variable is also called supervisory signal, which gives the modeler a way to estimate the parameters and evaluate the performance with a loss function properly defined for the type of supervisory variable. Our study was the sensitivity of blood pressure change to the BMI/sodium change, which can be understood as a slope quantity. It is not observed directly, therefore the tree model can not be applied to our problem directly. We had to model this sensitivity first and then built the tree; this process also affected the way of choosing the splitting variable and other aspects of tree construction.

Chapter 3

Recursive Tree Approach

3.1 Review of Early Stage Analysis on TONE

Before performing the Post GWAS study on this data, we investigated the same question with GWAS-like analysis [Kostis et al., 2013]. We regressed the systolic blood pressure on independent variables including clinical covariates and genetic polymorphisms. We classified the independent variables into five categories:

- body mass index related covariates: baseline BMI and its change
- sodium related covariates: baseline urine sodium and its change
- genotypes: there were 21 polymorphisms where each one is a 3-level nominal variable. Every polymorphism of a participant could be wild type, heterozygous and homozygous mutant, denoted as 1, 2 and 3 respectively
- predetermined essential covariates: age and gender
- selected additional clinical variables: elastic net was performed on all raw inputs and obtained a list of final additional variables

The Tone data contains normal and obese participants. All were randomized to receive different treatments. Therefore participants were mutually partitioned into six groups:

- Over weight participants, received both sodium reduction and weight loss treatments
- Over weight participants, only received weight loss treatment
- Over weight participants, only received sodium reduction treatment

- Over weight participants, control group
- Normal weight participants, only received sodium reduction treatment
- Normal weight participants, control group

We reorganized these six groups in to seven collections and performed all analysis on each collection. The underlying reason is that we would like to investigate weight sensitivity and salt sensitivity individually and jointly. Therefore for salt sensitivity analysis, we were promoted to analyze all participants received salt reduction treatment, solely or in a combined program, and for weight loss, we combined those got weight loss training. We also investigated two sensitivities simultaneously in the model.

The model we used at this step is linear regression. The input variables varied to include in the model. We looked at the individual effect brought by a single polymorphism as well as all together. We also considered the situation where the investigator-selected variables were added or not. Our primary response is systolic blood pressure change and the secondary is diastolic blood pressure change.

- Model 1: One polymorphism at the time + Baseline BP + BMI Change+ Baseline BMI + Sodium Change + Baseline Sodium (examines weight and salt sensitivity)
- Model 2: All polymorphisms + Baseline BP + BMI Change+ Baseline BMI + Sodium Change + Baseline Sodium (examines weight and salt sensitivity)
- Model 3: One polymorphism at the time + Baseline BP + BMI Change+ Baseline BMI (examines weight sensitivity)
- Model 4: All polymorphisms + Baseline BP + BMI Change+ Baseline BMI + (examines weight sensitivity)
- Model 5: One polymorphism at the time + Baseline BP + Sodium Change + Baseline Sodium (examines salt sensitivity)
- Model 6: All polymorphisms + Baseline BP + Sodium Change + Baseline Sodium (examines salt sensitivity)

When the analysis interest was systolic blood pressure sensitivity, all blood related variable were systolic related measurements and all were replaced with diabolic when we were investigating the secondary response. All significant associations between the SNPs and blood pressure change by regression models are summarized in table below.

3.2 Recursive Tree Model

Our regression analyses shared at least one drawback as the GWAS studies: contributions from multiple polymorphisms were not investigated. To solve this issue, we utilized the tree-based model. There are multiple ways to obtain the genotypes generated by several polymorphisms. One is directly assigning the indices to chose from the candidate pool. That would be a combinatorial number exploring quickly as the total number of polymorphisms increases. We will revisit it later in next chapter. Alternatively, one can produce the combination iteratively in a hierarchical way. There are advantages of it: iterative methods will give as many combinations as possible up to the capability of cpu and memory resources; it is relatively easy to control the total number of SNPs involved and it can be paralleled without too much effort if a regular desktop could not handle the computation. At this stage of analysis, we would like to trade off between exploring all possible combinations and limiting to a reduced space to make the computation feasible. The hierarchical structure and trade off consideration promoted us to consider binary tree based models. For a single tree model in supervised learning, there are classification tree and regression trees, depending on the type of response variables. What we did is closer to regression trees. We would like to review the idea of regression and extended it to our modified version.

3.2.1 Review of Classical Recursive Regression Trees

Tree is a data structure widely used in many fields. It is a basic data structure in computer science and has all varieties such as red-black tree, complete tree and more. We will not introduce them since only a binary tree would be the fundamental structure in our method. But there are some terminologies commonly used for trees.

- Node: a set containing values in the tree structure. They are typically linked together with edges.
- Edge: connection between one node to another.
- Root: the top node in a tree, positioned highest.
- Child: if two nodes are linked directly, then the one lower is the child of the higher.
- Parent: in a pair of directly linked nodes, the higher one is the parent.
- Siblings: nodes sharing the same parent.
- Descendant: a node reachable by repeated proceeding from parent to child.
- Ancestor: a node reachable by repeated proceeding from child to parent.
- Internal node: a node with at least one child, also called as intermediate node.
- Leaf: a node without any children, also called as terminal.
- Path: a sequence of nodes and edges connecting a node with a descendant.
- Level: the level of a node is defined by one plus the number of connections between the node and the root.

Let us consider the simplest case first. If I have a continuous variable and no inputs, then modeling the predicted value as the mean of observed responses under the context of regression. It used a indicator function as a estimator to the response and the choice of constant coefficient would be naturally the average. If there is one input variable, and assume it is orthogonal to the constant vector although it is not true in general, then two parts of the response variable are explained: the first part is the mean part which can be understood as the same as the no input case, the residual is explained by the unique input variable then. If the input covariate is not orthogonal to the constant vector, a upper triangle matrix serving as a linear mapping operator can transform the input columns and make them perpendicular. Therefore the regression can be

performed in a two stage manner. Thus I can consider the the scenario that there is a continuous response and a single input variable, and I would like to regress the response on the predictor without intercept term. The fitted model would be a straight line in the 2-d plain that minimizes the sum of squared errors among all lines passing through the origin. From the linear model, we can get even more complicated models such as local regression, polynomial regression, etc.

Smoothing is one desired property for these models. But we can also go backward a little bit. Recall that the classical way of introducing Lebesgue integration starts from simple functions, which are indicator functions on a bounded set; and then the summarization of simple functions forms step function. They can approximate any measurable function well in L_p space. We may relax the smoothing condition and use a step function as the prediction function of inputs. By adding more restrictions to the step function, the sum of squared error would be increased but we benefit from a very simple model and easy to interpret.

If we only allow the function take two different values on the whole x-axis, and only allow it have one jump in the function value, then this step function is the sum of two indicator functions with mutually exclusive supports and the union of their supports covers the whole domain. In this case, on each support, a constant is used to approximate the response, which would be the mean on observed points falling into the domain. It can be viewed as a binary prediction procedure. If the new observed input falls into one region, then its prediction response would be the mean on its area. It corresponds to a one-layer binary tree and the jump point of the step function are the splitting condition which determines where the new data point falls into. One reason that binary tree is popular is its easy interpretability. The whole data set is in the top root node. By using one feature, the data is partitioned into two parts and each part's response is close to a value of response; these averages are commonly distinct. The partition introduces one more parameter to make the intercept only regression model fits better.

At each level of the tree, the whole data is splitting into disjoint subsets. If one wants to make a prediction for a person given a binary tree, what the practitioner needs

to do is placing the person at the top of the tree, and asking him the question about the first input variable, to determine if the person would go left or right, repeating this step until the person reaches the terminal node, then the average value of the node would be the predicted value for this person. In other words, the person find its position in the small sets of the tree and use averaged peers' response as the prediction for this person. It is in a similar manner of survey with only two options for each question which eventually leads to the answer.

Regression tree works with both continuous and discrete input variables. Figure 3.1 is an example of regression trees. Each node are the produced children as the subset of data, illustrated as a rectangle or circle. Nodes without further splitting are the terminal nodes as a rectangle. We also notice that the same splitting variable may appear multiple times in the tree with different cutoff values. The binary splitting feature can be extended to more children for each parent node but it would be more parameterized and not easy to draw as a 2D graph.

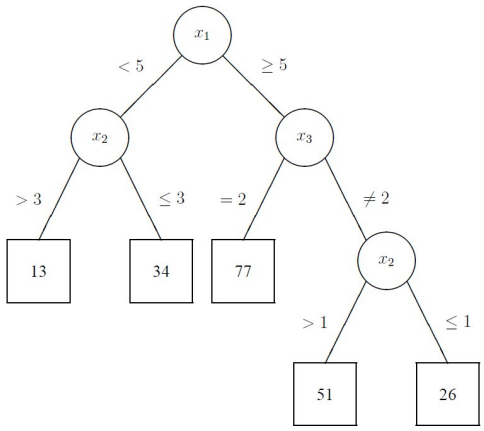

Figure 3.1: A Simple Example of Regression Tree

3.2.2 Algorithm of Classical Recursive Regression Trees

The regression tree is a greedy method. The term "Greedy" is commonly seen in optimization theory. When investigators are interested in finding the minimum values, it is often that the explicit solution is not available as a closed form. Therefore many iterative methods have been proposed to find the extreme values such as the popular

class of Newton's methods. For iterative method, the choice of step length is critical, which will affect the efficiency and some inappropriate choice would even miss the true solution. Greedy type of algorithms tend to achieve the best effect in reducing the objective function' value at each step.

Similarly, for regression tree method, the objective is to minimized the sum of squared errors between the observed responses and predicted nodes' mean. The value of this objective only depends on two factors: the choice of splitting variable and the position of jump point. The regression tree selects the combination of variable and splitting point, which the current minimal sum of squared errors are achieved by the pair. After current optimal splitting is formed, the next split on each child would be following the same rule. Therefore at each step, the method would find the best split and fits the definition of greedy. Such greedy methods do not guarantee to find the optimal solutions but if it does then it has superior computation complexity.

Let us formulate the method. Suppose the data consists of P input variables and N observations, the response variable is Y, features are X_i, and samples are (y_i, x_i), $i = 1, 2, ..., P$, where y_i is a number and x_i is a $1 * p$ vector. Without loss of generality, Y and all X_i's are continuous. At the first step, the root node contains all samples, assume X_{j_0}, some $j_0 \in \{1, 2, ..., P\}$, is the splitting variable, the partitioned regions for this variable are $R_1(t) = \{X_{j_0} < t\}$ and $R_2(t) = \{X_{j_0} \geq t\}$, some $t \in \mathbf{R}$, and the predictive function takes values c_1 and c_2 on left and right child respectively. The prediction function can be represented as $f(x) = c_1 I_{\{x_{j_0} < t\}} + c_2 I_{\{x_{j_0} \geq t\}}$. A typical choice of loss function is the squared loss $L(c_1, c_2, j, t) = \sum_{\{i: x_{i,j} < t\}} (y_i - c_1)^2 + \sum_{\{i: x_{i,j} \geq t\}} (y_i - c_2)^2$. The indices of samples are cut into two disjoint sets, therefore minimizing the sum is equivalent to minimizing each summation. For fixed splitting variable and cutoff point, i.e. given $j = j_0$, $t = t_0$,

$$\min_{c_1, c_2} L(c_1, c_2 | j_0, t_0) \Leftrightarrow \min_{c_1} \{ \sum_{\{i: x_{i,j_0} < t_0 | j_0, t_0\}} (y_i - c_1)^2 | j_0, t_0 \} + \min_{c_2} \{ \sum_{\{i: x_{i,j_0,t_0} \geq t | j_0, t_0\}} (y_i - c_2)^2 \} \tag{3.1}$$

Solving for the c_1, c_2, we have $\hat{c}_1 = \frac{\sum_{i: x_{i,j_0} < t_0} y_i}{\|R_1(t_0)\|}$ and $\hat{c}_2 = \frac{\sum_{i: x_{i,j_0} \geq t_0} y_i}{\|R_2(t_0)\|}$, where $\| \cdot \|$ provides the cardinality of a set. In other words, the average of observed responses

in each region. But this minimized loss still depends on the choice of j_0 and t_0, so our optimization objective is $\min_{j,t,c_1,c_2} L(c_1, c_2, j, t; Y, X) \Leftrightarrow \min_{j,t}\{\min_{c_1,c_2} L(c_1, c_2; Y, X|j, t)\}$. The methods of finding splitting variable and split point had raised a lot of attention: [Chou, 1991], [Murthy et al., 1994] and [Loh and Shih, 1997]. Efficient algorithms have been proposed to find the splitting variable and point. After the first split is finished, treat each child as the root node in previous step and preform the partition independently. Independence means that the partition procedure on each child is free to select any input variable and any cutoff point; they would not affect the choice of the other. [Quinlan and Cameron-Jones, 1995] pointed out there's a bias in the variable selection step of decision trees. A larger tree means the model contains more refined regions and corresponding parameters, therefore it tends to overfit the data. It would provide low predictive accuracy for a new observation. A way to restrict the size of a tree is to permit further splits only if the benefit exceeds certain threshold. It is similar to the adjusted R^2. The adjusted version of R^2 takes into account the number of input variables. One difference in the tree model is for each split, it is purely independent of other nodes, and we know the split would introduce fixed number of parameters: the boundary point of two subregions and the mean responses within each subregion. Therefore there is not need to consider the number new parameters since it is a constant. Therefore the perspective restriction would be a certain amount of increased goodness of fit. In the regression tree model, practitioners are free to choose a small real number as the threshold. As long as the splitting will improved the squared error loss by this threshold, the split is allowed to perform. It is clear that this criterion eliminate the potential paths which does not gain much in an intermediate partition but would improve greatly in further split, i.e. local non-optimal does not necessarily yield global non-optimal. Combining the idea of cross validation and tuning complexity parameters, another solution has been proposed to control the size of trees. For this class of tree models, the tuning parameter could be a multiplier to the maximum depth of the tree, the radius of the tree, the total number of nodes and more. [Hastie et al., 2009] suggested penalizing the total number of terminal nodes. Notice that another termination criterion is the minimal sample size of a parent node to be slitted. Using a small integer

such as 5 would produce a deep and wide tree. Based on the on this large tree, a cost complexity pruning was proposed to produce a less fitted tree [Breiman et al., 1984] and [Ripley, 1996]. Suppose T_0 stands for the large tree and T is any subtree of it. All terminal nodes are corresponding to a series of regions, denoted as R_i, $i = 1, 2, ..., K$, where K is equal to the total number of terminal nodes of tree T. If the cardinality of a tree is defined in this way, then $\parallel T \parallel = K$. Since the prediction of the outcome would be based on the region that a sample falls into, therefore only the squared errors in terminal nodes are of our concern. Averaging on the number of samples, the squared error of region i is

$$L_i(T) = \frac{1}{\parallel R_i \parallel} \sum_{i:x_i \in R_i} (y_i - \hat{c}_i)^2 \text{ , where } \hat{c}_i = \frac{\sum\limits_{i:x_i \in R_i} y_i}{\parallel R_i(t_0) \parallel}. \tag{3.2}$$

Define the cost complexity criterion as

$$C_\alpha(T) = \sum_{i=1}^{\|T\|} L_i(T) + \alpha \cdot \parallel T \parallel \tag{3.3}$$

5-folder or 10-folder cross validation [Kohavi et al., 1995] is a typical way to find the appropriate α. Suppose the best α is α_0, we would like to find the subtree T_{α_0} and minimize $C_{\alpha_0}(T)$. It can be proved that for each α, there is a unique solution $\hat{T}_\alpha$ minimizing $C_\alpha(T)$. The way of finding $\hat{T}$ for given α is a discrete procedure since producing a tree would keep increasing the number of nodes by constant units when expending a terminal to internal node. When $\alpha = 0$, $T_{\alpha_0} = T_0$. Increasing α would make a smaller tree and fit worse. Since we already obtain the full large tree, what we can do is to collapse the trees. At each step, find the internal node where the partition on it gives the smallest gain in $\sum L_i(T)$. When a node is collapsed, the tree is becoming smaller and α will increase. Keep this collapse procedure until the tree becomes a single node and α achieves its maximum possible value α_{max}. The whole path corresponds to a monotone step function of α between 0 and α_{max}. For any given α_0, find its position in this path and its subtree structure, which would be the solution.

There are several popular implementations of the classification and regress trees model. One influential free package is rpart. rpart is the abbreviation of Recursive

PARTitioning, which implements the idea from [Breiman et al., 1984]. Our modification is based on this package, therefore we share essentially the same parameter sets. I am stating the framework of constructing a regression tree and would adopt some terminology from rpart package. There are several options need further explanation before presenting the model building steps. In the rpart implementation, a few parameters are important: minsplit, minbucket, cp, usesurrogate and xval. Among them, minsplit, minbucket and cp are related to the termination condition of the node splitting procedures; usesurrogate functions only if there is any missing value in the data; and xval is the parameter associated with cross validation in the step of reducing the tree size, known as the pruning step.

minsplit is an argument taking integer values and functions as the threshold of the size of potential parent nodes. If neither minsplit nor minbucket is specified, then minspit takes value 20. Therefore if a node contains no less than 20 observations, then the program would attempt to split it, otherwise it becomes a terminal node. minbucket is an argument taking integers and serves as the threshold of the size of potential children nodes. With this option, if a feature and its best splitting point would result in a very small children nodes, then it would be excluded from the candidate features to compare the loss with other competitors. If minbucket is not assigned with a integer, then it takes one-third of minsplit. We can see that if a node is stopped due to minsplit or minbucket constraint, the computation is different. The program checks the minsplit criterion first, if it passed, then all possible variables would be tried, their optimal region boundaries would be calculated. This is an extra calculation. And then the minbucket condition would be checked. The default sets minbucket to be one third of minsplit, which implies that the author preferred to generate relatively balanced trees.

cp is the abbreviation of complexity parameter. For regression trees, more refined partitions lead to less squared error. Adding a split would fit the data better. If only a tiny gain is achieved by separating a parent node, then later when we collapse the tree to reduce variance, this severation would be likely reversed. Therefore we pick a tiny threshold for this value. Please be aware that there are two parameters with the same name of complexity parameter. The penalty coefficient is also name the same way.

To distinguish from it, we call the cp introduced here as lcp, meaning the complexity parameter for the loss improvement. And the penalty cp is denoted as pcp.

Surrogates options are for determining the way of dealing with missing values. It takes value 0, 1 or 2. Essentially this approach utilizes secondary variable if value of primary variable is missing, instead of eliminating such samples. The idea of surrogate was proposed [Breiman et al., 1984]. The motivation was to provide a sequence of feature variables to substitute the primary splitting variable when necessary. It happens in reality that the training data and testing data do not have exactly the same structure. For instance, a new observation in the testing data may have a missing value in certain features, and among these features, one was selected as the primary splitter in the training set. So when an alternative feature mimics the splitting feature, then we are still able to place this new observation in the tree and find its final position. Thus the new observation's response is predictable.

It is possible that there are missing values for the surrogate. So in principle, there would be a list of surrogates, ranked by their partition similarity to the primary splitter. When the primary splitting variable is not observed, the first surrogate would be used to send the observation to the left or right. If the first is missing, the second would be used. If all are missing, then the "blind rule" will be used: sending the observation to the majority of classified child node. Although users are free to select a number as long as it is no more than the total number of features but the program also gives the option to force the surrogates better than blind rule. There are two main options for controlling surrogates: usesurrogate and maxsurrogate.

- maxsurrogate: an integer which gives the maximum number of surrogates to compute after the main splitting variable and cutoff point are determined. Its default value is five. The intention of surrogates is not to get close performance of the primary splitter but get close to the partition result. If the value is set to zero, then the usesurrogate is also set to zero naturally.

- usesurrogate: takes value 0, 1 or 2. If it is 0, then there is no surrogate computed, thus if the primary splitting variable is missing for a observation, it would not

be placed in any of its child node. If the value is 1, then there is not blind rule to consider; only use all surrogate variables in order, if all are missing then the observation would not be sent down to the branches further. If 2 is assigned, the surrogates list and blind rule would be combined. The program will choose surrogates which perform better than the blind rule. [Breiman et al., 1984] recommends using 2 for this option.

In the second stage of tree construction, we need to trim some branches to trade off between bias and variance. The main tool in this step is cross-validations. This is a widely used method for estimating prediction error. The estimated error is used to choose the value of tuning parameter. The motivation of cross validation is to produce a training data set and a testing set from limited samples. It partition the data into training and testing part and does the same regular validation procedure, but it also reuses the data and switch the role of training and testing parts, which is the innovation part of this method.

In general, it is K-folder cross validation, where K is an integer no more than N - the total number of observations. The dataset would be partitioned into K parts evenly, and the whole method requires iterative computation on each folder. The steps of cross validation: for $k = 1, 2, ..., K$, remark the $k-th$ folder as the validation set, and the rest $K-1$ folders are training set. Fit the model to current training part, calculate the prediction error on the $k-th$ folder with fitted model, and then repeat the procedure for all k's and average over all prediction errors. Suppose $I_K(\cdot)$ is the function assigning observations to folders' indices $1, 2, ..., K$, $\hat{f}_{-I_K(I_K^{-1}(i))}(\cdot)$ is the fitted model when the folder $I_K(i)$ is not included in the training set, where $I_K(I_K^{-1}(i))$ are the sample indices which are in the same folder as observation i and the minus sign means excluding. The predicted value on object i is $\hat{f}_{-I_K(I_K^{-1}(i))}(x_i)$. If we use mean squared error loss, then the prediction error by cross validation is

$$CV(\hat{f}, K, \alpha) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{f}_{-I_K(I_K^{-1}(i))}(x_i))^2 \tag{3.4}$$

In the equation, α is the potential tuning parameter for the model, but it is cumbersome to add it to the subscript of f, so it is ignored on the right hand side of the equation. For

fixed K, the cross-validated prediction error depends on the valued of α, so in principle the minimizer $\hat{\alpha}$ is the best performer if CV represents the true prediction error.

Let us review the class of errors which affects the model assessment and selection. Suppose $\mathcal{T}$ is the training set and its trained model is $\hat{(}f)$, the test error, also called generalization error is

$$Err_{\mathcal{T}} = E\big(L(Y, \hat{(}f)(X))|\mathcal{T}\big) \tag{3.5}$$

where pairs of (X, Y) are random samples from the population the and training set is considered to be fixed quantities. But in reality, the training set is also randomly drawn from the population, therefore to eliminate such randomness, the expected test error is defined as

$$Err = E\big(L(Y, \hat{(}f)(X))\big) = E\big(Err_{\mathcal{T}}\big) \tag{3.6}$$

Hastie et. al. pointed out that it appeared that when K was a small number such as 5 or 10, the cross validated prediction error estimated expected test error while when K was large, for instance, taking the same value as the sample size, it appeared to estimate the conditional error; but the simulation study showed that it only approximates the expected error well. It is obvious that when $K = N$, all training sets share a high similarity, therefore the estimator is approximately unbiased but has high variance. If the K is limited down to a small number, it tends to overestimate the expected error but reduces the variance. It is recommended to choose K to be 5 or 10 [Kohavi et al., 1995]. It is also recommended to apply one-standard error rule to cross validation. It requires the calculation of standard error for different tuning parameters, or for model with different complexities since slightly different turning parameter may lead to the same model. Suppose the cross validated prediction error is minimized at α_0 with m_0 variables in the model, say $L_{\mathcal{T}}(m_0)$, and the model with m_0 variables has slightly worse error loss. But if the error $L_{\mathcal{T}}(m_0 - 1)$ falls into the $L_{\mathcal{T}}(m_0) \pm sd(L_{\mathcal{T}}(m_0))$, then the model with $m_0 - 1$ features are preferred. And if there is a even more compact model satisfying this criterion, then that would be preferred than $m_0 - 1$. So it can be formulated as looking for

$$\text{argmin}_k L_{\mathcal{T}}(m_0 - k) : L_{\mathcal{T}}(m_0 - k) \geq L_{\mathcal{T}}(m_0) - \text{s.d.}(L_{\mathcal{T}}(m_0)), k \in \mathbb{N} \tag{3.7}$$

The cross validation method was used in the tree model's pruning step. Equation (3.3) illustrates the loss function with a penalty term to the tree complexity. In order to apply cross validation, we simply need to replace the $\hat{c}$ in equation (3.2) as the average value of assigned samples given the tree is built on $K-1$ folders' data. One related question is how to find the corresponding subtree for given α.

Let us review the procedures of modeling a regression tree first. The framework of building a regression tree consists of two stages:

1. Build a fully partitioned tree
    (a) Specify the variables as the candidate of split variables
    (b) Determine the minimal bucket, minimal split, loss improvement complexity parameter(lcp), max surrogates, max tree depth and the way of using surrogates
    (c) At each tree level, check if current depth exceeds the max depth requirement. If not, then leave all nodes as terminals. If yes, perform the following tasks:
        i. Check if the minimal split constraint is satisfied by current node. If yes, go to next step, otherwise leave current node as a leaf.
        ii. Attempt every candidate feature, and its associated binary partition point. All splits do not achieve improved prediction error (less than lcp) would not be considered. All splits do not produce next generation with more than minimal bucket samples would not be used even it produces the best partition performance.
        iii. Compare the improvement of sum of squared errors among all variables. Find the best as the splitter.
        iv. Set level to be level+1 and go to step (c).
2. Generate a reduced tree
    (a) Specify the number of folders in cross validation, which is xval in package *rpart*.

(b) Obtain the estimated expectation and standard deviation of prediction error for α's associated with different tree sizes.

(c) Use weakest link pruning to obtain the path of subtrees for α's associated with different tree sizes.

(d) Determine the optimal α value as α_0 and print its associated collapsed subtree.

3.3 Modified Recursive Regression Trees

I would like to introduce the way we constructed our tree model for this specific post-GWAS study. There are several aspects to consider instead of applying the traditional version directly. As mentioned in Chapter 2, our data sets consists of clinical part and genetic part. Therefore if we combine them and use all as the candidate for potential splitters, the underlying assumption is that these two datasets have the same importance and they functions the same in predicting the blood pressure sensitivity since they would be competing each other. That would be inappropriate for several reasons: biologically we do not know which factor plays a more important role. It is not a fair comparison. Even this assumption is true, by introducing more polymorphisms, the chance of selecting SNPs over clinical measurements would be increased and vice versa. It is not a fair comparison even the pool size of two types of variables would not be essential. Assume age is as important as a polymorphism in reality, say rs5186, but all patients are limited to be no younger than 60 years. Conditioning on this criterion, age's predictability might be changed and it requires a way to make adjustment. A potential question would how to measure the importance, and this has to be done for each variable or a bundle together.

Another issue is if the tree is generated mixed with polymorphisms and biological traits, then there is not way to link multiple SNPs since the splitting procedure is independent to its children and polymorphisms competes with each other solely. This motivates us to assign roles to the variables sets based on their function, the data nature and our research purpose.

There is another issue preventing us from performing regression directly is that the original method is designed for supervised learning, meaning that the outcome values are available. The observed response is critical in but not limited to these steps:

1. In the selection of splitter, observed responses serves to determine the best feature and the split point
2. After the partition, the mean observed responses works as the approximation to the predicted value in a node
3. Without the response variable, cross validation is impossible and so is the pruning step.

Therefore it plays a important role in both the steps of constructing the complete tree and pruning the tree to reduce variance. The way we proposed would be introducing a more complicated regression model within a node. Therefore we have more measurements instead of the average only. A more complicated model brings more descriptive quantities to the model fitting and may be utilized in the tree construction.

3.3.1 Choice of Splitters

Given these considerations, our first idea is to assign different functionalities to the sets of features. We would like to use only one part of the TONE data as the candidate pool for the node partition. The advantage is there is less concern about the importance weights of genetic variables and clinical features. We choose to use genetic variables for this purpose with following reasons:

1. There are less concern about the data modes if using genetic pool.
2. Each tree path corresponds to a genotype.
3. Clinic data can be used within each node.
4. Computation complexity is better than the traditional tree.

The SNP data has a better structure: each variable is categorical with 3 levels; while clinic data consists of continuous features such as age, ordinal variables such as the

workout rates and nominal variables such as the gender. So if using clinic data, we were still facing the issue of determine the splitting importance and how to make adjustment. For the well-structured SNP data, all inputs are of the same type, and the observed values have the same biological meaning. If a tree is split by different polymorphisms, then any path from root to leaf corresponds to a particular genotype and this correspondence is unique. The genotype is defined to be an realized genetic type determined by a single or multiple polymorphisms. A genotype identifies a subset of participants sharing the same characteristic in their DNAs of the area that we are interested in.

3.3.2 Split Variable Selection

The way of selecting the variables to build trees in our version is different from the traditional regression tree. We need to distinguish the responses in the clinical data and our objective. In the clinical trial, the blood pressure before and after the intervention were recorded. They and their difference could become the response. But the objective of our post GWAS analysis was to look into the ratio of change in blood pressure and change in BMI or sodium. In other words, we would like to obtain the dropped amount of blood pressure given unit improvement in BMI or sodium, controlling for other variables. This quantity is called sensitivity in our analysis and was not observed directly. It is not feasible to use sensitivity as the supervisor. We can not compute the difference between estimated (predicted) sensitivity and the truth. Alternatively, we choose the improvement of goodness of fit to the blood pressure change as our criterion. It turns part of statistical analysis into a supervised approach. The underlying reason that we take this approach is we would like the model containing the sensitivity represents the data well. A model replicated the data well gives us the confidence to adopt the associated sensitivity.

One related question is which quantity to use as the indicator of goodness of fit. There are traditional R^2 and adjusted R^2. We need to estimate the sensitivity within every node under the same model. We may apply a nested model or a saturated one with more inputs, but the input features would be determined before the tree construction,

and would not be altered in any intermediate step. It is not necessary to control for the number of inputs.

The way we looked at the improvement is straight forward: fit the model for observations in the parent node and calculate the sum of squared errors, i.e. the squared sum of differences between observed blood pressure drop and predicted drop given by the model, denoted as RSS_{parent}; for each partition, fit the model and calculate the RSS, denoted as RSS_{left} and RSS_{right}; calculate $\Delta RSS \triangleq RSS_{right} + RSS_{left} - RSS_{parent}$. The splitter for current node would be the one improves this number most. We do not need to adjust for the number of inputs or observation, since all potential candidates are competing with each other under the same model frame. Each polymorphism would be three variants: homozygous, heterozygous and wild type. These terms are used to describe the genotype at a single locus on the DNA sequence. Homozygous refers to the case that a pair of identical recessive alleles located at the locus, Heterozygous means the pair consists of two different alleles and Wildtype means both dominant alleles are presented. We did not make any assumption that which one or two of them would function together. And given the binary partition, we consider the cases where observations are placed together if they have the same genotype.

The selection procedure is listed below:

For $i = 1, 2, ..., 21$,

1. Choose the $i-th$ polymorphism SNP_i, attempt three possible partitions:
   (a) Homozygous vs the other
   (b) Heterozygous vs the other
   (c) Wildtype vs the other
2. Each partition creates two subgroups of current node. Check if it satisfies the minimal split and minimal bucket requirement, if passed then go to next step.
3. Calculate the improvement in the goodness of fit $\Delta RSS(SNP_i, Split_j)$.
4. Choose the polymorphism and the partition which minimize all ΔRSS's.

3.3.3 Within Node Regression

In the procedures we listed above, each node fits a model independently. But there is another choice: fit the model with a subgroup indicator. More precisely, suppose the indictor function of left child node is

$$I(x) = \begin{cases} 1, & \text{if } x \text{ is in left child node.} \\ 0, & \text{otherwise.} \end{cases} \tag{3.8}$$

If $I(x)$ is included in the model and has a coefficient β_1, and suppose the coefficient for BMI change is β_2, then β_1 is the sensitivity for right child node and $\beta_1 + \beta_2$ is the sensitivity for left child node. It takes little effort to show that if the response and inputs are standardized and centered, the estimate coefficients are equivalent to the separate model without the indicator function. What we present in this thesis are the results without using indicator function.

3.3.4 Control Parameters

Depends on the size of different treatment groups, the optional parameters controlling the varies on groups. In Section 3.1, we introduced the components of participants in each group.

1. Over weight participants, received both sodium reduction and weight loss treatments
2. Over weight participants, only received weight loss treatment
3. Over weight participants, only received sodium reduction treatment
4. Over weight participants, control group
5. Normal weight participants, only received sodium reduction treatment
6. Normal weight participants, control group

We take control of the size of parent nodes and children nodes as below: The number of folders is set to be five for cross validation to determine the tree complexity penalty parameter. We select a small *lcp* in order to build a large initial tree, which is 0.001.

Group	Minimal Split	Minimal Bucket
1,2,3,4	40	25
5,6	60	30

Table 3.1: Controlling Thresholds of Nodes' Minimal Sizes

3.3.5 Test Procedures

After building a complete tree, we perform the same pruning surgery as the classical regression tree. Pruning the tree provides a compact tree to us with much less nodes and depth. The next objective for us is to figure out the relative sensitivity from the output of the tree. In other words, reducing weight or sodium intake would generally reduce the blood pressure. But we would like to know people with which genotype benefit most from it and whose are only on par with the overall improvement. We need to rank all terminal and intermediate nodes based on the interest.

Suppose the sensitivity of any node is T_i, of the root node is T. We are interested the difference between them. But all sensitivities are not observable, so we need to investigate the difference between estimated sensitivities. Since those statistics are data-dependent, we need to consider the quality of the estimators. In other words, we have to take into account the standard deviation of the estimated slope.Without loss of generality, we consider the weight sensitivity problem. Assume β_0 is the coefficient of weight change in the model as regressing systolic blood pressure change to basic covariates for root node, and β_1 is the corresponding coefficient for any other node. We are interested in three aspects:

- The sign of $\beta_1 - \beta_0$
- The value of $|\beta_1 - \beta_0|$
- The standard deviation of $\beta_1 - \beta_0$

If the difference is negative, then we may conclude that the people do not receive as much benefit as the majority from the intervention of the genotype identified by this particular node. Recall that we have multiple combinations of interventions for the participants. The tree model would be only applied to non-placebo group due to our interest. There

were three interventions introduced to all treatment groups. Some only got sodium reduction, some only lost weight by training, and the rest got the combined treatments. If the tree was built for patients in the salt intake control program and got significantly negative difference for the node, it might be concluded that for this particular genotyped people, eating less salt make have negative impact, which contradicts the common sense. But at least for those who did not get significant positive increment, taking control of body's sodium level would not lower the blood pressure. One potential analysis is to look at the same genotype receiving other treatments. If none of them works well, then people with this particular genotype would be recommended to take medication. The sign is not very critical if the standardized value is close to zero. But it makes more sense when the standardized value is far from the origin. It needs attention from biologists especially when the absolute value is large and the sign is negative.

After obtaining the estimated values of β_1 and β_0 as $\hat{\beta}_1$ and $\hat{\beta}_0$, we need to consider how to calculate the variance of the estimated sensitivities' difference. About the standard deviation of estimated $\beta_1 - \beta_0$, it is obvious that we can not assume those two $\hat{\beta}$'s are independent since the design matrix for two regression procedures are nested. It is easy to observe from the tree structure that an ancestor always contains the observations of any node. It implies that the sensitivity of an ancestor is estimated with a larger portion of data, while descendant's sensitivity is estimated with part of data in ancestor. In terms of bias, it is not concerning us but we have to use caution for estimated variances. It can be proved that the variance of difference in estimated sensitivities is equal to the difference in the variances of estimated sensitivities. It is formulated and proved below.

Propersition 3.1. *Suppose the estimated slop for any ancestor root is T_0, and its descent's estimated sensitivity is T_1, both are from the least square estimation, and the assumptions of linear regression hold, then*

$$Var(T_1 - T_0) = Var(T_1) - Var(T_0). \tag{3.9}$$

Proof. Suppose for the ancestor node, the design matrix is X, and the response is Y; for the descent node, the design matrix is X_1, and corresponding response vector is Y_1.

Assume X has N rows and P columns, and X_1 has M rows. Without loss of generality, we can rearrange the order of observations, such that the first M rows of X is the same as X_1. Then we can rewrite the data of ancestor as $Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix}$, $X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$.
The linear model is $Y \sim X\beta + \epsilon$. Its estimated parameter vector is $\hat{\beta}_0 = (X'X)^{-1}X'Y$.
With partial data, the estimated parameter is $\hat{\beta}_1 = (X_1'X_1)^{-1}X_1'Y_1$.
Define $\tilde{X} = \binom{X_1}{0}$ by replacing the X_2 in X with zeros. Obviously,

$$X_1'X_1 = \tilde{X}'\tilde{X},\ X_1'Y_1 = \tilde{X}'Y. \tag{3.10}$$

Thus $\hat{\beta}_1 = (\tilde{X}'\tilde{X})^{-1}\tilde{X}'Y$.
Consider the j-th element of $\hat{\beta}_0$ and $\hat{\beta}_1$: $T_1 \triangleq c'\hat{\beta}_1$, $T_0 \triangleq c'\hat{\beta}_0$, where c is all zeros but the i-th position, corresponding to the order of BMI change or sodium change in the feature space. Define $T = T_1 - T_0$, then we have

$$\begin{aligned}
&Var(T) = Var(c'\hat{\beta}_1 - c'\hat{\beta}_0) \\
=&c'Var(\hat{\beta}_1 - \hat{\beta}_0)c \\
=&c'Var\big[(\tilde{X}'\tilde{X})^{-1}\tilde{X}'Y - (X'X)^{-1}X'Y\big] \\
=&c'\big[(X'X)^{-1}X' - (\tilde{X}'\tilde{X})^{-1}\tilde{X}'\big]Var(Y)\big[(X'X)^{-1}X' - (\tilde{X}'\tilde{X})^{-1}\tilde{X}'\big]'c \\
=&\sigma^2 c'\big[(X'X)^{-1}X' - (\tilde{X}'\tilde{X})^{-1}\tilde{X}'\big]\big[(X'X)^{-1}X' - (\tilde{X}'\tilde{X})^{-1}\tilde{X}'\big]'c \\
=&\sigma^2 c'\big[(X'X)^{-1} + (\tilde{X}'\tilde{X})^{-1} - (\tilde{X}'\tilde{X})^{-1}\tilde{X}'X(X'X) - (X'X)^{-1}X'\tilde{X}(\tilde{X}'\tilde{X})^{-1}\big]c \\
&\text{Notice that } \tilde{X}'\tilde{X} = X_1'X_1 = \tilde{X}'X, \text{ we have} \\
=&\sigma^2 c'\big[(X'X)^{-1} + (\tilde{X}'\tilde{X})^{-1} - (\tilde{X}'\tilde{X})^{-1}\tilde{X}'\tilde{X}(X'X) - (X'X)^{-1}\tilde{X}'\tilde{X}(\tilde{X}'\tilde{X})^{-1}\big]c \\
=&\sigma^2 c'\big[(X'X)^{-1} + (\tilde{X}'\tilde{X})^{-1} - (X'X)^{-1} - (X'X)^{-1}\big] \\
=&\sigma^2 c'\big[(\tilde{X}'\tilde{X})^{-1} - (X'X)^{-1}\big] \\
=&\sigma^2(\tilde{X}'\tilde{X})^{-1}_{ii} - \sigma^2(X'X)^{-1}_{ii} \\
=&Var(\beta_{1,i}) - Var(\beta_{0,i})
\end{aligned}$$

□

The key idea in the proof is to express the partial estimator as the linear combination of the complete response vector. We achieve it by creating a dummy input matrix and utilizing its property.

Researcher devoted to investigate the issues about removing or adding features to the current input matrix. A complete and practical treatment has been provided from the applied point of view [Montgomery et al., 2012].

But the problem of using partial data with regression model has not received much attention. In today's big data environment, popular tools usually distribute a huge dataset by sending different columns to machines (as data nodes). It requires the master node(who coordinates all computation tasks) know the size of the whole data and then determine how many columns to send to each data node, therefore a complete copy of data has to present somewhere in the connected nodes network. Hadoop and Spark handle data storage in the stated way above. But in reality, data are in the form of streams. There would be new samples collected from time to time. And the it is desired to update the analysis by using the increment portion instead of to redo the computation with the new portion and existing part. Our Proposition 3.1 may provide some insight to the updating procedure if such by-row framework is possible. Even beyond the big data developing environment, this property may find its position in classical regression. The result is short, precise and has its application. The most important is no one mentioned it under fundamental regression context. We noticed this fact when we do the comparison between sensitivities of root node and any internal and terminal nodes. Assuming $\hat{\beta}_0$ and $\hat{\beta}_1$ is against intuition but we could not find any statement about the variance of this quantity. I think one potential reason is that in most cases researcher collected and cleaned the data as a preprocessing step, so the data for further analysis would remain the same.

Propersition 3.2. *The statistics T_0 and T_1 are both unbiased for β.*

This fact is trivial, so we do not provide any proof to it. As a consequence, the expectation of the difference is zero. After figuring out the basic first and second order moments of our test statistic, when we perform the test, there is one more problem to mention. The variance of error term is not observable, so we need to estimated it by mean squared error of the regression model. But since we have applied this model twice with two non-identical data sets, we got two estimates of the variance of normal error

term. This estimator would be more reliable with more observations, so we adopted the one reported from the ancestor node. Therefore the estimated standard deviation of β's has to be corrected for the descendant node. We use the same notation as the theorems above. Estimated variance of T is

$$\widehat{Var(T)} = \hat{\sigma^2}(X_1'X_1)^{-1}_{ii} - \hat{\sigma^2}(X'X)^{-1}_{ii} = \frac{\hat{\sigma^2}}{\hat{\sigma_1^2}}\hat{\sigma_1^2}(X_1'X_1)^{-1}_{ii} - \hat{\sigma^2}(X'X)^{-1}_{ii},$$

where $\hat{\sigma_1^2}(X_1'X_1)^{-1}_{ii}$ is the estimated variance of T_1 using partial data X_1 and $\hat{\sigma^2}(X'X)^{-1}_{ii}$ is the estimated variance of T using complete data X. Each time fitting a model would return a set of estimates: the estimated coefficients and their standard deviation, mean squared error of the model, coefficient of determination, all associated p-values and mroe. Among them, what we need are the MSE, estimated coefficient and its standard deviation of the BMI change or sodium change. For each node, we need to return the sensitivity, partially estimated variance, and MSE of current node with partial data. With these statistics, we are able to make adjustment to the variance estimation of the difference in the slop. There was an implementation issue when we tried to make the adjustment to our tree model. The *rpart* package only supports us to return a single value for each node. What we did is to look at the ranges of three quantities and use a shorten scientific expression to bind them and form a value. The principle is that we use about 4 digits for each number. We extracted and sliced the combined digits with Perl to obtain these 3 estimates. If there is only a few regression models to fit, then collecting results procedure is easy. But considering that we are building a complete large tree, even the splitter and partition point had been determined, there are still the same number of models as the nodes.

3.3.6 Comparison with Traditional Regression Trees

Our trees model is associated by the traditional regression tree, therefore they share common components. First of all, both are in the binary tree structure, which gives the model more interpretability and makes the calculation relatively feasible. Constructing a tree may be from bottom to top or top to bottom. The disadvantage of bottom-up

is that as long as a node is formed, it is difficult to eliminate observations in followed constructions or reassign an observation to another node. Therefore the structure is very sensitivity to the first few levels of combination. For top-down approach, it is also not possible to adjust the node members but is less affected by this issue,since we may always eliminate observations from current node by a further partition. The traditional regression tree does not use the bottom-up approach either. We adopted the same strategy and start with a single node.

In general, a parent node may have various number of children. The way of determining the splitting region and choice of split variable are similar; both seeks to fit the data well. The termination conditions are also similar. The way of reducing the size of tree (pruning) is the same. After a tree is obtained, the interpretation of nodes are the same.

In contrast to the traditional regression tree, within each node, we do not model the response as a constant. Instead a regression model is implemented and used to estimate the sensitivity. The sensitivity is defined to be the change in the dependent variable given unit change in BMI or Sodium measurements. The dependent variable is our primary interest - systolic blood pressure. This quantity is estimated as the slope in the regression model. The modeled response helps us in the step of tree construction although its change is our interest. Another difference is that in our model, the tree uses two input data sets and their functionalities are different. The genetic part of data provides the set of potential splitters, while the non-genetic part is used to determine the coefficients and influence the tree construction. The roles of response variable are not identical. In traditional regression tree, the estimated response variable are of importance while in our modification, it is an intermediate quantity which helps us to determine the goodness of fit, but not our ultimate interest. Our modified version uses the same supervised idea to prune the tree, but there is no way to perform training on the slops since they are not observable. Below are summarized comparison.

Shared

1. Hierarchical and binary structure

2. Splitting Options and criterion
3. Termination condition
4. Pruning method
5. Interpretation of nodes

Differences

1. Separated groups of input variables
2. functionalities of inputs int the tree construction
3. Role of response variables
4. Post process to the tree building

3.3.7 Results

Two results are presented here. We analyzed the weight and sodium program and restricted observations to patients who only received a single treatment. The tree result of Figure 3.2 is from 110 people who received only weight loss treatment. In each node, we showed the number of patients of some genotype, the sensitivity and the split information. We can see that the first split genetic feature is rs4646994 and the partition point separated the group into wildtype and others. The slopes in the graph are estimated parameters for weight change. We can see that for all patients in the group, the overall change is a drop of 2.52 in systolic blood pressure given unit drop of body mass index. The commonly used levels of BMI are as below:

- Underweight: BMI is less than 18.5
- Normal weight: BMI is 18.5 to 24.9
- Overweight: BMI is 25 to 29.9
- Obese: BMI is 30 or more

The commonly defined high, normal and low SBPs are as below:

- Hypotension: SBP is less than 90

- Normal: SBP is 90 to 119
- Prehypertension: SBP is 120 to 139
- Hypertension: SBP is 140 or more

The top three significantly positive nodes are marked with orange. They are corresponding to polymorphisms: rs1800629, rs19822073, rs1800896. The combination paths are shown in the figure. Based on the model, we are able to make the prediction that if a person's genotype is the same the path indicated, he/she will benefit significantly if he reduces the weight and does not even need medication. But reversely, if a person did not gain or even worse than average, non-pharmacological intervention does not help him/her. Medication is suggested for such people. The second results are from subgroup receiving salt intake control program. Similar to the previous tree result: for each cell, we included information about size, sensitivity and the partition point. The way of interpreting red nodes is in the same manner. We compared the beneficial drop in blood pressure given unit change of sodium measurement versus overall level and picked the top nodes with associated genotypes. Patients belonging to these nodes benefited by eating less salt. Other patients in the other direction may need to take medication to help them to reduce the blood pressure. It is easy to notice that there are repeated snips in Figure 3.3. For instance, rs 19822073 appears twice in the left tree and it is within the path of two significant nodes. It is possible that the node with 71 patients is also positive enough that its children may also show a strong sensitivity compared to the overall level. We also notice that the overall sensitivity is negative and the significance is determined by both the numerator (difference in the slopes) and the denominator (difference in standard deviations).

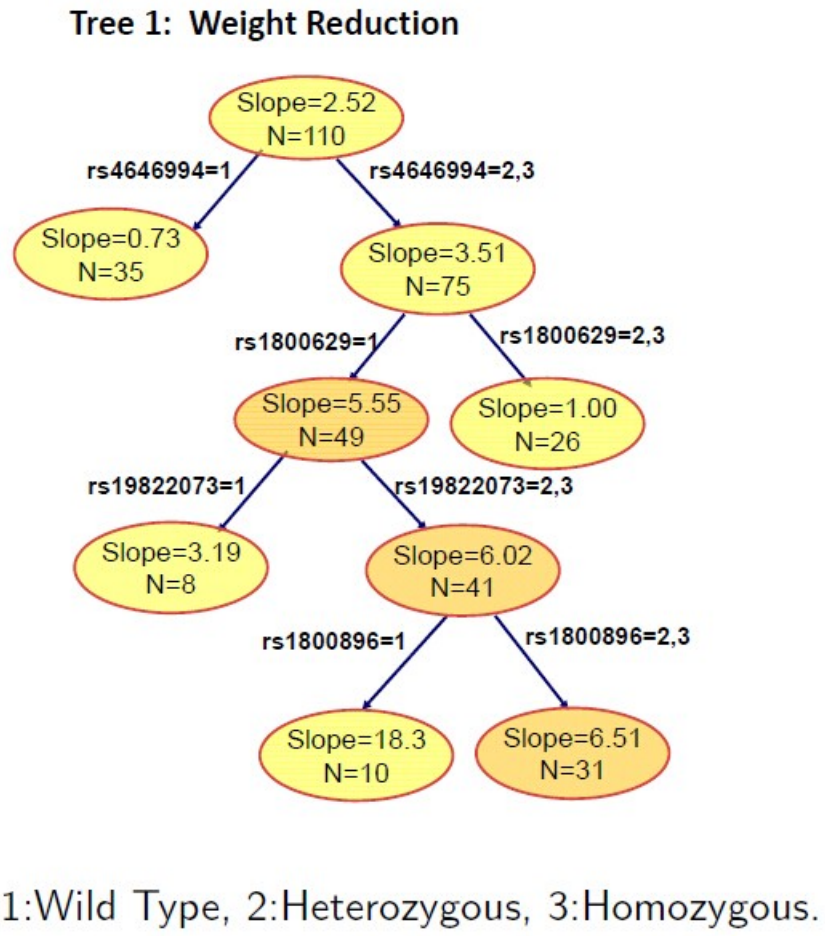

Figure 3.2: Recursive Tree Result for Weight Loss Group

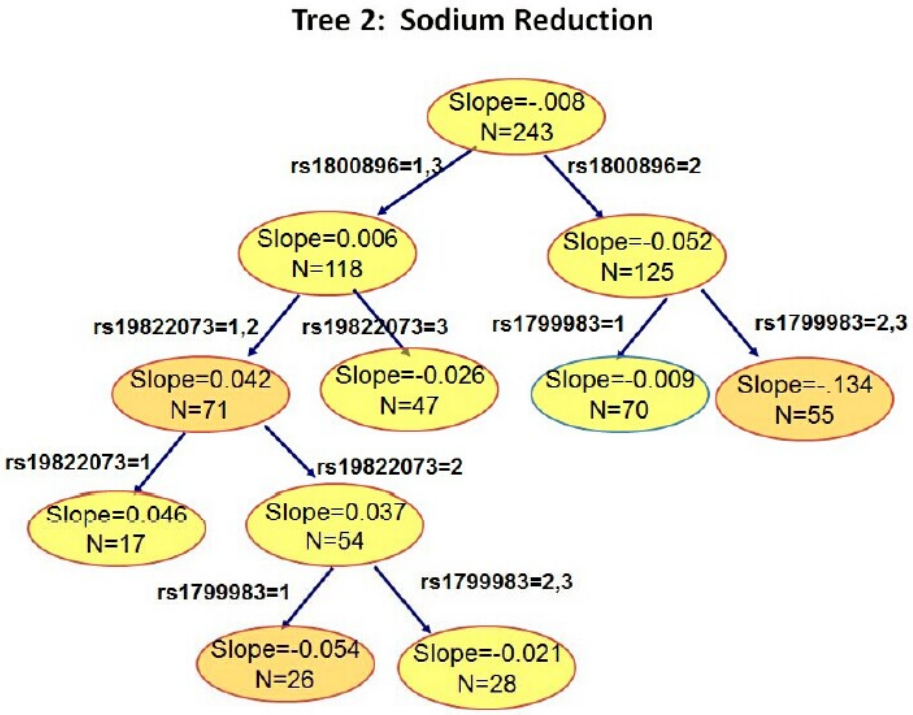

Figure 3.3: Recursive Tree Result for Sodium Reduction Group

Chapter 4

Exhaustive Search Approach

4.1 Motivation

We have introduced the tree approach in the previous chapter. The model is easy to interpret and understand, even for people without much statistical background. The binary tree also preserves as much data as possible at each level. It can be seen that if a parent has more than 2 children, the size of observations in nodes will decay more quickly. Such partitions are contained in potential splits of binary trees, therefore the binary structure is favored. On the other hand, the hierarchical structure is sensitivities to the data. It is obvious that every lower level of partition depends on the previous partition heavily. If a observation is incorrectly partitioned into a child node, there is no way to correct it and the error will be propagated to the terminal.

This is the cost to pay for getting computational and structural convenience. Researchers have proposed ways to solve it by using more trees instead of a single one for supervised learning problems. For our task, one impact from the tree structure is: if we already have a subtree with k levels, i.e. it has been split at most 2^k times, and we are limited to the 2^k genotypes for further combination. 2^k is exponentially large, but we need to consider that there are termination conditions which prevents the tree grows to a complete tree. Such conditions also limit the hight of the tree. Therefore the total genotypes is far less than 2^k. which means many interesting combinations are suppressed as the candidate for further split. In other words, the tree structure obtains the local optimal objective but may not achieve the global optimal. Many combinations are ignored in the tree approach.

So in this chapter, we would like to take all possible combinations into account with caution. Another drawback of the traditional regression tree is that the predicted value

are step functions on the feature space. In general, we expect the the modeled function to be continuous or at least piecewise continuous. Since the tree will be pruned and terminated instead of a very fine final partition, it won't be able to approximate to the continuous response. It is improved in our version since we model the continuous response within each feature region. But we did not overcome it completely since we only have piecewise continuity for the response. The tree approach starts from the genotype directly. Exhaustive search approach starts from a slightly different point.

The way of starting from genotypes are of natural thinking, but there is a mapping issue. We know that each node consists of certain participants and they were identified by the tree path. But there is no guarantee that the subset is uniquely determined by the tree's path. In fact, multiple combinations will yield the same subset of people. Let us summarize it below:

- Distinct SNP combinations may yield the same subset
- Adding an additional SNP may not change the subset
  - The added SNP identifies the majority of the treatment group.
  - It excludes those have been eliminated from current combination.

We would like to start with the subsets of observations and map back to genotypes. This can be viewed as an alternative to the way tree model does. It does not improve the essential analysis but is convenient for computation. We use this exhaustive model to overcome some drawbacks and provides alternative solution to the tree approach.

4.2 Methods

We start with the objective at the beginning of this section: searching for all possible genotypes and corresponding nodes. Genotypes are constructed by a subset of polymorphisms. We use the total number of involved polymorphisms as a parameter to control the searching process. Meanwhile, we do not want to specify which polymorphisms when generating genotypes since the combination rule makes it hopeless to list

all. One advantage of tree model is that it is easy to obtain the next level from the current recursive tree, which means it is convenient to involve more polymorphisms. The property is also desired in the exhaustive approach. The way of representing genotypes are different from that of the recursive tree model, so we need a way to determine the total number of polymorphisms in the genotypes. Let us start with the first level of the polymorphisms.

4.2.1 Binary Representation for a Single Snip

The way of representing a subgroup of all participants is not unique. The one we choose is "0" or "1" representation as indicators. Suppose we have a fixed order of all participants. For any subset of the whole set, we may use a "0" or "1" vector to express it, by letting "1" indicate some participant belongs to this subset at the corresponding position. Since the order of all observations are fixed, if the i-th element of the vector is 1, then the i-th patient is in the subset. So each subset corresponds to a vector and all vectors form a set with cardinality of 2^N, where N is the number of observations.

We would like to connect the binary representation with the snip information. We start with a single polymorphism for a single observation, and then extend it to a group of people; eventually we would get every snip involved. It is known that a single polymorphism can be Wildtype, Heterozygous and Homozygous. For a particular person, he/she can only be one type out of the three and it is expressed as binary vectors easily. If a person is Wildtype, we may code the corresponding genotype as (T, F, F) to represent that this person is true as a Wildtype but not Homozygous or Heterozygous. It is obvious that we can use a single digit to represent it since we may use $(1, 0, 0)$ as an alternative to (T, F, F) and the binary 100 is 4. But the reason that we did not want to go further by using a single digit is that later we need to use the representation for more observations and polymorphisms. "*And*" and "*OR*" operations are easy to compute for binary numbers, other representations may require extra work to convert back to 0's and 1's.

The three biological genotypes are of nature but we would like to extend it to three more. The underlying reason is that there is no evidence whether the sensitivity is

Genotype	Representation
Wildtype	(T F F F T T)
Heterozygous	(F T F T F T)
Homozygous	(F F T T T F)

Table 4.1: Binary Representation of a Single SNP

different across from all genotypes. It is possible that two of them are identical while the other are different, in terms of the sensitivity. For instance, a person of Homozygous might exhibit the same sensitivity as another person of Heterozygous, therefore being Wildtype or not is the factor that affect the benefit of blood pressure drop given weight or sodium reduction. So we would like to ask if a person is Wildtype or not, Homozygous or not, and Heterozygous or not. So in the example earlier, we would like to add three more elements to the vector. Suppose the ordering of elements are: Wildtype, Homozygous, Heterozygous, Not Wildtype, Not Homozygous, Not Heterozygous, then a Wildtype observation is represented as (T, F, F, F, T, T). Similar representations are for Homozygous and Heterozygous people. Table 4.1 shows three cases for a single snip on one observation. Our next mission is to use this representation for all observations and focuses on a single SNP.

Suppose we have five observations and their genotypes for a particular SNP is: Heterozygous, Wildtype, Homozygous, Homozygous, and Heterozygous. Each person is corresponding to a binary representative with length 6. Combining them row by row, we will get a 5 by 6 matrix. This matrix represents the genotypes of these people for a particular polymorphism. We extend this matrix to all observations within a particular treatment group. The weight loss group contains 110 participants, so its associated representation matrix has a dimension of 110 by 6. Table 4.2 illustrates the matrix for five observations and T, F are replaced by 1, 0, respectively.

4.2.2 Binary Representation for a General Genotype

In the previous section, we have introduced the way of producing binary representations for a single SNP. We would like to generalize the procedure to multiple SNPs. We have

Observation	Representation
1	(0 1 0 1 0 1)
2	(1 0 0 0 1 1)
3	(0 0 1 1 1 0)
4	(0 0 1 1 1 0)
5	(0 1 0 1 0 1)

Table 4.2: Matrix Representation of A Sample with 5 Obervations

shown how to produce the binary matrix for any set of observations. We perform the same generating steps as the previous section for all 21 polymorphisms. We call these matrices as indicator matrices. They are named in this way because each column in the matrices is a genotype. Each column has the same number of binary elements and if it is 1, then the corresponding observation belongs to the genotype. We have 21 matrices and based on them, we are able to tell if a particular participant belongs to a particular genotype. Now we are able to tell if they belong to a genotype consisting of two polymorphisms, say A and B. If the person does not belong to A or B, then we can tell that this person would not have the genotype generated by A and B. He/She will be of this genotype only if the person belongs to both A and B. This logic is the same as OR operator. This is one reason that we use the binary representation instead of others. This step is key to the exhaustive approach. Alternative representation might be: using strings to produce the genotype directly, get subset of patient id's, etc. The first representation is more natural to read for human beings. For instance, if the genotype A and B are rsA and rsB. rsA is homozygous or wildtype and rsB is heterozygous. Let us use 1, 2, 3 to express wildtype, homozygous and heterozygous in this example. then this genotype is $rsA! = 3$ & $rsB == 3$.

In order to express this genotype to be a way that the R program can recognize, we have to replace all unequal by equal, and use brackets to assign priorities. In order for the program to understand the genotype, it may have to be rewrote as $(rsA == 1||rsA == 2)\&\&(rsB == 3)$ in R, where $||$ is OR and $\&\&$ is AND. It is more redundant than the natural way of expression. Considering the total number of polymorphisms,

the length of such expressions will increase rapidly. Saving all paths in the memory is an extensive task. For this project, this approach can take about five polymorphisms without using additional processes on a main steam desktop. It generates paths directly from the genetic information and then find the patients. It does not implement any special structures and its results will include all tree paths in the previous chapter. Due to the efficiency consideration, we decided not to proceed with this approach, even it works naturally for small number of polymorphisms.

Alternatively, we may express the genotypes based on the subgroups using observation id. This is similar to our exhaustive approach. We use binary representations, and each column/path is correspondent to a set of participants. We may use the subsets and union between them to produce more complicated paths. This is equivalent to the exhaustive approach. The issue is that how to get the union quickly and know how the path is generated by which polymorphisms. This provides a practical consideration about the exhaustive method. The binary expression has additional benefits that works for a long path with sufficient 1's. We do not want to investigate a genotype with insufficient sample size. So attention will be paid to the subset with relatively large size. Therefore the list of observations will not be a factor to help saving storage space. We will discuss it in a later section. These considerations lead to the current approach. We formulate the statement above as below.

In general, let us assume that there are M polymorphisms. Using the indicators introduced in previous section, we obtain a binary matrix for each:

$$A_1 = \begin{pmatrix} 0 & 1 & \cdots & 1 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & 1 & \cdots & 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & \cdots & 1 & 1 \end{pmatrix}, \cdots, A_M = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 1 & \cdots & 1 & 0 \end{pmatrix}.$$

P_i is the indicator matrix for the $i-th$ SNP. It has six columns and the same number of rows as the sample size. Every column from these matrices corresponds to a realized genotype, called a path. The second step is to get a more complicated genotype from two matrices. What we do is choosing two columns from two indicator matrices from A_1 to A_M and perform an element by element product. WLOG, we would like to produce

a genotype from polymorphisms i and k. Let j and l are the column indices.

For $j = 1, 2, ..., 6$; $l = 1, 2, ..., 6$;

- Select two paths $\tilde{x}_{i,j}$ and $\tilde{x}_{k,l}$ from i-th and k-th indicator matrices respectively
- Calculate $\tilde{x}_{new}[s] = \tilde{x}_{i,j}[s] * \tilde{x}_{k,l}[s]$,

  $s = 1, 2, ..., N$,

  $j = 1, 2, ..., 6; l = 1, 2, ..., 6$,

  $s = 1, 2, ..., M - 1$,

  $k = i + 1, i + 2, ..., M$,

All $\tilde{x}_{new}$ are the paths of two snips' combination.

Let i, k go over every pair of polymorphisms, we obtain all new $\tilde{x}$ and they form all genotypes generated by any two polymorphisms. We use the notation of P_1 to be the matrix consisting of all paths from A_i's, $i = 1, 2, ..., 21$. Let P_2 be the matrix contains all such $\tilde{x}_{new}$. The number of columns of P_2 is $18M(M - 1)$.

After obtaining the genotypes from two indicator matrices, we extend to more polymorphisms. The extension is feasible based on the previous results. Let P_k be the paths involving k polymorphisms. We know that $P2$ can be generated by using all pairs of distinct columns in P_1, or can be viewed as two different columns from two P_1's. P_3 will be generated by using one column from P_1 and the other column from P_2. To obtain P_k, we use all columns in P_1 and P_{k-1} and multiply together as above. In this way, we are able to produce genotypes with as many polymorphisms as possible.

4.2.3 Redundant Paths

Due to the limited sample size, we can not get 1 to 1 correspondence between paths and genotypes. Although each path is generated by a genotype, but multiple genotypes may produce the same path and it happens in our data. When we produce P_k iteratively, we perform a simple check: compare the newly generated path with the paths in P_1.

If there is not change from P_1 then we consider the new path to be not informative to us since even introducing more polymorphisms, it only removes a minority of the observations and does not change the subset. Suppose the number of new paths are K, then this check takes $6 * K$ comparisons of vectors. We did not perform the self-check: i.e. compare all pairs of paths in P_K. The reason is that is quadratic instead of a linear number of complexity. It requires $\frac{K*(K-1)}{2}$ comparisons. We perform it later after the sample size check. The reason of checking sample size, i.e. counting the number of 1's in a path, is the same as one stopping criterion in the recursive tree model: we want to have sufficient number of observations having a particular genotype. We use the same cut-off number as the tree model. The sample size check is also performed on the P_1

Group	Minimal Split	Minimal Bucket
1,2,3,4	40	25
5,6	60	30

Table 4.3: Parameters of Nodes Sizes

before proceeding to more polymorphisms. For simplicity, we still call these indicator matrices P_i, $i = 1, 2, ..., K$, after paths with insufficient 1's are removed.

4.2.4 Practical Issues

One advantage of this approach is that we do not have to maintain all genotypes in the memory; they can be saved as a formatted matrix in a text file. All such matrices are in the same number of rows. It is easily to flip the matrices over and keep each path as a row. Given the storage on disk and row based form, we are able to deal with the large amount of data on the hard drive.

Typical big data questions are in face of several challenges. One is the huge amount of data. Other other is the computation time. We are facing the data amount issue. To solve the storage issue, there are distributed file systems for people to use. But if there is no such a resource available, we may consider using the hard drive instead of the memory. Usually hard drive has much more capacity than the memory (¿10 times space). But we also know that the computation on hard drive will be very slow due to the I/O limits. Therefore we would like to reduce the amount of computation on the

hard drive.

In the previous section, we have reduced the amount of data to process in some degree. In addition to that, we only perform the necessary part on the hard drive: producing genotypes. In order to increase efficiency, we use C++ instead of R for this part of processing. In the C++ program, we use string operations to generate new paths and save each P_i in a txt file. As we mentioned earlier, we use rows for the binary paths instead of columns since it is more convenient for the program to read and write. By doing so, we partially solved the computation issue. We also save the vector as a string such as "010010..." instead of integers of elements. It helps to save more space. But the text files are still large and we want to read it into the memory for further analysis. For instance, if we include 722 observations, and produce the indicator matrices with up to five SNPs, the txt file will be approximately 90 GBs. Hence we need more to fit it into the memory of a desktop. We do not want to get more than O(n) in terms of the number of strings. Note that here the complexity are expressed with unit of a string or a genotype. It is not measured by the total elements in the indicator matrices. One reason is that we do not always have the row vector in these matrices. Earlier in order to save space, they have been compressed to strings. The length of the string can be as large as $2^31 - 1$, which is about 2 billion. The observed length is the same as the number of observations under investigation, so we won't have any issue to save these strings even the experiment involves more patients. This is an advantage that we are able to predict the string length. If we start with genotypes directly, it is not possible to predict and maintain the fixed length of all strings.

We can further reduce the size of paths' files given the nature of them. They consists of only 0's and 1's, with a fixed total number. In this case, we are able to compress the stings even more. The first step is to convert them back to vectors from strings. The second step we do is to use a different encoding scheme to express 0's and 1's.

4.2.5 Tetrasexagesimal Encoding Scheme

Tetrasexagesimal is a collection of binary-to-text encoding schemes. They are used to express binary data with ASCII characters to save space. Such schemes are also

called Base64 schemes. The reason of this name is that it maps 64 different binary strings/vectors to different ASCII letters. It is widely used in applications such as emails to save storage. The differences between Base64 schemes are minor: they have different mappings and slightly different ASCII codes. The principle is to select 64 ASCII characters that are printable and can be input by the keyboard. These characters typically include all letters A-Z, a-z, all numbers 0-9, and 2 more special characters. There are other Base32 or Base16 encoding schemes, they use less printable characters and compress less. Basically, Base32 achieves half of the compression rate of that of Base64, if the original coding are in binary format. Base64 is a typical choice for another reason: it utilizes most printable characters on the keyboard. Table 4.4 shows how one can choose to convert the binary string to characters:

Binary	Char	Binary	Char	Binary	Char	Binary	Char
000000	A	010000	Q	100000	g	110000	w
000001	B	010001	R	100001	h	110001	x
000010	C	010010	S	100010	i	110010	y
000011	D	010011	T	100011	j	110011	z
000100	E	010100	U	100100	k	110100	0
000101	F	010101	V	100101	l	110101	1
000110	G	010110	W	100110	m	110110	2
000111	H	010111	X	100111	n	110111	3
001000	I	011000	Y	101000	o	111000	4
001001	J	011001	Z	101001	p	111001	5
001010	K	011010	a	101010	q	111010	6
001011	L	011011	b	101011	r	111011	7
001100	M	011100	c	101100	s	111100	8
001101	N	011101	d	101101	t	111101	9
001110	O	011110	e	101110	u	111110	+
001111	P	011111	f	101111	v	111111	-

Table 4.4: Mappings of Base64 Characters

We also need to deal with the cases that the strings are typically not evenly divisible by 6. We may leave the reminder part as it is since its length is at most 5. So keeping this part as uncompressed would not increase too much storage. The worst case is that the original sting is of length 5 or 11. If the length is 5, the compression rate is 0. If it is 11, then after the converting, the length is 6.

We need to distinguish the converted part and reminder part. The reason is that

"0" and "1" are also used as the converted characters in the scheme table while original reminder also consists of these two chars. We may use another special character to sperate them. The one we use is "&". Figure 4.1 shows an example that how the encoding is done for a given string. We match every six binary elements from the beginning, map it based on Table 4.4, until there is insufficient elements. And then we insert a septation mark "&" and add the rest as a substring.

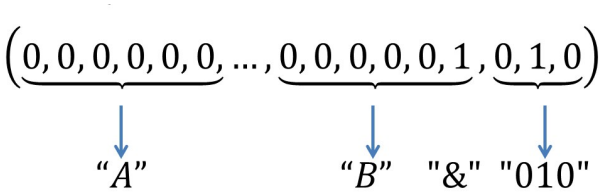

Figure 4.1: An Example of Base64 Compression

After converting strings to Base64 encoding scheme, we read it into the memory , identify duplicates and remove them. Finally, we convert them back to binary format. The reason of performing removal step in the memory is that we can apply efficient algorithms quickly. The strings before and after conversion has 1-1 and onto relation. So we can look for the duplicates in the Base64 format. Hash set can achieve O(n) complexity, or use sorting algorithms to achieve O(n*logn). Ram operations will be much more efficient than performing the same task on the hard drive. After removing all duplicates, we convert the strings back to binary paths. For the all patients group, with 5 snips, the paths' file is as large as over 90 GB. We recode them and reduce to 17 GB; this is the step 1 in Figure 4.2. After removing duplicates, they become 2.4 GB in the compact version; it is the step 2 in Figure 4.2. Finally they are expressed as the binary form with the size of 13.8 GB. If the compress version is still too large for the computer after step 1, we may divide and conquer: partition the original binary string file into multiple, and perform the same steps 1 and 2. After that, combine them in the memory and look for duplicates across partitions, which means we need an additional Step 2 for divide and conquer. The final step is the same.

We use C++ for this encoding change and the rest are done in R. There are some tips for R users when dealing with relative large data sets:

$$\approx 90GB \overset{①}{\Longrightarrow} \approx 17GB \overset{②}{\Longrightarrow} \approx 2.4GB \overset{③}{\Longrightarrow} \approx 13.8GB$$

Figure 4.2: Change to Storage Space of Processed SNP Data

- Allocate memory at startup
- Avoid large number of columns
- The memory may not be released to OS immediately
- Manually control the intermediate objects
- Utilize existing HPC packages

4.2.6 Ranking of Paths

In the recursive model, we implicitly produce genotypes as the paths in the trees from top node to terminals (and their sub-paths). Each of them corresponds to a particular subset. We perform regression models on each subset, and compare the sensitivities to the top node. In the exhaustive approach, all subsets of interest are obtained directly from indicator matrices. We would like to perform the same task and compare the sensitivities. There are two differences from tree approach when the comparison is done:

- Number of testings: large vs small
- Compare genotypes or paths

The second difference is that we would like to investigate the sensitivity difference for each group, and see which corresponding genotype is associated with the particular group. It happens that multiple genotypes produce the same subset. We will report all later. This question will be left for further investigation.

The first difference is that because the tree model has multiple conditions to stop and also a way to trim down. The stop condition includes insufficient sample size and insufficient benefits of further splits. The exhaustive search approach does the sample

size check but does not incorporate other stopping criterions. We will introduce how to set the total number of polymorphisms in the next section. As a result, each tree only contains several genotypes to compare but exhaustive results contains a large number of paths. Therefore, false discoveries become an issue and we want to adjust for it. Dr. Cabrera proposed Permuted FDR to solve this problem. The idea is as below:

- Perform analysis for all genotypes/paths and obtain p-values with ordering $p_{(i)}$
- Generate a null hypothesis by permuting the genetic part of the data
- Each permutation produce a set of new p-values, find 0.5 and 0.1 percentile of them, denote as $p_{(i),\alpha}$
- FDR correction: $q_{(i)} = p_{(i)}/p_{(i),\alpha}$. Use q's as the adjusted p-values.

Earlier we have introduced the difference between two components of the data: snip part and clinical part. The null hypothesis is that these two are independent. If this is true. then we may reassign the clinical data of one person to another without changing the analysis results much. If the null is not true, then the clinical observation (change of blood pressure in our case) is related to their genotypes. The permutation randomize the associations and allows researchers to recalculate statistics for each permutation. The permutation provides us with the baseline of false discovery rate to have significant results. We adjust the calculated p-values based on this and re-rank the top significant genotypes/paths. But one permutation might be not sufficient for this analysis and might be biased, so we permutate 100 times and pick the median p-value as the base level for adjustment. By multiple permutation, we get the empirical distribution of the statistic under the null. Another underlying assumption is that all genes has the same frequency to appear. Figure 4.4 illustrates the way of permuting the associations between 6 observations. The response variable is also included in the right panel of data when doing permutation.

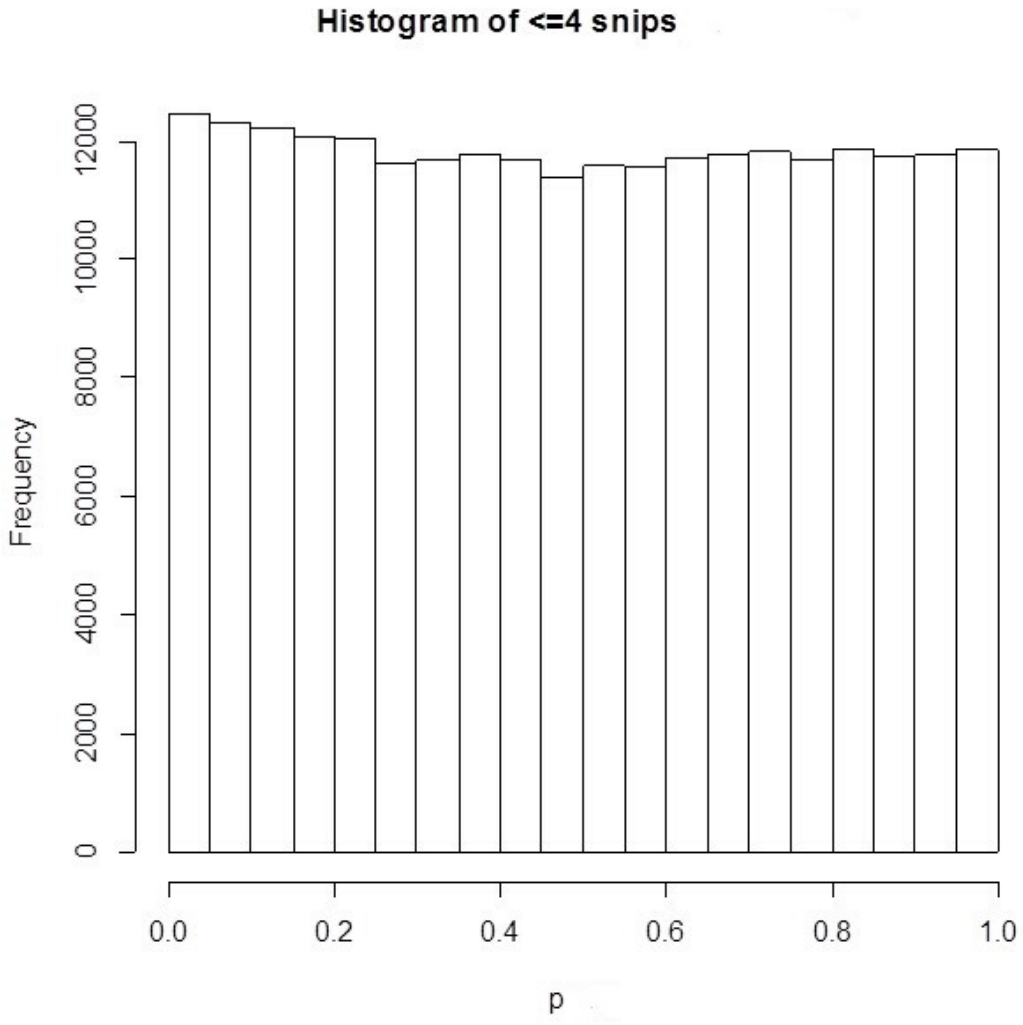

Figure 4.3: Histogram of P-values before FDR Adjustment

4.2.7 Termination Levels: Multidimensional Scaling

One question that we haven't solve is that how to know when to stop generating new paths. We have seen from previous sections that we are able to proceed iteratively. So technically we can get all 21 polymorphisms involved to generate the paths. One true barrier is that we only have about 700 observations. Figure 4.4 shows the number of paths versus the total number of snips, without removing redundant strings. Y axis is the base 10 log of the total paths counts. We are able to see that it increases quickly first and then slower. But it is a huge number compared to our sample size. With more polymorphisms, the new path are more like to be the same as before, which introduced

	rs1	rs2	rs3	rs4	rs5
S1	WT	HO	WT	HT	HT
S2	HT	HO	WT	WT	WT
S3	HO	WT	HO	HO	WT
S4	HT	HT	WT	WT	HT
S5	WT	WT	HT	HT	HO
S6	HO	WT	HT	HO	HT

	Age	Sex	BMI	SMK	...
S1	67	M	27	Y	...
S2	72	F	25	N	...
S3	65	M	32	N	...
S4	69	M	29	Y	...
S5	75	F	22	Y	...
S6	66	M	26	Y	...

Figure 4.4: Example of Permutations for FDR

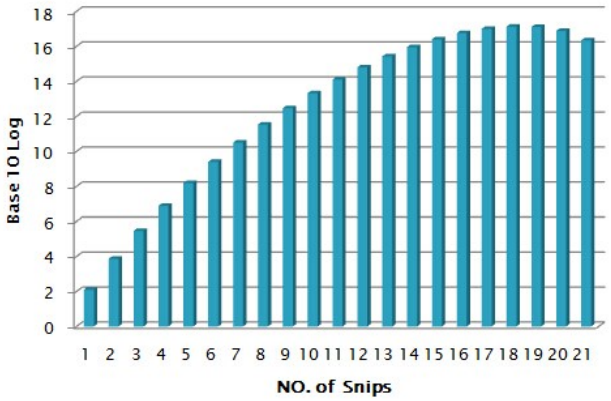

Figure 4.5: Number of Paths v.s. Number of Polymorphisms

more noise to the analysis. Another issue is that with more snips, the size of subgroup is smaller since there is a chance that some observations do not satisfy the genotype requirement, and will be eliminated from the group. So newly generated paths are more likely to have insufficient same size and do not contribution to the analysis. In contrast, these paths take storage and computing resources and it is not desired. One possible solution is that we use the information from the tree results. For instance, suppose we look into the weight loss group, the tree result has a maximum height. or the significant nodes have height information, we may use that number to decide how many snips to use in the paths. But it is preferred that we determine the levels independently, so that the results from two approaches can be used for validation purpose. The motivation of validating the results is that we are doing exploratory analysis and want to verify the results obtained by the previous approach.

It has been mentioned in the paragraph that adding more snips to the paths may not gain more paths accordingly, due to the limited sample size. We would like to use this information gain to determine the number of total snips in the paths. We use the idea of multidimensional scaling (MDS) to answer the question visually. The prerequisites is summarized below:

- Generate all paths up to a level
- Calculate the sensitivity on all associated subsets
- Test the difference to the whole treatment group
- Adjust the p-values with FDR

After all prerequisites are satisfied, we are able to proceed to the next task. The steps of applying MDS to the exhaustive paths is as below:

- Determine the maximum number of levels K
- Select top significant paths
- Define and calculate distances
- Project to a lower dimension
- Visually determine the cutoff level

The first step is necessary because in the second step, we look for the top paths out of the candidates and the total number of candidates varies from different K's. I choose the top 5% 20% based on the visual density of points and also the change of the shape. The MDS test is also a repeated procedure because we will gradually increase the K and compare it with $K + 1$. If the results of $K + 1$ does not show any advantage over K, then K would be selected as the optimal cutoff. The idea of this select is partially motivated by the selection of k in k-mean. The difference is that we would like to see if the new layout will be represented by the old layout, where layout are the points of reduced dimension for certain K, new refers to an additional snip in the paths.

The third step is to reduce the dimension of paths so that we are about to print them on a 2-d plane for visual check. Details will be provided. The forth step is simply showing the layouts to researchers and look for the clusters and representative points.

The main objective of MDS is to reduce the dimension of a collections of points. The way it achieves it is that MDS tries to preserve the relative positions between each other. In a high dimensional space, all points have a distance to any other points. Such distances determine the positions of these points uniquely up to shift, rotation and reflection. When reducing to a lower dimension, the structure can not be preserved perfectly; some distortion would be introduced. Multidimensional scaling's objective is to minimize such distortion. In our problem, the points are the paths. We would like to see if current paths are representative enough for new paths. If it is the case, then there is no need to look at more varieties of subsets. There are multiple versions of multidimensional scaling and we use the classical version. Let P_i be any path, and for any pair of P_i and P_j, we can define the distance between them as $d_{i,j}$.

$$d_{i,j} = \frac{\sum_m |(P_i(m) - P_j(m))|}{\text{length of } P_i}, \; m = 1, 2, ..., \text{length of } P_i.$$

$d_{i,j}$ is the proportion of distinguished observations between subset P_i and P_j. Given that the length of all paths are the same, this definition is essentially Manhattan distance. We define the distances for all pairs of paths and obtain a distance matrix D. Assume the total number of paths is S, then $D = \begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,S-1} & d_{1,S} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,S-1} & d_{2,S} \\ \vdots & \vdots & \ddots & \vdots & \\ d_{S,1} & d_{S,2} & \cdots & d_{S,S-1} & d_{S,S} \end{pmatrix}$ We need to specify the dim of subspace that the paths reduce to, say s. Then the objective is to find S vectors in R^s, such that their distances are close to the pairwise distances. Let such vectors be x_i, $i = 1, 2, ..., S$, and it is desired that the distance between x_i and x_j $\| x_i - x_j \|$ is close to $d_{i,j}$. More precisely, we want to solve the problem:

$$\min_{\{x_1, x_2, ..., x_S\}} \sum_{i,j} (d_{i,j} - \| x_i - x_j \|)$$

The choice of s is determined by our motivation: obtaining a 2-d scatter plot. We use $L2$ norm for the distances between x's because it has the property that it is invariant

under translation, rotation and reflection. So essentially it does not change the view of plots.

4.2.8 Results of MDS

We would like to show the results from classical multidimensional scaling. We start with the MDS for 2 snips and eventually go up. The idea is to see the distribution of top paths and determine by the clustering shapes. The ideal case will be seeing the shorter paths are placed in the center of longer paths. If it happens, then it means that it is likely that longer paths are adding variations to the shorter paths, and such long paths can be viewed as shorter path plus noise. The noise is introduced by new polymorphisms that has minor intersection to the current subset. Here longer paths means they are generated by more snips and vice versa. Alternatively we may view the longer path as the intersects of two shorter paths without overlapped snips. For newly added paths, it is not necessary that the path itself corresponds to a small set of observation. It may correspond to the majority of the subgroup but the old path may also identify almost the same subgroup. So the small intersection is conditional, not marginal.

The way we select the number of maximum length is motivated by Elbow Method. It is used in the unsupervised clustering algorithms to determine the number of clusters. As long as there is a metric depending on the number of clusters and the metric can be used to measure some performance, then we may apply this approach. The way of applying Elbow method in clustering is that for each fixed number of clusters, the optimal partition is found, and the total dissimilarity is recorded. It is obvious that with the increase of clusters number, the dissimilarity will be improved. But to avoid overfitting, it is also desired to control the clusters separation. So measuring the benefits brought by additional clusters is the way of applying Elbow method. In our approach, the metric will be different since we do not want to cluster genotypes. Instead we would like to find representative of clusters. The metric will be how are shorter paths close to longer paths in top significant genotypes. We would like to see if it happens for smaller K's and see when such will disappear. We can image that for high level of paths this

approach may not work for two reasons. One is that the noise will increase to the shorter paths since we are facing millions or even more genotypes. Another reason is that there is no guarantee that the increment of 1 is the best option. It is possible that for some snips adding two more to the path will show the pattern of keeping the shorter paths around the center of clusters. This is the case that this approach does not have a solution. But the chance of have such issue is small since we are looking at all possible combinations, so all sub components of a path will be investigated as long as they do not fail to meet the requirement of minimal observations. We start with 2 polymorphisms, and eventually extend to more. The results are for the weight loss group and showed in Figure 4.6. There are two plots in the figure. The left graph

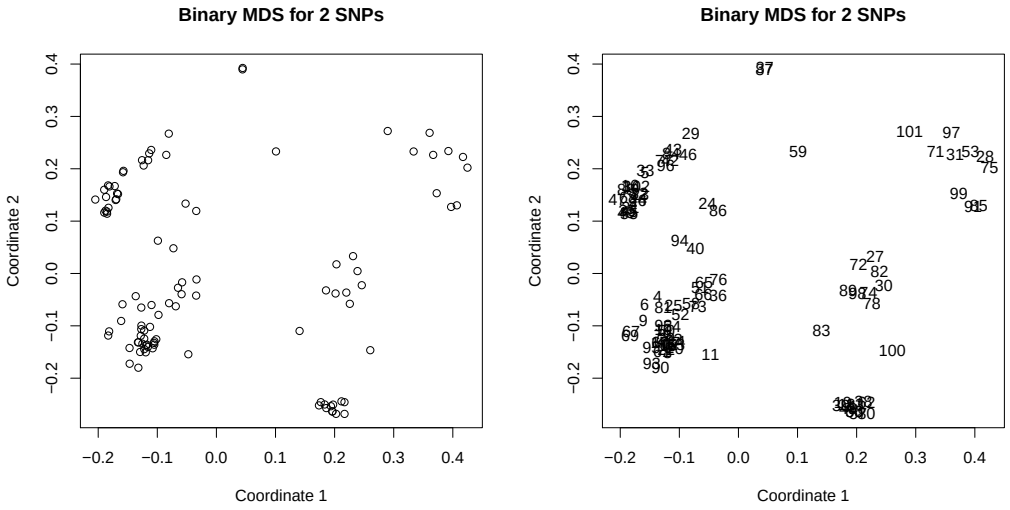

Figure 4.6: 2 SNPs MDS Result for Weight Loss Group

shows all top paths while the right one mark them in different numbers. Marked with different numbers will show the overlapped points better. From the graph, we can see that there are mainly 3 dense clusters and 2 more less dense. We do not want to stop at this level when finding the top genotypes since we prefer to do multiple genotypes instead of a single one. Finding single paths will be similar to GWAS studies. It is worth mentioning that we also investigate the top 2-snip path in more combinations. Because later we want to put paths with different lengths together and separate them. So we are comparing the shapes of two sets of paths. These paths are all of the same

length but obtained differently. Figure 4.6 shows top length 2 genotypes. Figure 4.6 uses not only length 2 but also longer and look at the top paths together. Among them we obtain the same number of genotypes with length 2 and perform the same MDS procedure. We compare the left sides of Figure 4.6 and 4.7. The 3 dense clusters are

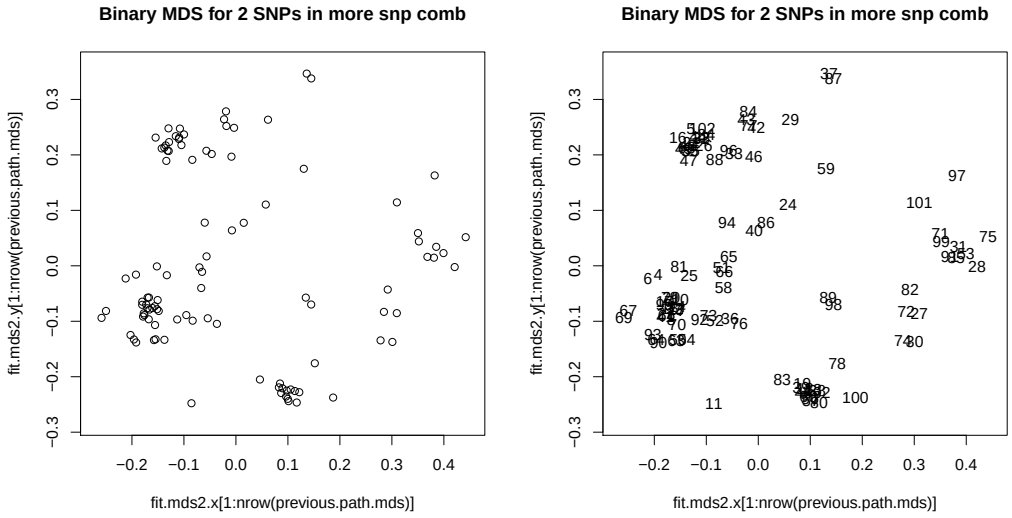

Figure 4.7: 2 SNPs MDS Result for Weight Loss Group

basically preserved at top left, bottom left and bottom right. There are some slight change to the relative position and shapes but they do look similar. In Figure 4.6, one loose cluster is still maintained while the other become more loosen. So based on the outcome, we may say that the main clustering are preserved. In order to make sure that the clusters in the same position are made of the same paths, we assign a same number to each path in two graphs. The first thing we may tell is that the difference is minor in terms of the path assignment. Isolated points such as path 11, 59, 100 in Figure 4.6 are still not associated with any clusters in Figure 4.6. The relative relations in position does not change too much. For instance, path 37 and 87 are almost overlapped before and still close to each other after. We also notice that the members of dense clusters do not change too much, which is agree with the previous observation on isolated points. The reason for the difference between Figure 4.6 and 4.7 is that in the first figure, we only project paths of length 2 while in the second figure, we project paths of length 2 and more; after that we only extract the coordinates that corresponds to the paths

(points) of two polymorphisms, therefore there is a difference between these two graphs. But we have seen that the impact of different is not influential: the number of dense clusters are maintained, the shape is almost unchanged, the distances are less affected and the isolated points are not wrongly clustered into others. So these permit us to proceed to the next step of determine the maximum number of polymorphisms. The principle of this threshold is that we want to capture the majority of top significant paths, without introducing many variations as noises.

Let us plots the MDS for both 2-polymorphism and 3-polymorphism paths. From

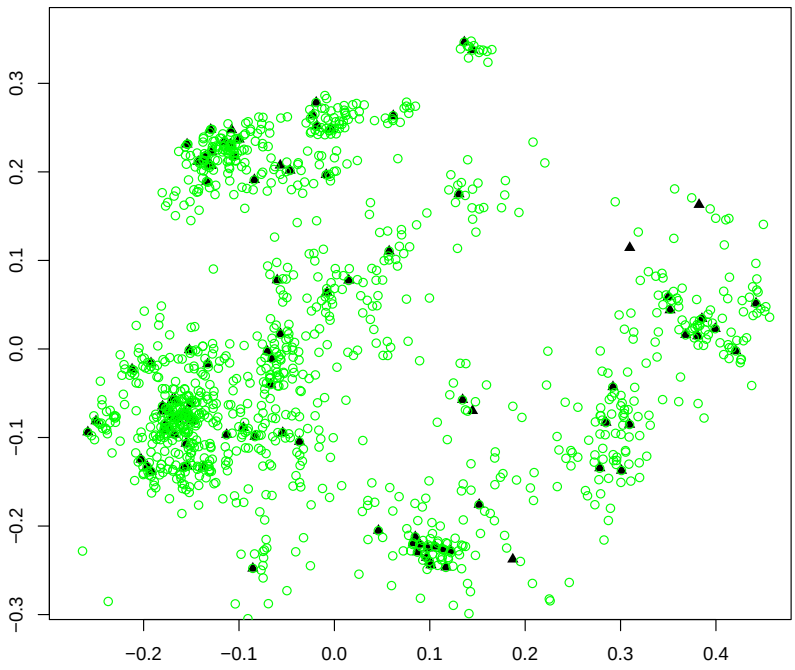

Figure 4.8: MDS Result for Top Paths of Length 2&3

Figure 4.8, we are able to see that there are more clusters than paths with only 2 polymorphisms, which is as expected. There are three of the major clusters that are placed in a similar manner as the length 2 graph, which shows a pattern that we are adding new points in three ways: adding isolated points, adding more nearby points to existing ones, adding new clusters. Since we are able to observe new clusters, we may tolerant the noise: nearby points to existing clusters. It is not clear if isolated points

are noise or not. The reason is that they might be from a potential cluster later, or might remain isolated after adding new paths. So we focus on the newly added clusters and all nearby points to clusters. It is marked as black for length 3 paths and green for length 2 paths. By using different colors, we are able to investigate if the longer paths are variations to shorter paths. It appears to be the case, that we are able to find black triangles near the center of every major clusters. This is a sign that we may consider to reduce from length 3 to length 2 if necessary.

The next step is to see if paths of length 4 can be represented as variations to that of length 3. An issue raised is that how many top paths we want to examine. There is no certain answer to that. Our approach is to plot different number of top paths, and choose the one that is balance between the clusters and representatives. More precisely, we would like to see

- A clean graph of clusters
- Existence of shorter paths in the clusters

After adding more and more paths, the picture will be more messy and we would be unable to tell the clusters in the pictures since we limit the project to be in 2-d. We also desire to have shorter paths as representatives around the center of clusters. To achieve this, we need to add enough paths to ensure that there are representatives in the projection, and also do not want to add too many paths such that clusters are not be identified visually. So we plot multiple top paths after performing multidimensional scaling. Figure 4.9 is a plot of MDS with 100 paths including length 2, 3 and 4. Three different colors are used to distinguish paths of different the lengths. We are able to tell that there is an cluster near the bottom right which does not contain shorter representatives, which means we may want to add more paths. In Figure 4.10, we add more paths. The same cluster in Figure 4.9 does not have any shorter representative but some other clusters begin to have paths of length 2 instead of length 3 as members, which promotes us to add more to the projection procedure. So we obtain Figure 4.11. In this plot, we find that there are pink dots in the less dense cluster that does not have shorter paths in previous graphs, which means that adding the additional paths to

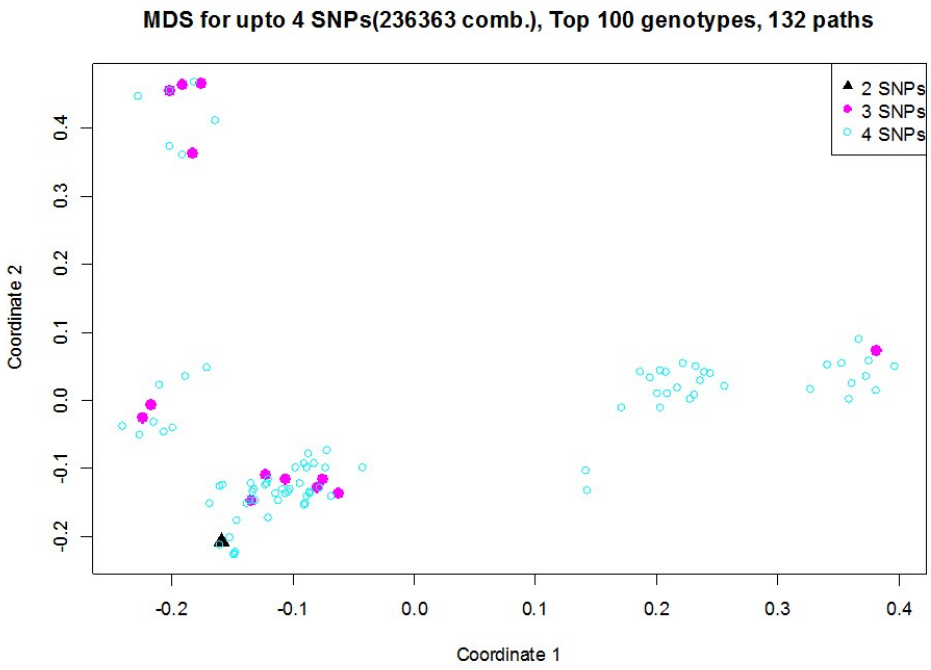

Figure 4.9: MDS Result for Top 100 Paths of Length 4

Figure 4.10 is desired. We keep this procedure until the clustering are hard to recognize or distinguish. Figure 4.12 shows more paths and in principle, we do not see a clear increment of clusters above 500. It is hard to recognize the boundary of clusters above 2000.

We move on to paths including 2, 3, 4 and 5 polymorphisms. The adjust p-value cut off we use is $(0.0005, 0.00075, 0.001, 0.002, 0.004, 0.005)$. We are able to see that the picture is becoming messy, but major clusters are still containing shorter colors; which indicates that there is no need to proceed further.

Let us investigate the case with 6 snips. All cluster are not easy to tell apart, and we notice that there are other colors than gray spread over the graph, which indicates that even there are clusters, we can still find shorter paths to represent those of length 6. So we may conclude that paths of 6 snips can be reduced to the length=5 case. And based on the illustration on these 5 snips MDS plots, we can consider to use paths only involving 3 or 4 snips to represent the "center" of clusters; here center means the view of variation, not a real center based on any distance metrics.

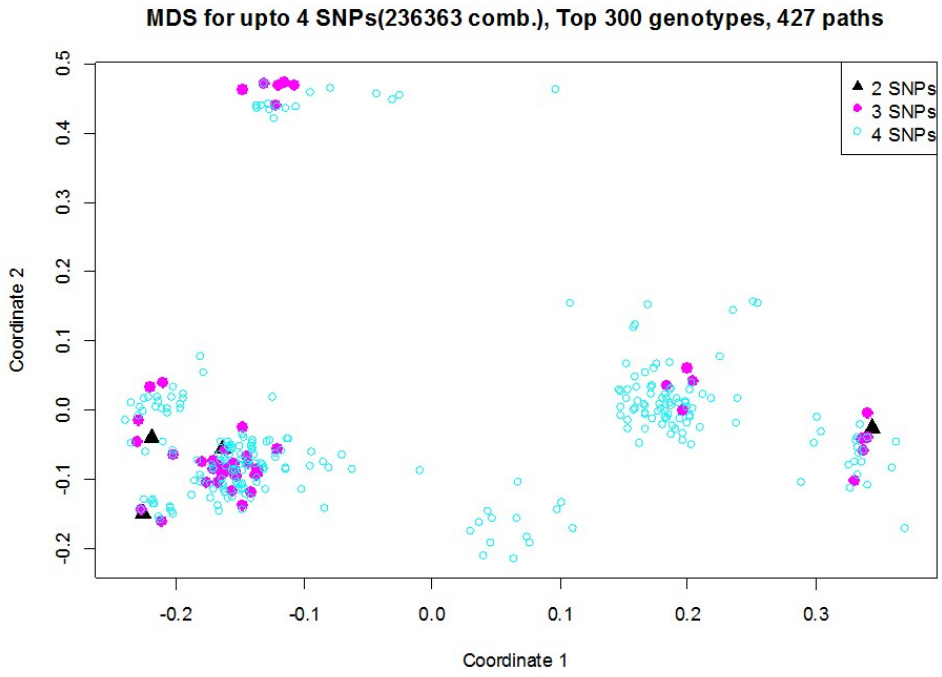

Figure 4.10: MDS Result for Top 300 Paths of Length 4

4.3 Results and Discussions

We restrict our attention to paths involving up to 4 polymorphisms, based on the analysis in the previous section. This number is also agree with the height of the trees that we constructed in the previous chapter. It might be a signal that for the size of our data, we may not want to investigate genotypes with 5 or more polymorphisms. This does not invalidate the strategies when dealing with a large number of genotypes. If we are facing a larger amount of data in the future, such technics will be valuable for other researchers.

Before presenting the results, it is worth emphasizing that the output from exhaustive search and that from recursive tree are different. The exhaustive search is path based while recursive tree is genotype based and they are not 1 to 1 and onto. We need to convert paths back to genotypes. Therefore after we obtain the maximum number of polymorphisms, we pick the top paths and report the genotypes. We do not know how many top paths are worth reporting, but we do believe that higher significance worths more attentions, and such results are helpful for people with hypertension. If they are

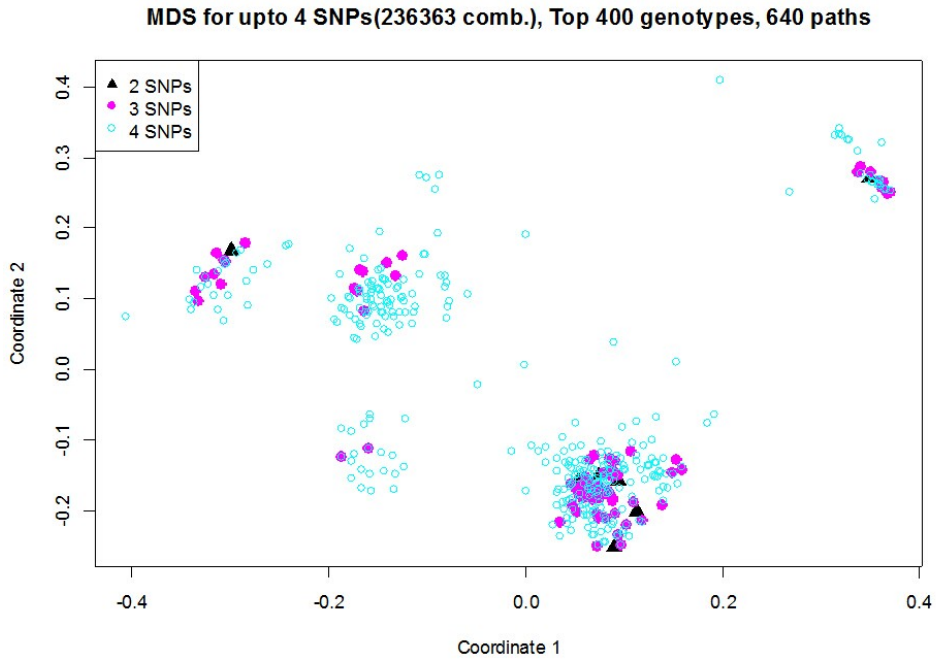

Figure 4.11: MDS Result for Top 400 Paths of Length 4

among the top positively sensitive group, then non-pharmacological interventions are likely to help them reduce blood pressure. Since the top one corresponds to the most benefits, at least based on our data, we would like to focus on such genotypes and make recommendations to patients. If others are among the top negative group, the the results are also helpful. The reason is that we may suggest them to take medication, instead of weight control. Otherwise, given that weight loss or sodium reduce does not help them to lower the blood pressure, but if they are unaware of it and do not want to take medication, then it will make the hypertension worse. We show the results from weight loss treatment group that the sensitivity are significantly positive. Let us compare it with the tree results for the same group (Figure 3.2). The recur-

Rank	Genotype	Size	Slope
1	(rs073=2) & (rs872=1 or 3) & (rs796=1)	31	8.05
	(rs073=2) & (rs872=1 or 3) & (rs796=1 or 3)		
2	(rs994=2) & (rs896=1 or 2) & (rs629=1) & (rs629=1)	35	6.62
3	(rs073=2) & (rs506=1 or 2) & (rs629=1 or 2)	45	6.21

Table 4.5: Top 3 Positive Paths with Last 3 Digits in Names Displayed

sive tree is trimmed down to 4 levels, so the longest path involves 4 polymorphisms;

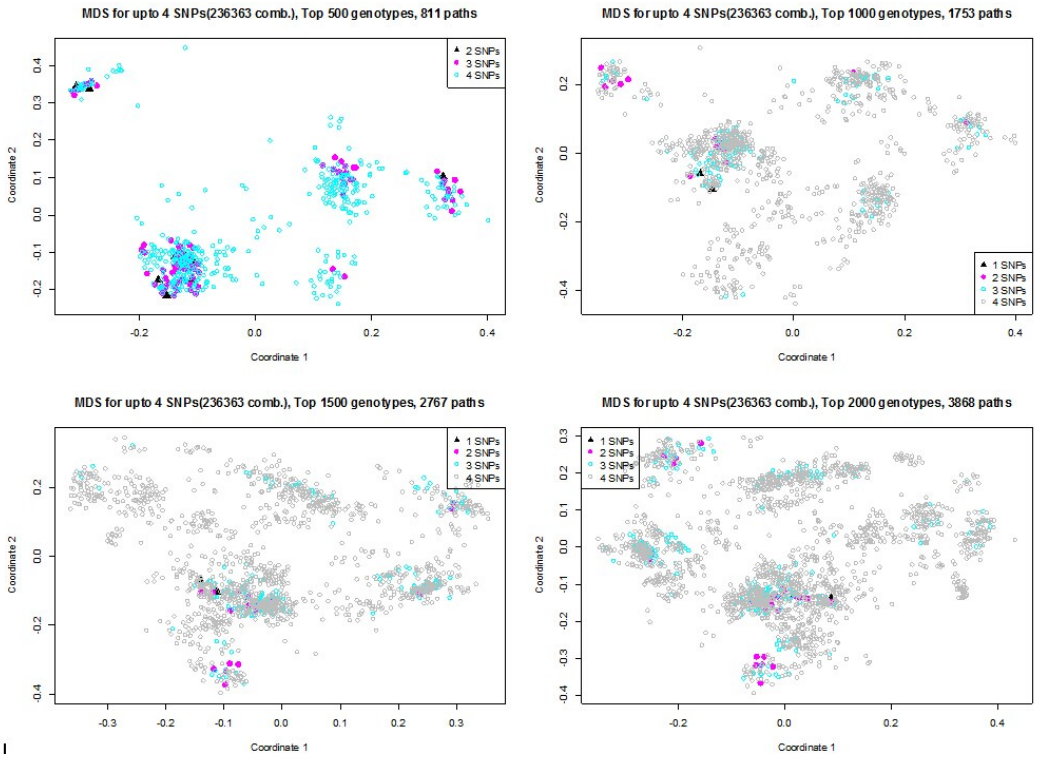

Figure 4.12: MDS Results for Top 500, 1000, 1500 and 2000 Paths of Length 4

meanwhile this tree is not complete, so the shortest path contains only one snip. There are three significant positive nodes with sufficient sample size are marked in the tree. They involve 4 polymorphisms: rs4646994, rs1800629, rs19822073, rs1800896. Those produce the top 3 in the tree nodes. The highest slope is 6.51. In the top 3 paths in the exhaustive results, the slopes are higher as expected. The involved polymorphisms are rs19822073, rs1800872, rs1800796, rs4646994, rs1937506. We do see that there are two common polymorphisms -rs4646994 and rs1882073 and kind of validate the previous results. It might be possible that some polymorphisms refer to the majority of the group. The table below shows the corresponds observation in a certain genotype identified by one of these polymorphisms. We can see that rs4646994 and rs1882073 are evenly distributed among three types of genotypes, so it is likely that these two snips are important in identifying the genotypes. We also see that some of other snips may introduce noises such as rs1800796. We also looked into more snips, let us look at the adjusted p-values and top significant subsets in Figure 4.16. We are able to see that after the adjustment, there are slight change to the monotonicity. All these values

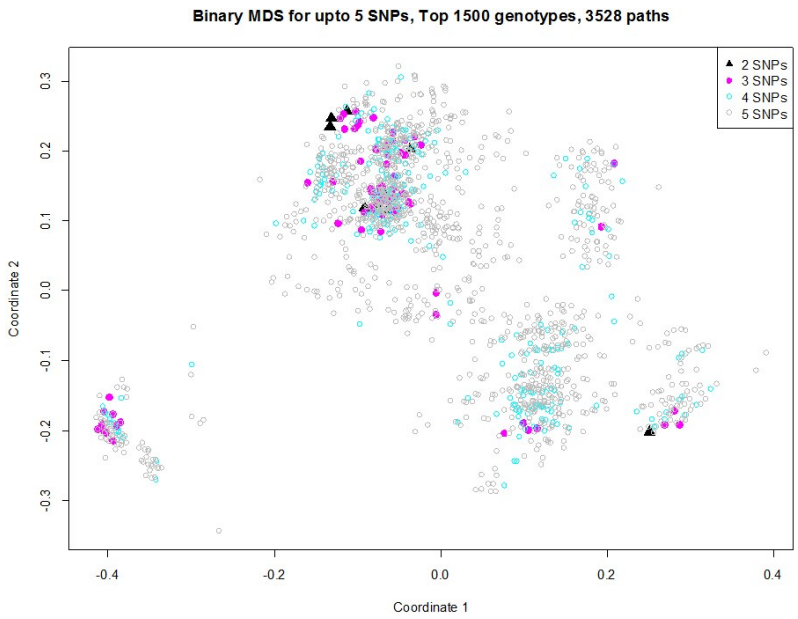

Figure 4.13: MDS Result for Top 1500 Paths of Length 5

Polymorphism	Heterozygous:2	3:Homozygous mutant:3	Wildtype:1
rs4646994	52% (58)	16% (18)	32% (36)
rs19822073	47% (53)	35% (39)	18% (20)

Table 4.6: Distributions in Weight Loss Group for Overlapped SNPs

are arranged in the order that before the FDR procedure. That is why the picture is not monotonic. We also list the top two significant genotypes patients to show the overlapping of them. It appears that the top two groups are overlapped, but they have a large portion of different observations, which means the top 2 paths identify different genotypes.

Let us look at the results of exhaustive approach from another perspective. We desire the Earlier we looked into top few significant result but they may not reflect the case that there are relatively strong sensitivity but not able to be listed in the top few. So we look into the overall distribution of the times that all polymorphisms appear in the top paths. The motivation is that the top several paths may not represent the overall situation that among significantly sensitivity subgroups, how large is the portion that a single snip takes? This is a different question than what we introduced earlier. In this case, we do not care the observed values of the snips while in the previous tree

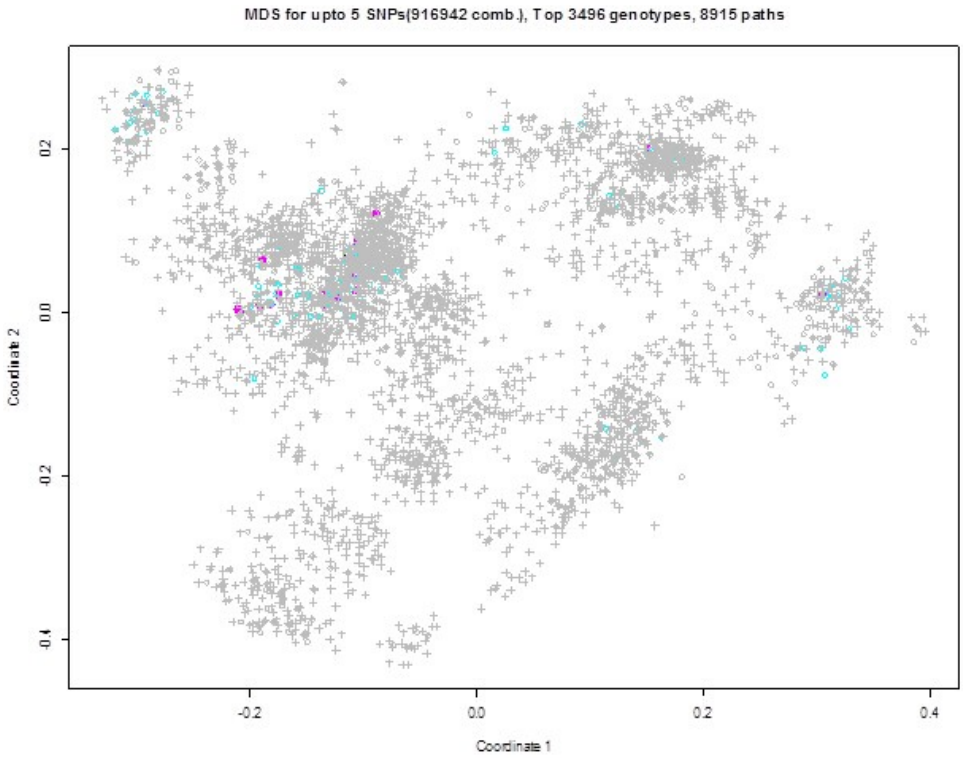

Figure 4.14: MDS Result for Top 3000 Paths of Length 5

and exhaustive search, we care if a snip is Homozygous, Heterozygous or Wildtype. The previous study aims at personalized medicine to help particular patients. The overall distribution provides insight to researchers for further study. What we do here is to identify major clusters visually first. We know that every point in the cluster corresponds to genotypes and they are constructed by several snips. We count the times that a snip is involved in all points of a cluster and draw the histogram. The clusters are constructed by top genotypes. Each histogram is from a cluster. Each column is a polymorphism's frequency in the top and they are in the same order of the list of polymorphisms in Charpter 2. All 3 bottoms and left top are in a similar pattern while the rest two are in different patterns. In the top right cluster, only a few snips are involved in the top significant paths while other clusters involve more. A further investigation might be needed.

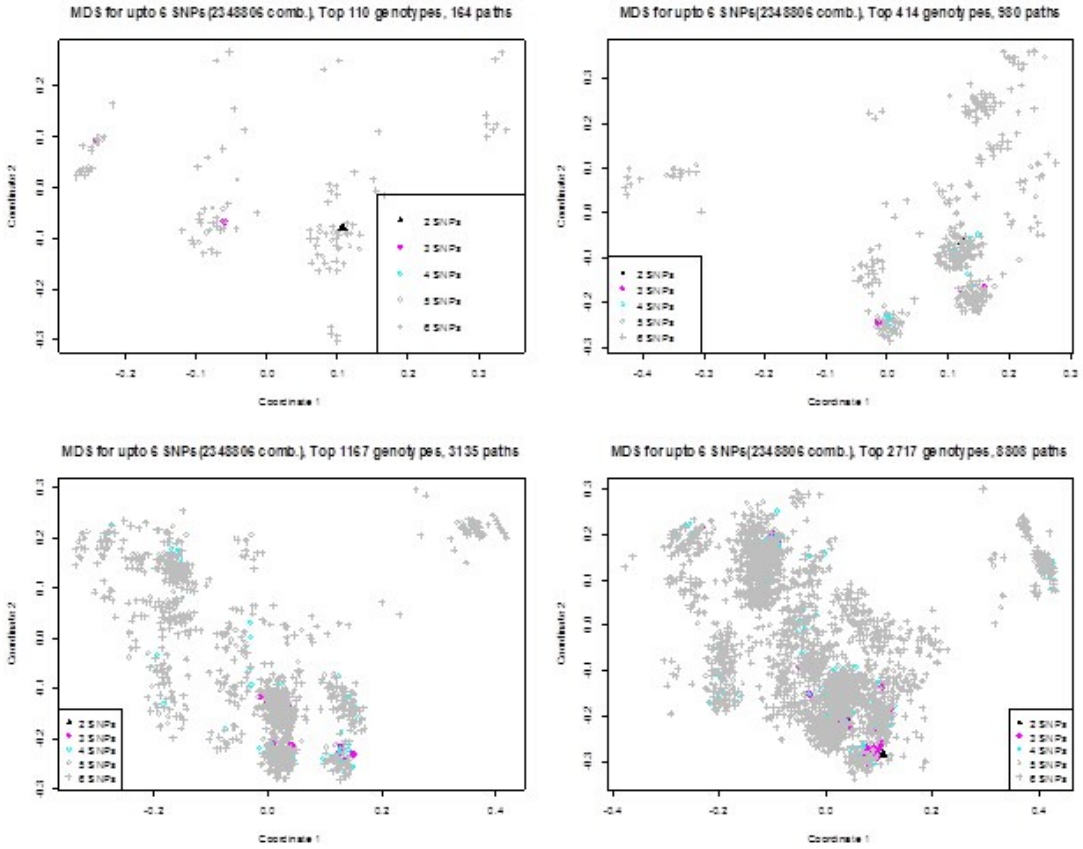

Figure 4.15: MDS Result for Top 3000 Paths of Length 6

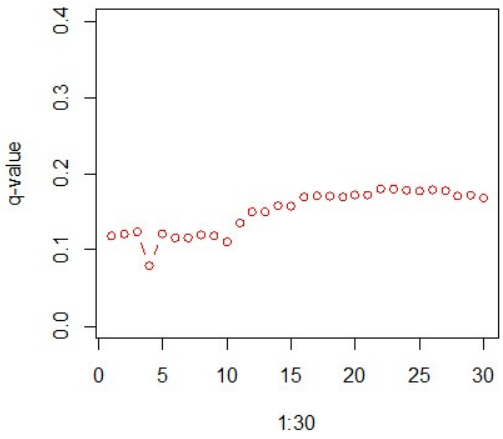

Figure 4.16: Adjusted P-values for Top 30 Paths of Length 4

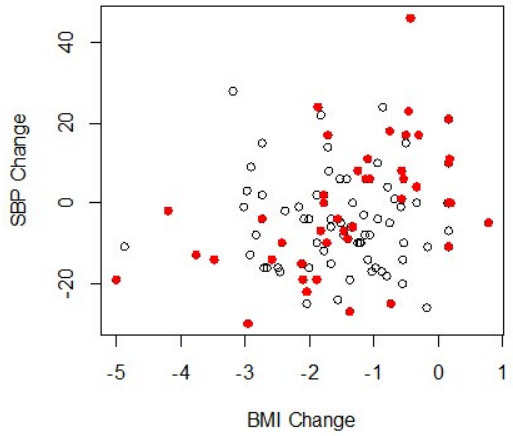

Figure 4.17: Top Significant Set in Paths of Length 4

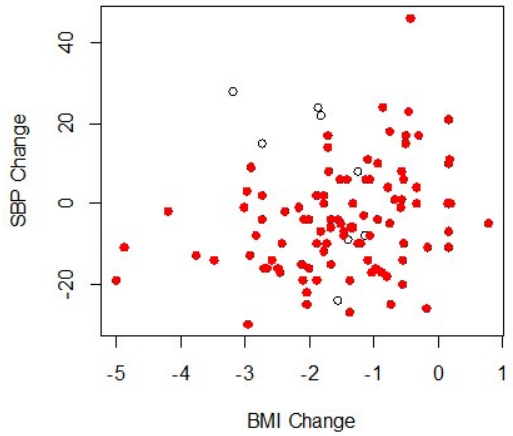

Figure 4.18: The Second Top Significant set in Paths of Length 4

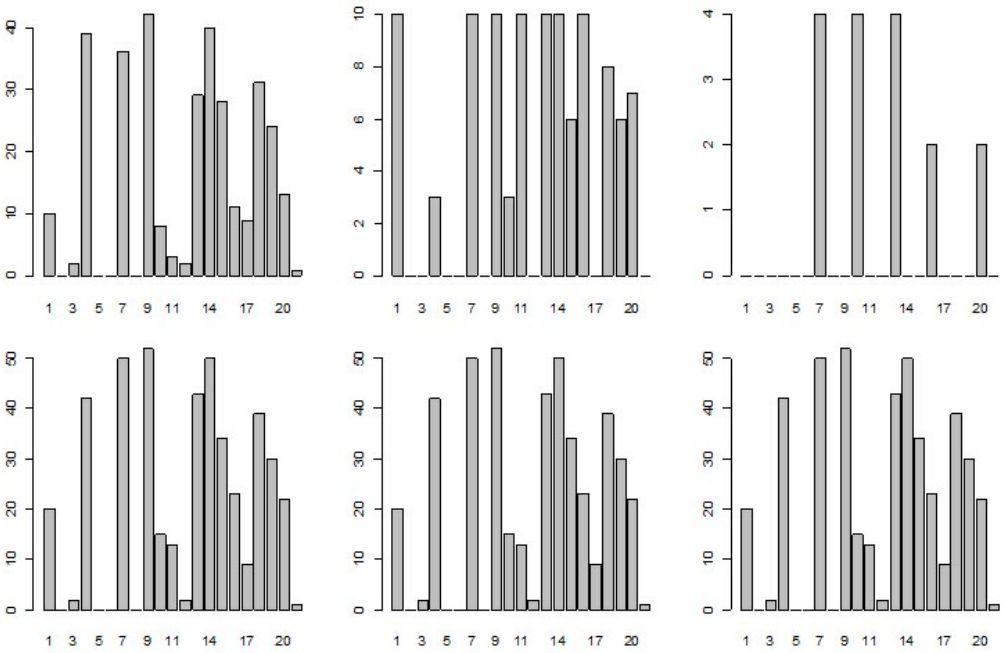

Figure 4.19: Distribution of SNPs in Top 200 Paths of Length 4 by Clusters

References

[Baranzini et al., 2009] Baranzini, S. E., Galwey, N. W., Wang, J., Khankhanian, P., Lindberg, R., Pelletier, D., Wu, W., Uitdehaag, B. M., Kappos, L., Polman, C. H., et al. (2009). Pathway and network-based analysis of genome-wide association studies in multiple sclerosis. *Human molecular genetics*, 18(11):2078–2090.

[Breiman et al., 1984] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

[Burton et al., 2007] Burton, P. R., Clayton, D. G., Cardon, L. R., Craddock, N., Deloukas, P., Duncanson, A., Kwiatkowski, D. P., McCarthy, M. I., Ouwehand, W. H., Samani, N. J., et al. (2007). Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661–678.

[Cantor et al., 2010] Cantor, R. M., Lange, K., and Sinsheimer, J. S. (2010). Prioritizing gwas results: a review of statistical methods and recommendations for their application. *The American Journal of Human Genetics*, 86(1):6–22.

[Chan and Arvey, 2012] Chan, M. E. and Arvey, R. D. (2012). Meta-analysis and the development of knowledge. *Perspectives on Psychological Science*, 7(1):79–92.

[Chou, 1991] Chou, P. A. (1991). Optimal partitioning for classification and regression trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):340–354.

[Evangelou and Ioannidis, 2013] Evangelou, E. and Ioannidis, J. P. (2013). Meta-analysis methods for genome-wide association studies and beyond. *Nature Reviews Genetics*, 14(6):379–389.

[Friedman et al., 2010a] Friedman, J., Hastie, T., and Tibshirani, R. (2010a). A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*.

[Friedman et al., 2010b] Friedman, J., Hastie, T., and Tibshirani, R. (2010b). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

[Friedman et al., 2000] Friedman, J., Hastie, T., Tibshirani, R., et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407.

[Friedman, 2001] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

[Glass, 1976] Glass, G. V. (1976). Primary, secondary, and meta-analysis of research. *Educational researcher*, pages 3–8.

[Goddard and Hayes, 2009] Goddard, M. E. and Hayes, B. J. (2009). Mapping genes for complex traits in domestic animals and their use in breeding programmes. *Nature Reviews Genetics*, 10(6):381–391.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R. (2009). *The elements of statistical learning*, volume 2. Springer.

[Heffner et al., 2009] Heffner, E. L., Sorrells, M. E., and Jannink, J.-L. (2009). Genomic selection for crop improvement. *Crop Science*, 49(1):1–12.

[Jiang et al., 2007] Jiang, P., Wu, H., Wang, W., Ma, W., Sun, X., and Lu, Z. (2007). Mipred: classification of real and pseudo microrna precursors using random forest prediction model with combined features. *Nucleic acids research*, 35(suppl 2):W339–W344.

[Klein et al., 2005] Klein, R. J., Zeiss, C., Chew, E. Y., Tsai, J.-Y., Sackler, R. S., Haynes, C., Henning, A. K., SanGiovanni, J. P., Mane, S. M., Mayne, S. T., et al. (2005). Complement factor h polymorphism in age-related macular degeneration. *Science*, 308(5720):385–389.

[Kohavi et al., 1995] Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, volume 14, pages 1137–1145.

[Kostis et al., 2002] Kostis, J. B., Wilson, A. C., Hooper, W. C., Harrison, K. W., Philipp, C. S., Appel, L. J., Espeland, M. A., Folmar, S., and Johnson, K. C. (2002). Association of angiotensin-converting enzyme dd genotype with blood pressure sensitivity to weight loss. *American heart journal*, 144(4):625–629.

[Kostis et al., 2013] Kostis, W. J., Cabrera, J., Hooper, W. C., Whelton, P. K., Espeland, M. A., Cosgrove, N. M., Cheng, J. Q., Deng, Y., De Staerck, C., Pyle, M., et al. (2013). Relationships between selected gene polymorphisms and blood pressure sensitivity to weight loss in elderly persons with hypertension. *Hypertension*, 61(4):857–863.

[Latham, 2011] Latham, J. (2011). The failure of the genome. *The Guardian*, 17.

[Lewinger et al., 2007] Lewinger, J. P., Conti, D. V., Baurley, J. W., Triche, T. J., and Thomas, D. C. (2007). Hierarchical bayes prioritization of marker associations from a genome-wide association scan for further investigation. *Genetic epidemiology*, 31(8):871–882.

[Li et al., 2011] Li, J., Das, K., Fu, G., Li, R., and Wu, R. (2011). The bayesian lasso for genome-wide association studies. *Bioinformatics*, 27(4):516–523.

[Loh and Shih, 1997] Loh, W.-Y. and Shih, Y.-S. (1997). Split selection methods for classification trees. *Statistica sinica*, 7(4):815–840.

[McCarthy et al., 2008] McCarthy, M. I., Abecasis, G. R., Cardon, L. R., Goldstein, D. B., Little, J., Ioannidis, J. P., and Hirschhorn, J. N. (2008). Genome-wide association studies for complex traits: consensus, uncertainty and challenges. *Nature Reviews Genetics*, 9(5):356–369.

[McClellan and King, 2010] McClellan, J. and King, M.-C. (2010). Genetic heterogeneity in human disease. *Cell*, 141(2):210–217.

[Montgomery et al., 2012] Montgomery, D. C., Peck, E. A., and Vining, G. G. (2012). *Introduction to linear regression analysis*, volume 821. John Wiley & Sons.

[Muehlschlegel et al., 2010] Muehlschlegel, J. D., Liu, K.-Y., Perry, T. E., Fox, A. A., Collard, C. D., Shernan, S. K., Body, S. C., et al. (2010). Chromosome 9p21 variant predicts mortality after coronary artery bypass graft surgery. *Circulation*, 122(11 suppl 1):S60–S65.

[Murthy et al., 1994] Murthy, S. K., Kasif, S., and Salzberg, S. (1994). A system for induction of oblique decision trees. *arXiv preprint cs/9408103*.

[Newcombe et al., 2009] Newcombe, P. J., Verzilli, C., Casas, J. P., Hingorani, A. D., Smeeth, L., and Whittaker, J. C. (2009). Multilocus bayesian meta-analysis of gene-disease associations. *The American Journal of Human Genetics*, 84(5):567–580.

[Quinlan and Cameron-Jones, 1995] Quinlan, J. R. and Cameron-Jones, R. M. (1995). Induction of logic programs: Foil and related systems. *New Generation Computing*, 13(3-4):287–312.

[Ripley, 1996] Ripley, B. D. (1996). *Pattern recognition and neural networks*. Cambridge university press.

[Samani et al., 2007] Samani, N. J., Erdmann, J., Hall, A. S., Hengstenberg, C., Mangino, M., Mayer, B., Dixon, R. J., Meitinger, T., Braund, P., Wichmann, H.-E., et al. (2007). Genomewide association analysis of coronary artery disease. *New England Journal of Medicine*, 357(5):443–453.

[Schmutz et al., 2004] Schmutz, J., Wheeler, J., Grimwood, J., Dickson, M., Yang, J., Caoile, C., Bajorek, E., Black, S., Chan, Y. M., Denys, M., et al. (2004). Quality assessment of the human genome sequence. *Nature*, 429(6990):365–368.

[Sebastiani et al., 2005] Sebastiani, P., Abad, M., and Ramoni, M. F. (2005). Bayesian networks for genomic analysis. *Genomic signal processing and statistics*, pages 281–320.

[Speliotes et al., 2010] Speliotes, E. K., Willer, C. J., Berndt, S. I., Monda, K. L., Thorleifsson, G., Jackson, A. U., Allen, H. L., Lindgren, C. M., Luan, J., Mägi, R., et al. (2010). Association analyses of 249,796 individuals reveal 18 new loci associated with body mass index. *Nature genetics*, 42(11):937–948.

[Stahl et al., 2010] Stahl, E. A., Raychaudhuri, S., Remmers, E. F., Xie, G., Eyre, S., Thomson, B. P., Li, Y., Kurreeman, F. A., Zhernakova, A., Hinks, A., et al. (2010). Genome-wide association study meta-analysis identifies seven new rheumatoid arthritis risk loci. *Nature genetics*, 42(6):508–514.

[Szymczak et al., 2009] Szymczak, S., Biernacka, J. M., Cordell, H. J., González-Recio, O., König, I. R., Zhang, H., and Sun, Y. V. (2009). Machine learning in genome-wide association studies. *Genetic Epidemiology*, 33(S1):S51–S57.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

[Tomita et al., 2004] Tomita, Y., Tomida, S., Hasegawa, Y., Suzuki, Y., Shirakawa, T., Kobayashi, T., and Honda, H. (2004). Artificial neural network approach for selection of susceptible single nucleotide polymorphisms and construction of prediction model on childhood allergic asthma. *BMC bioinformatics*, 5(1):120.

[Verzilli et al., 2008] Verzilli, C., Shah, T., Casas, J. P., Chapman, J., Sandhu, M., Debenham, S. L., Boekholdt, M. S., Khaw, K. T., Wareham, N. J., Judson, R., et al. (2008). Bayesian meta-analysis of genetic association studies with different sets of markers. *The American Journal of Human Genetics*, 82(4):859–872.

[Visscher et al., 2012] Visscher, P. M., Brown, M. A., McCarthy, M. I., and Yang, J. (2012). Five years of gwas discovery. *The American Journal of Human Genetics*, 90(1):7–24.

[Waldmann et al., 2013] Waldmann, P., Mészáros, G., Gredler, B., Fuerst, C., and Sölkner, J. (2013). Evaluation of the lasso and the elastic net in genome-wide association studies. *Frontiers in genetics*, 4.

[Wan et al., 2009] Wan, X., Yang, C., Yang, Q., Xue, H., Tang, N. L., and Yu, W. (2009). Megasnphunter: a learning approach to detect disease predisposition snps and high level interactions in genome wide association study. *BMC bioinformatics*, 10(1):13.

[Whelton et al., 1998] Whelton, P. K., Appel, L. J., Espeland, M. A., Applegate, W. B., Ettinger Jr, W. H., Kostis, J. B., Kumanyika, S., Lacy, C. R., Johnson, K. C., Folmar, S., et al. (1998). Sodium reduction and weight loss in the treatment of hypertension in older persons: a randomized controlled trial of nonpharmacologic interventions in the elderly (tone). *Jama*, 279(11):839–846.

[Wray et al., 2007] Wray, N. R., Goddard, M. E., and Visscher, P. M. (2007). Prediction of individual genetic risk to disease from genome-wide association studies. *Genome research*, 17(10):1520–1528.

[Yang et al., 2011] Yang, J., Lee, S. H., Goddard, M. E., and Visscher, P. M. (2011). Gcta: a tool for genome-wide complex trait analysis. *The American Journal of Human Genetics*, 88(1):76–82.

[Yesupriya et al., 2008] Yesupriya, A., Evangelou, E., Kavvoura, F. K., Patsopoulos, N. A., Clyne, M., Walsh, M. C., Lin, B. K., Yu, W., Gwinn, M., Ioannidis, J. P., et al. (2008). Reporting of human genome epidemiology (huge) association studies: an empirical assessment. *BMC medical research methodology*, 8(1):31.

[Yuan and Lin, 2006] Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

[Zeggini et al., 2008] Zeggini, E., Scott, L. J., Saxena, R., Voight, B. F., Marchini, J. L., Hu, T., de Bakker, P. I., Abecasis, G. R., Almgren, P., Andersen, G., et al. (2008).

Meta-analysis of genome-wide association data and large-scale replication identifies additional susceptibility loci for type 2 diabetes. *Nature genetics*, 40(5):638–645.

[Zou and Hastie, 2005] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.