Canonical Correlation and Clustering for High Dimensional Data

Author: Qing Ouyang

Supervisor: Dr. Angelo J. Canty

McMaster University

A thesis submitted in partial fulfilment of the requirements for the degree of

Master of Science

Abstract

Multi-view datasets arise naturally in statistical genetics when the genetic and trait profile of an individual is portrayed by two feature vectors. A motivating problem concerning the Skin Intrinsic Fluorescence (SIF) study on the Diabetes Control and Complications Trial (DCCT) subjects is presented. A widely applied quantitative method to explore the correlation structure between two domains of a multi-view dataset is the Canonical Correlation Analysis (CCA), which seeks the canonical loading vectors such that the transformed canonical covariates are maximally correlated. In the high dimensional case, regularization of the dataset is required before CCA can be applied. Furthermore, the nature of genetic research suggests that sparse output is more desirable. In this thesis, two regularized CCA (rCCA) methods and a sparse CCA (sCCA) method are presented. When correlation sub-structure exists, stand-alone CCA method will not perform well. To tackle this limitation, a mixture of local CCA models can be employed. In this thesis, I review a correlation clustering algorithm proposed by Fern, Brodley and Friedl (2005), which seeks to group subjects into clusters such that features are identically correlated within each cluster. An evaluation study is performed to assess the effectiveness of CCA and correlation clustering algorithms using artificial multi-view datasets. Both sCCA and sCCA-based correlation clustering exhibited superior performance compare to the rCCA and rCCA-based correlation clustering. The sCCA and the sCCA-clustering are applied to the multi-view dataset consisted of PrediXcan imputed gene expression and SIF measurements of DCCT subjects. The stand-alone sparse CCA method identified 193 among 11538 genes being correlated with SIF#7. Further investigation of these 193 genes with simple linear regression and t-test revealed that only two genes, ENSG00000100281.9 and ENSG00000112787.8, were significant in association with SIF#7. No plausible clustering scheme was detected by the sCCA based correlation clustering method.

For my wife, Zuoyi.

Acknowledgements

I thank Dr. Shui Feng for his valuable guidance throughout of my pursuit of a higher degree in Statistics. I thank Dr. Fred Hoppe, Dr. Sharon McNicholas and Dr. Narayanaswamy Balakrishnan for their teaching which greatly expanded my body of knowledge and deepened my understanding in the statistical theory and methods.

Particularly, I would like to thank Dr. Angelo Canty for supervising my thesis work with my greatest gratitude. His encouragement, thought-provoking questions and suggestions greatly inspired me in my pursuit of an interesting research. I would like to thank my collaborators from the SickKids hospital: Dr. Andrew Paterson, Dr. Delnaz Roshandel and the HPF IT help desk, for the access to research data and computing resource; A special thank you to Dr. Sara Good, who generously provided me the access to the PrediXcan imputed expression dataset resulted from her recent study. I also wish to thank Dr. Roman Viveros-Aguilera and Dr. Joseph Beyene for examining my thesis work. Last but not least, I would like to thank my family for their love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The DCCT/ EDIC study

1.1.1 Diabetic Control and Complication Trials(DCCT)

The Diabetes Control and Complications Trial (DCCT) study was a clinical trial designed to test the hypothesis that "achieving near-normal glucose would ameliorate the long-term complications of diabetes" (DCCT/EDIC Research Group, 2014) and investigate the possibility of delaying or preventing the complications of type 1 diabetes (T1DM) through intensive insulin therapy. The study was conducted over 1441 T1DM patients at 29 clinical centres across North America, from 1982 to 1993. The study consisted of two treatment groups - one treatment group used intensive therapy, which aimed at achieving non-diabetic level of glycemia as safely as possible, and the other group used conventional therapy, which aimed to maintain safe asymptomatic glucose control (DCCT/EDIC Research Group, 2014). There were also two patient cohorts - the primary prevention, consisting of patients without retinopathy symptom, and the secondary intervention, consisting of patients at an early stage of retinopathy. The participants were recruited during 1983 - 1989 under the criteria summarized in Table 1.1, and randomly assigned to either intensive or conventional treatment group upon enrolment. The experimental set-up and descriptive information of the participants are summarized in Table 1.2 by treatment group. Participants in the intensive treatment group received insulin through at least three daily injections or continuous subcutaneous insulin infusion using external pumps guided by self-monitoring of blood glucose. Whereas in the conventional treatment group, participants received only one or two insulin injections daily and there was no self-monitoring of glucose. In the case of glycemia exceeding the pre-set upper bound of 13.5%, the patient was switched to intensive therapy independently of whether any symptom was presented.

Glycated hemoglobin (HbA1c) and blood pressure measurements were taken quarterly from the participants of the conventional treatment group and monthly from the intensive treatment group. Several other measurements were taken for various studies, such as the Density Gradient Ultracentrifugation (DGUC) and the Skin Intrinsic Fluorescence (SIF). Over 99% of participants were studied for on average 6.5 years before termination of the trial. The study exhibited significant reduction in the level of glycated hemoglobin under intensive treatment, resulted a mean HbA1c of 7.2% for intensive treatment compare to the mean HbA1c of 9.1% for conventional treatment as Figure 1.1 shows. This translated to a 35 to 76% reduction in the early stages of micro-vascular complications(DCCT/EDIC Research Group, 2014).

Common Characteristics	
Fasting C-peptide	<0.2nmol/L
History of Cardiovascular disease	No
Hypertension	No
Dyslipidemia	No
Neuropathy	No
Other Severe Diseases	No
Primary Prevention Cohort	
T1D Duration (Years)	1-5
Evidence of retinopathy	None
Albumin excretion rate (AER)	< 40 mg per 24 h
Secondary Intervention Cohort	
T1D Duration (Years)	1-15
Evidence of retinopathy	At least one microaneurysm in either eye
Albumin excretion rate (AER)	< 200 mg per 24 h

Table 1.1: The enrolment criteria for DCCT participants

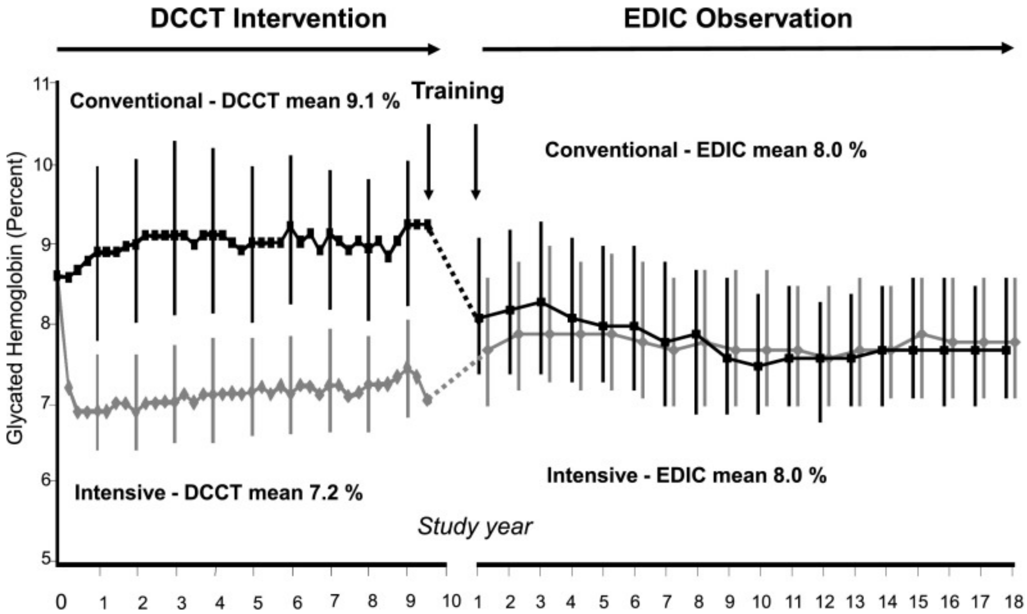

Figure 1.1: The median glycated hemoglobin (HbA1c) level of DCCT and EDIC participants by the original treatment group. The vertical bars refer to the 1st and 3rd quantiles. Source: (DCCT/EDIC Research Group, 2014)

	Intensive	Conventional
Cohort(n)		
Primary Prevention	307	344
Secondary	330	323
Gender(n)		
Male	332	363
Female	305	304
Characteristics of participants		
Age of Enrolment (years)	27.2 ± 7.1	26.5 ± 7.1
Duration of participation (years)	6.3 ± 1.7	6.2 ± 1.6
Eligibility HbA1C (%)	9.07 ± 1.58	9.00 ± 1.61
Mean HbA1C (%)	7.22 ± 0.93	9.06 ± 1.24
Stimulated C-peptide at DCCT baseline (pmol/ml)	0.111 ± 0.119	0.117 ± 0.119

Table 1.2: Experiment set-up and characteristics of total 1304 DCCT participants by treatment group. Source: (Paterson et al., 2009)

1.1.2 Epidemiology of Diabetes Interventions and Complications (EDIC)

Upon observing the improved result in HbA1c level with intensive treatment, the DCCT was prematurely terminated in 1993 as it was no longer ethical to keep the intensive treatment from the other half of the patients. In order to further investigate the durability of the effect of the intensive insulin therapy, the researchers initiated a long term follow up and observational program - Epidemiology of Diabetes Interventions and Complications (EDIC). As an observational study, researchers visited the participants less frequently than in the DCCT, however the same key measurements(HbA1c, complications) were taken from the participants as in the earlier study. The gap in the glycemia level between the two original therapy groups gradually narrowed and eventually disappeared, as shown in Figure 1.1 (right). The long term observational window of EDIC also enabled the researchers to investigate the possible impact of intensive insulin therapy over some more advanced complications. The study demonstrated effectiveness of intensive therapy in preventing several advanced complications including retinopathy, nephropathy, and autonomic manifestations of neuropathy (DCCT/EDIC Research Group, 2014).

1.2 Human Genome, GWAS and PrediXcan

1.2.1 Human Genome

The Human Genome consists of 23 pairs of chromosomes, including 22 pairs of autosomes and 1 pair of sex chromosomes. Each parent contributes to half of an individual's genome content as a result of sexual reproduction process. A chromosome is essentially a Deoxyribonucleic acid (DNA) macro-molecule folded into an extremely condensed form under the effect of package proteins. The collection of DNA molecules carries the entire set of genetic instructions by which human being grow and reproduce. Each DNA molecules is composed of two biopolymer strands, which coil around each other and form a double helix structure. The basic building blocks for DNA are four types of nucleotides. These four types of nucleotides are adenine(A), thymine (T), cytosine (C) or guanine(G). Nucleotides form base-pairs through a hydrogen bond, with A pairs with T and C pairs with G. Genetic instructions are coded by the sequence of nucleotides. A section of DNA which codes for a functional molecule (such as a protein) is called a gene. A gene can influence a specific characteristic of through a complex chain of molecular processes, therefore genetic variation can lead to variation in traits. The most common form of genetic variation involves changes to one single base pair at a given location on the DNA called a single-nucleotide polymorphism (SNP). In recent years a wide range of human disease have been found to be associated with SNPs.

A locus refers to a certain location in the genome and the variants of the DNA sequence at such locus are referred to as alleles. The genotype of a SNP is determined by the combination of two alleles for a diploid organism. Let B and b denote the major allele (the most frequently observed in population) and minor allele (the less frequently observed) respectively, then BB and bb represent homozygous alleles and Bb represents heterozygous alleles. Allele frequency is the percentage of the population that carries a certain allele, which can also be interpreted as the probability of observing that allele in a randomly selected individual. Minor Allele Frequency (MAF) then refers to the allele frequency of the less common allele. We typically denote the genotype with no minor allele 0 (e.g. BB), genotype with one minor allele 1 (e.g. Bb), and the genotype with a pair of minor alleles 2 (e.g. bb). Linkage Disequilibrium (LD) measures the level of association between two SNPs at different loci. SNPs are said to be in LD if the joint distribution of the genotypes is different from the distribution assuming they are independent (e.g. the product of their marginal distribution).

1.2.2 Genome-wide Association Study (GWAS)

A widely used method in statistical genetics is the genome-wide association study (GWAS), which is used to identify SNPs that are associated with a phenotype of interest. SNP data are collected from the subjects and genotyped via some genotype calling algorithm. Poor sample data quality may result in missing values of SNP genotypes, these SNPs may be abandoned or have their genotypes imputed via statistical inference techniques using know haplotypes in a population (typically from a large human genetic study program such as HapMap or the 1000 Genomes Project) (Y. Li, Willer, Sanna, & Abecasis, 2009). A dosage value from 0 to 2 are calculated, where dosage of 0 refers to the genotype bb and 2 refers to the genotype BB. The traits measurements are regressed against each SNP. Test of significance is performed individually on the resulted SNP coefficients and the associated p-values are calculated. Significantly small p-value indicates strong association between the SNPs and the trait of interest. The most commonly used tool to visualize the result is the Manhattan Plot, with SNPs plotted on the horizontal axis and the negative base-10 logarithm of the p-values of the observed odd ratio on the vertical axis. Most of SNPs will have a low profile due to low level of association to the trait, Spikes in the plots will represent the SNPs that are significantly associated to the trait of interest. Multiple comparisons problem can arise as GWAS typically performs a large number of statistical inferences simultaneously, and therefore the p-value threshold for significance needs to be corrected(Miller, 1981). Various techniques for multiple testing correction exist and the most widely used on is perhaps the Bonferroni adjustment, which deems a test score significant only if the corresponding p-value is less than α/n, where α and n refers to the significant threshold and number of separate tests (Johnson et al., 2010; Noble, 2009).

1.2.3 PrediXcan

Expression quantitative loci(eQTLs) are the genomic regions that influence the messenger RNA (mRNA) level which indicates the gene expression level (Rockman & Kruglyak, 2006), and how actively a gene is transcribed influences the abundance of certain types of protein which eventually links to the variation of traits. PrediXcan is a gene-based association method that aims to directly test the molecular mechanisms through which genetic variation affects phenotype (Gamazon et al., 2015). With the built-in gene expression imputation model, the PrediXcan predicts the expression of genes that are regulated by eQTLs (B. Li et al., 2018).

Genomic and transcriptomic data from three different sources were used to develop a parsimonious additive linear model for gene expression - the whole blood RNA-Seq data and genome-wide genotype data for 922 individuals from the Depression Genes and Networks(DGN) cohort, all of European ancestry, were used to generate the model; RNA-Seq data from 421 lymphoblastoid cell lines from the Genetic European Variation in Health and Disease (GEUVADIS) consortium and the Genotype-Tissue Expression (GTEx) RNA-Seq Data across 9 tissues were used for testing the model trained by the DGN data. The gene expression is proposed to be characterized by an additive linear model for the form,

$$Y_g = \sum_k \omega_{k,g} X_k + \epsilon \tag{1.1}$$

where Y_g denotes the expression level of gene g, $\omega_{k,g}$ stands for the effect size for SNP k for the expression level of gene g, X_k denotes the dosage for SNP k in the set of all cis-regulatory SNPs, and ϵ represents environmental factors that influence the gene expression level, therefore the summation $\Sigma_k \omega_{k,g} X_k$ represents the Genetically Regulated Expression (GReX).

The model was trained using LASSO and Elastic Net, the eQTLs that are identified to be in association with the expression traits and their estimated effect sizes

($\hat{\omega}_{k,g}$) are stored in the PredictDB data repository by the GTEx tissue type and are available through (http://predictdb.org/).

To implement the PrediXcan method, we initially impute the genetically regulated expression for each subject with the additive linear model,

$$G\hat{Re}X_g = \sum_k \hat{\omega}_{k,g} X_k \tag{1.2}$$

and then associate the imputed gene expression values with the physiological traits in the same fashion as GWAS, to identify genes whose genetically regulated expression is significantly associated with the traits of interest.

Compared to GWAS which typically require 5-10 million single tests of significance, PrediXcan features a much smaller multiple testing burden and usually only requires roughly 10 thousands tests. PrediXcan also utilizes the relatively more accessible SNPs data to impute the gene expression for a gene-based association study, with no actual transcriptome data required, making this method widely applicable to many existing studies with SNP genotype datasets.

1.3 Motivating Problem - the Skin Intrinsic Fluorescence data

Advanced glycation end products (AGEs) are the end result of a complex chain of biochemical process under the condition of accelerated glycation due to hyperglycemia, and are known to be risk factor for micro-vascular and macro-vascular diabetes complications. Given the fluorescent nature of some AGEs, non-invasive means such as optical spectroscopy can be applied to measure the accumulated level of AGEs in the skin. Compare to the traditional skin biopsy method, this greatly promotes the feasibility of large scale study of the association between genetic variation and AGEs. Such optical spectroscopy devices emit light at multiple wave length (visible and near-ultraviolet) to illuminate the subject's left forearm skin. The induced skin fluorescence reflectance is captured by a specially designed fiber-optic probe and relayed to a spectrograph (Hull et al., 2014). The skin AGEs level can be characterized by 15 measurements of skin fluorescence reflectance, ordered by the excitation wavelength and emission range, this can be considered as a 15-dimensional feature vector. Previous studies had revealed the association between markers near the N-acetyltransferase 2 (NAT2) gene and skin fluorescence traits (Eny et al., 2014). Roshandel et al. (2016), performed a meta-GWAS study involving 1359 patients from the Diabetes and Complications Control Trail and 278 patients from the Wisconsin Epidemiologic study of Diabetic Retinopathy to identify additional genetic loci that are associated to the skin fluorescence traits in type I diabetes. Beside the known locus of NAT2, a new locus on chromosome 1 was found to be significantly associated with the SF in T1D patients, and such association was not observed for non-diabetic patients (Roshandel et al., 2016).

1.4 Canonical Correlation Analysis

1.4.1 Basics of Canonical Correlation Analysis

Many genetic studies, such as the earlier described skin intrinsic fluorescence study, generate so called "multi-view data", where the subjects are portrayed by two feature vectors, each feature vector consists of a set of variables. Researchers are often interested in studying the correlation structures between the two domains of variables. For example, we may want to study the correlation between an individual's gene expression profile (approximately 10,000 variables) and the skin fluorescence measurements (15 variables). An useful analytical approach to such multi-view data is Canonical Correlation Analysis (CCA). The purpose of canonical correlation analysis is to identify and quantify the correlation structure between two sets of random variables (Fern, Brodley, & Friedl, 2005).

Let us consider a multi-view data in which subjects are described by two feature vectors, $X = (x_1, x_2, ...x_p)^T$ and $Y = (y_1, y_2, ...y_q)^T$. Mathematically, CCA seeks the transformations a and b, respectively to X and Y, such that the linear correlation between the two transformed quantities $u = a^T X$ and $v = b^T Y$ (called canonical variables) is maximized (Hotelling, 1936). That is,

$$(a^*, b^*) = \underset{a,b}{\operatorname{argmax}}\, Corr(u, v) \tag{1.3}$$

Similar to principal component analysis, we denote the u and v found as above u_1 and v_1 and name it the first pair of canonical variables. If we repeat this process subject to the constraint that the newly found canonical variables are uncorrelated with u_1, v_1, then we obtain the second pair of canonical variables, u_2 and v_2. We may continue this procedure up to $d = \min(p, q)$ times and acquire up to d-th pair of canonical variables. Let r_k denote the correlation between the the k-th pair of canonical variables, this algorithms yields canonical variables with decreasing correlations, that is $r_k > r_{k+1}$, for $k = 1, \ldots, d-1$.

Computationally, this optimization problem can be solved by finding the eigenvalue and eigenvectors of two matrices M_x and M_y. Let Σ_{xy} be the covariance matrix with the (i, j)-th entry $\sigma_{x_i y_j}$, where $i = 1, \ldots, p$ and $j = 1, \ldots, q$ and let

$$M_x = \Sigma_{xx}^{-1} \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx}$$

and

$$M_y = \Sigma_{yy}^{-1} \Sigma_{yx} \Sigma_{xx}^{-1} \Sigma_{xy}$$

The eigenvalues of M_x and M_y are identical and in fact the k-th eigenvalue equals to the square of the k-th canonical correlation, that is $\lambda_k = r_k^2$. If we arrange the eigenvalues in decreasing order, the corresponding k-th eigenvectors of M_x and M_y are the transformation vectors a_k and b_k. Let $\boldsymbol{M}$ denote a canonical correlation model (Assume we only utilize the first K order canonical variables),

$$\boldsymbol{M} = \{(u_k, v_k), r_k, (a_k, b_k), k = 1, \ldots, K\}$$

where (u_k, v_k) and r_k are the k-th pair of canonical variables and their corresponding correlation coefficient, and (a_k, b_k) stands for the corresponding transformation vector. We refer to $\boldsymbol{M}$ as a CCA model (Fern et al., 2005). Because the correlation rapidly becomes weaker as k increases (that is, as k increases, the canonical variable pairs contains less and less useful information), in most real world application, it is sufficient for us to only consider the first 1-3 pairs of canonical variables. In this study, only the first order canonical variable will be used.

1.4.2 Regularized Canonical Correlation Analysis

Special treatment is required when the CCA is implemented over high dimensional data, where the number of feature variables greatly exceeds the number of observations. The standard CCA we described earlier cannot be effectively performed due to ill-conditioned variance-covariance matrices that arise in the high dimensional setting. In such cases, the resulted canonical correlation will always be close to 1 and not actually provide any meaningful information(González, Déjean, Martin, Baccini, et al., 2008). One way to tackle this issue is to include a data regularization step prior to implementation of the standard CCA.

A cross validation based regularization approach was firstly proposed by Vinod (1976) and further developed by Leurgans, Moyeed, and Silverman (1993). A pair of tuning parameters $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$ is introduced to replace the original covariance matrices Σ_{XX} and Σ_{YY} by

$$S_{XX}(\lambda_1) = \Sigma_{XX} + \lambda_1 I_p$$

and

$$S_{YY}(\lambda_2) = \Sigma_{YY} + \lambda_2 I_q$$

where I_p and I_q are diagonal matrices of dimension $p \times p$ and $q \times q$ respectively.

Define $\rho_\lambda^{(-i)}$ be the first the order canonical correlation of CCA, having the i-th observation removed, and $(a_\lambda^{(-i)}, b_\lambda^{(-i)})$ be the corresponding projection vector associated with the first order canonical covariates. We carry out this calculation for all subjects in a leave-one-out cross validation manner and obtain n pair of projection vectors $\{(a_\lambda^{(-i)}, b_\lambda^{(-i)})\}_{i=1}^n$. Define the leave-one-out cross validation score (CV-score) as (Leurgans et al., 1993),

$$CV(\boldsymbol{\lambda}) = \mathrm{Corr}(\{X_i a_\lambda^{(-i)}\}_{i=1}^n, \{Y_i b_\lambda^{(-i)}\}_{i=1}^n) \tag{1.4}$$

A good $\boldsymbol{\lambda}$ would be the one that maximize the leave-one-out cross validation score, that is,

$$\boldsymbol{\lambda^*} = (\lambda_1^*, \lambda_2^*) = \underset{(\lambda_1,\lambda_2)}{\mathrm{argmax}}\, CV(\lambda_1, \lambda_2) \tag{1.5}$$

Finding the best $\boldsymbol{\lambda}$ becomes an optimization problem on the $\boldsymbol{R^2}$. A strategic approach to perform this optimization would be constructing a"grid of points" over the region of "reasonable" values for the $\boldsymbol{\lambda}$, and evaluate the CV-score at each grid point and simply pick the $\boldsymbol{\lambda}$ corresponding to the maximized CV-score (Friedman, 1989; González et al., 2008; Guo, Hastie, & Tibshirani, 2006). Such region of search depends on the experience of user, in the absence of prior knowledge, it is recommended that one may apply this optimization process recursively to approach the optimal $\boldsymbol{\lambda}$ - first construct the searching grids over $[0, 1] \times [0, 1]$ and then locate the region where the optimal $\boldsymbol{\lambda}$ may be reached and further construct searching grids over such region (González et al., 2008). However a significant drawback of this cross-validation regularization is the associated computing cost, when the dimension of the dataset rises the required computing time increases dramatically.

Schäfer and Strimmer (2005), proposed a analytical and computationally more efficient approach of estimating the covariance matrix in the high dimensional setting based on the principle of shrinkage estimation and the Ledoit-Wolf Lemma. Consider a dataset of p variables and sample size n, the empirical covariance matrix S is a $p \times p$ matrix with entries

$$s_{i,j} = \frac{1}{n-1}\sum_{k=1}^{n}(x_{k,i} - \bar{x}_i)(x_{k,j} - \bar{x}_j)$$

for $i = 1, \ldots, p$ and $j = 1, \ldots, p$, where $\bar{x}_i = \frac{1}{n}\sum_{k=1}^{n} x_{k,i}$. However this empirical covariance matrix is ill-suited to the high dimensional case and tend to perform very poorly in estimating the true covariance matrix Σ. Schäfer and Strimmer (2005), proposed to replace the empirical covariance matrix with a shrinkage estimator. Let $\tilde{S}$ denote the shrinkage estimator, construct a convex combination of S and a target matrix T such that

$$\tilde{S} = \delta S + (1 - \delta)T \tag{1.6}$$

where δ is the shrinkage parameter in the range 0 to 1 and T is a diagonal matrix with entries $t_{i,j} = s_{ii}$(the diagonal entry of the empirical covariance matrix S) if $i = j$ and 0 otherwise. Define a risk function $R(\delta)$,

$$R(\delta) = E[\sum_{i=1}^{p}(\tilde{S}_i - \Sigma_i)^2] \tag{1.7}$$

and δ is chosen such that the risk function $R(\delta)$ is minimized, that is,

$$\delta^* = \min_{(0,1)} R(\delta) \tag{1.8}$$

Instead of carrying out the optimization through computationally expensive procedures such as Cross-validation, Schäfer and Strimmer (2005) pointed out the optimal shrinkage parameter δ can be achieve analytically by employing a lemma derived by Ledoit and Wolf (2003). Assume the existence of the first two moments of S and T, Equation (1.7) can be expanded and re-written as (Schäfer & Strimmer, 2005)

$$R(\delta) = \sum_{i=1}^{p} \delta^2 \operatorname{Var}(T_i + (1-\delta)\operatorname{Var}(S_i) + 2\delta(1-\delta)\operatorname{Cov}(T_i, S_i) + [\delta E(T_i - S_i) + \operatorname{Bias}(S_i)]^2 \tag{1.9}$$

Through some tedious algebraic calculation after applying the result of Ledoit and Wolf (2003), the optimal δ is obtained as

$$\delta^* = \frac{\sum_{i=1}^{p} \operatorname{Var}(S_i) - \operatorname{Cov}(T_i, S_i) - \operatorname{Bias}(S_i)E(T_i - S_i)}{\sum_{i=1}^{p} E[(T_i - S_i)^2]} \tag{1.10}$$

Replace all the expectation, variance and covariance in Equation (1.10) with the sample estimates, this yields

$$\hat{\delta}^* = \frac{\sum_{i=1}^{p} \hat{\mathrm{Var}}(S_i) - \hat{\mathrm{Cov}}(T_i, S_i) - \hat{\mathrm{Bias}}(S_i)(T_i - S_i)}{\sum_{i=1}^{p}(T_i - S_i)^2} \tag{1.11}$$

Applying Equation (1.10) to our optimization problem leads to the following expression,

$$\hat{\delta}^* = \frac{\sum_{i \neq j} \hat{\mathrm{Var}}(r_{ij})}{\sum_{i \neq j} r_{ij}^2} \tag{1.12}$$

where r_{ij} is the empirical correlation coefficient(e.g. $r_{ij} = \frac{s_{ij}}{\sqrt{s_{ii}s_{jj}}}$, $i, j = 1, \ldots, p$).

Compare to the cross-validation based regularization described earlier, this approach is significantly less computationally expensive as it requires simple algebraic calculation to the sample correlation coefficients. The efficacy and computational cost of both methods will be evaluated in chapter 2.

Both standard canonical correlation analysis and the cross-validation based regularization can be carried out by the R package `CCA` (González & Djean, 2012). Another R-package `mixOmics`, developed by mixOmics project team allows the implementation of the alternative regularization through shrinkage (Rohart, Gautier, Singh, & LeCao, 2017).

1.4.3 Sparse Canonical Correlation Analysis

The regularized CCA solves the ill-conditioned covariance matrix problem when applying classical CCA to the high dimensional dataset. However, in the real-world genetic research, it is common that within a huge number of genetic variables, only a very tiny subset of them are actually associated with the phenotypes of interest, while all the other variables constitute the background noise. For a study intending to examine the correlation structure between the genetic and trait domain, a sparse representation of the canonical loadings for both domains would provide improved model interpretability.

Witten, Tibshirani, and Hastie (2009), presented a sparse CCA method using some optimization algorithm that is part of a technique they named Penalized Matrix Decomposition (PMD).

Let X denote the $n \times p$ data matrix of View 1 from a multi-view dataset, and Y the $n \times q$ matrix for the View 2, then the classical canonical correlation analysis seeks the canonical vectors u and v such that the correlation between the canonical variates

Xu and Yv are maximized (Hotelling, 1936), and algebraically, this is equivalent to the following optimization problem (Witten et al., 2009),

$$\text{maximize}_{u,v} u^T X^T Y v \quad \text{having } u^T X_T X u \leq 1 \textit{ and } v^T Y_T Y v \leq 1 \tag{1.13}$$

Witten et al. (2009), proposed to introduce sparsity to the output by imposing L_1 penalty on the u and v, that is,

$$\begin{aligned} \text{maximize}_{u,v} u^T X^T Y v \quad \text{having } u^T X_T X u \leq 1, \ \ v^T Y_T Y v \leq 1 \\ P_1(u) \leq c_1 \text{ and } P_2(v) \leq c_2 \end{aligned} \tag{1.14}$$

where $P(\cdot)$ denotes the L_1 penalty constraint, and c_1 and c_2 refer to the bounds of penalty for u and v respectively. Some researches had shown that treating the variance-covariance matrix as diagonal can potentially produce satisfactory result in high dimensional case (Dudoit, Fridlyand, & Speed, 2002; Tibshirani, Hastie, Narasimhan, & Chu, 2003). Replacing the the $X_T X$ and $Y_T Y$ in (1.14) with Identity matrices yields the following form,

$$\begin{aligned} \text{maximize}_{u,v} u^T X^T Y v \quad \text{having } ||u||_2^2 \leq 1, \ \ ||v||_2^2 \leq 1 \\ P_1(u) \leq c_1 \text{ and } P_2(v) \leq c_2 \end{aligned} \tag{1.15}$$

To solve this optimization problem, Witten et al. (2009) proposed the following algorithm (Algorithm 1),

Computation of the first order canonical vectors
1. Initialize v for $\|\|v\|\|_2 = 1$
2. Until convergence, Do:
(a) $u \leftarrow \text{argmax}_u \, u^T X^T Y v$ having $P_1(u) \leq c_1, \ \|\|u\|\|_2^2 \leq 1$
(b) $v \leftarrow \text{argmax}_v \, u^T X^T Y v$ having $P_2(v) \leq c_2, \ \|\|v\|\|_2^2 \leq 1$
3. $d \leftarrow u^T X^T Y v$

Table 1.3: Algorithm 1: Computation of first order canonical vectors, source (Witten et al., 2009)

In practice, Witten et al. (2009) suggest using the first right singular vector of $X^T Y$ as the initial value of v in step 1. The output u, v are the first order canonical vectors and d is the first order canonical correlation. To compute multiple order of canonical vectors, the following algorithm 2 was proposed, which involves repeatedly implementing algorithm 1,

Computation of K orders of canonical vectors
1. Let $X^TY^{(1)} \leftarrow X^TY$.
2. For $k = 1, ..., K$
(a) Find u_k, v_k and d_k of $X^TY^{(k)}$ using Algorithm 1
(b) $X^TY^{(k+1)} \leftarrow X^TY^{(k)} - d_k u_k v_k^T$

Table 1.4: Algorithm 2: Computation of K orders of canonical vectors, source (Witten et al., 2009)

where $K = \min(p, q)$, the output u_k, v_k refer to the kth order canonical vector and d_k is the kth order canonical correlation.

Witten et al. (2009) tested this sparse CCA method on both simulated data and real genomic data. In both cases, the proposed method demonstrated its capability to successfully impose sparsity on the model output and identify the true sparse factors that are in correlation.

1.5 Correlation Clustering

Canonical correlation analysis is designed to detect the global linear correlation between two domains of a dataset and one can expect it to perform poorly if some type of correlation sub-structure exists in the dataset. For example, if the subjects of a genetic study can be sub-divided into groups by the correlation behaviour between certain genetic variables and some phenotype measurement. Then a straight forward application of CCA will not yield meaningful output. One way to tackle this limitation is by incorporating a mixture of local linear correlation models such that each local model captures the local linear correlation structure within the whole dataset.

Fern et al. (2005), proposed a canonical correlation analysis based correlation clustering algorithm, intended to simultaneously group subjects in to clusters according to their local correlation structure in the dataset, such that within each cluster, two domains of the multi-view dataset are identically linear correlated, and each cluster is portrayed by a local CCA model. The correlation algorithm was proposed based on the following two intuitions:

- Intuition 1. If sub-correlation structure exists across the multi-view data matrix, the linear correlation generated by canonical correlation analysis on such dataset is expected to be very weak.

- Intuition 2. For a set of instances, one should be able to predict one canonical variate from another (within the pair of canonical variates of the same order) using a simple linear regression model, if strong linear correlation exists across the two domains.

Based on these two intuitions, Fern et al. (2005) proposed a K-means style clustering algorithm. The core idea is to initialize the algorithm by randomly assigning each subject to one of the k (pre-determined) clusters. Within each iteration, canonical correlation analysis is separately applied to each of the k clusters of subjects to generate a local CCA model. Then for each subject in the sample, a "correlation" based distance between the subject and the an existing cluster is calculated and the subject is assigned to the cluster corresponding to the least distance. It is hoped that through this iteration process a k-group clustering scheme would be formed such that within each cluster the variables across the two domain correlates in the same way (Fern et al., 2005). A full description of the algorithm is presented in Table 1.5.

Model Input
- A multi-view dataset of n subjects with two domains, each domain described by a feature vector $\vec{x}$ or $\vec{y}$ - k, a pre-determined number of clusters - d, the number of canonical variables used by local CCA model
Model Output
- k clusters of grouped subjects - k local CCA models, one for each cluster
Clustering Algorithm
1. **Initialization** Randomly assigning each subject to k clusters 2. **Local CCA model** Apply canonical correlation analysis to each cluster i, and contruct a local CCA model $M_i = \{(u_j, v_j), r_j, (a_j, b_j), j = 1...d\}$, $\forall i = 1...k$, where (u_j, v_j) are the j-th pair of canonical variables, r_j the correlation coefficient between the j-th canonical variables and (a_j, b_j) the corresponding projection vector 3. **Reassignment** - for each local CCA mdoel M_i for the cluster i, construct a family of d linear regression models $\hat{v_{j,i}} = \beta_{j,i} u_{j,i} + \alpha_{j,i}$ for $j = 1, \ldots, d$ - for each subject in the cluster i, compute its canonical variables $u_{j,i}$ and $v_{j,i}$ using local model M_i, and the estimation $\hat{v_{j,i}}$ using the regression model described above, for $j = 1, \ldots, d$ - compute the weighted error for cluster i as $err_i = \Sigma_{j=1}^{d} \frac{r_{j,i}}{r_1} (v_{j,i} - \hat{v_{j,i}})^2$ - Reassign the subject to the cluster with the minimal err^i 4. **Output** Return the current clusters and CCA models if no re-assignment occurs or reaching the maximum iteration. Otherwise, repeat step 2.

Table 1.5: The CCA correlation clustering algorithm

Like regular k-means clustering, the proposed method is essentially a greedy algorithm, which means the final output is initial condition dependent and the iteration can potentially become stuck with some local optimal solution. One way to tackle this issue and improve the accuracy of the algorithm is by repeating the process multiple time with different initializations and compare the outputs at different trials. However unlike the usual k-means clustering based on traditional distance metrics, the proposed correlation clustering algorithms do not guarantee convergence, furthermore, the new clusters resulting from each iteration are not guaranteed to have stronger local linear correlation than before(Fern et al., 2005). The test result from Fern et al. (2005) suggests that the objective function (prediction error) typically rapidly decreases before it begins to oscillate within some relatively narrow range, and based on their experience they recommend setting a maximum number of iterations of 200.

Another important consideration for the practical application of this algorithm is number of clusters to be used. The best guidance would come from our prior knowledge about the studied subjects (such as case-control set-up or clustering naturally existing in the population e.g. ethnicity, gender, presence of a certain disease). In the absence of prior knowledge, there are various computational techniques (such as gap statistics and cluster ensembles) that can aid in the selection of k, however these techniques are purely numerical, the reasonableness of their outputs needs to be examined with caution. The algorithms also require the user to specify the number of pairs of canonical covariates d to be included in the local canonical correlation model. In this thesis we set d to be 1.

The proposed clustering algorithms was tested on a simple artificial dataset in order to examine its efficacy. The testing data was a mixture of two equal-sized datasets each with a distinct correlation pattern, for a total of 2000 subjects. The experiment demonstrated that the proposed method was able to successfully form a partition over the artificial dataset based on the correlation sub-structure and recover the original local linear correlation structure by their design. The proposed algorithm performed consistently well on the artificial dataset, on average only 2.5% of the 2000 subjects were assigned to the wrong cluster (Fern et al., 2005). However higher level of instability of the algorithm was observed when it was applied to a real-world earth science data which naturally has greater complexity in terms of the underlying correlation structure. Nevertheless, in the application to the earth science dataset, the proposed algorithm was still able to identify interesting patterns in the data that the traditional CCA was incapable of finding (Fern et al., 2005).

1.6 Review of Related Work

Existing studies concerning both genetic variation and complex traits are primarily GWAS based. In an earlier study, the N-acetyltransferase 2 (NAT2) was the only locus known to be associated with the SIF (Eny et al., 2014). In Roshandel et al. (2016), a meta-GWAS study was performed over 1359 subjects from DCCT/EDIC and 278 subjects from the Wisconsin Epidemiologic Study of Diabetic Retinopathy(WESDR) with the aim of identifying additional genetic loci influencing skin fluorescence in type 1 diabetes. A new locus, rs7533564 on Chromosome 1 was found to be significantly associated with the SF in the type 1 diabetes patients, and such association was not observed for Non-Diabetic subjects (Roshandel et al., 2016).

Waaijenborg, de Witt Hamer, and Zwinderman (2008) applied a penalized canonical correlation analysis to DNA-markers (e.g.polymorphisms, gene copy numbers) and gene expression data with the aim to investigate the inter-domain correlation structure and to identify groups of co-expressed and co-regulated genes. They adapted elastic net to the conventional canonical correlation analysis to address the issues raised by high dimension data and to improve the interpretability of the output. The hybrid method was demonstrated to work over the high dimension data. Parkhomenko, Tritchler, and Beyene (2007, 2009) independently developed an sCCA algorithm that works very similar to the sCCA by Witten et al. (2009).

Very few direct application of canonical correlation analysis to genetic studies were found, possibly due to the high dimensional nature of the genetic datasets. There are a number of applications of sCCA in the genetic researches. Subramanian, Chidester, Ma, and Do (2018), applied both CCA and sparse CCA to examine the correlation structure between cellular feature imagings and gene expression data of 615 breast cancer samples from The Cancer Genome Atlas (TCGA) program (https://cancergenome.nih.gov/), and were able to uncover significant correlation of several cellular image features with expression of PAM50 genes. Chi et al. (2013), extended the sCCA model to account for correlation structure in both datasets and applied their method to a simulation study to investigate the correlation between genetic variants and phenotypic variations in brain function and structure. Witten and Tibshirani (2009), further extended the sparse CCA method in Witten et al. (2009) in two ways - a sparse supervised CCA was developed by incorporating experiment outcome measurement and a sparse multiple CCA was proposed that allows performing sparse CCA and simultaneous integrative analysis over datasets with more than two domains. Chen, Han, and Carbonell (2012), extended the sparse CCA method in Witten et al. (2009) via a “structured-sparsity-inducing penalty” , a technique of introducing sparsity incorporating the group structural prior knowledge, in order to study the correlation between genetic variation and expression traits in yeast cells.

Clustering algorithms had been widely applied to genetic studies, especially to gene expression data, however the use of clustering methods has been primarily limited to performing data visualization and generating hypotheses about the relationships between genes (Ben-Dor, Shamir, & Yakhini, 1999; D’haeseleer, 2005; Eisen, Spellman, Brown, & Botstein, 1998; Herrero, Valencia, & Dopazo, 2001; Jiang, Tang, & Zhang, 2004; Yeung & Ruzzo, 2001). My focus in this thesis is different - I look to investigate the correlation structure between imputed gene expression and a multivariate phenotype, and identify potential sub-correlation structures in the dataset. This requires clustering algorithms that group subjects based on a correlation-based distance rather than the traditional distance metrics, hence we focused on the correlation clustering algorithm proposed by Fern et al. (2005). Several related works have been found. Lei, Miller, and Dubrawski (2017), proposed a correlation clustering algorithms named Canonical Least Square (CLS) clustering method. Similar to the clustering algorithm by Fern et al. (2005), the CLS clustering constructs local CCA models on the interim clusters, however the CLS clustering re-assigns the subjects based on the Euclidean distance between the subjects and each interim cluster instead of the squared error on predicted canonical covariates as in the CCA clustering. Sun, Lu, Xu, and Bi (2015), developed a multi-view sparse Co-Clustering algorithm via proximal alternating linearized minimization (PALM) which co-clusters row features and column features simultaneously through decomposing multi-view data matrices into product of sparse rows and columns. However, to my understanding this clustering method does not concern the correlation structure of the multi-view dataset therefore it is not best suited to our research purpose. The CCA correlation clustering algorithm, along with the two related works, were proposed and developed concerning only regular multi-view data, that is where sample size exceeds the dimension of feature vector. We were not able to find any evaluation studies of this framework in the high dimensional case or any literature assuring its efficacy when applied to the high dimensional data. We found a number of applications of the correlation clustering method to earth science data, however up this point, we were not able to find any literature concerning the application of this method on any genetic study.

1.7 Rationale and Objectives of the Thesis

1.7.1 Rationale of the thesis

Existing studies have shown association between genetic variation and skin intrinsic fluorescence measures. These studies are typically GWAS-based, which rely on single variant test of association for each SNP across the entire genome to identify loci showing significant association with the phenotypes of interest. PrediXcan allows us to impute the locally genetically regulated expression via parameters stored in the PredictDB and test the association between the genetic profile to phenotype at the gene level, this greatly reduces the computational cost(approximately 10,000 genes vs. approximately 5-10 million SNPs) and can be done without actual transcriptome data, which are often unavailable as the gene expression id cell-type dependent and acquiring such data usually require invasive procedure. In this thesis I propose an integrative approach to examine the skin intrinsic fluorescence data using canonical correlation analysis and correlation clustering. Treating the imputed gene expression and the SIFs measures as a multi-view dataset, through canonical correlation analysis I will aim to examine the correlation structure between gene expression and SIFs measures. With the CCA-based correlation clustering algorithm, I will investigate whether sub-structure exists across the domains.

1.7.2 Objectives of the thesis

This thesis consists of the following three research objectives,

- **Objective 1**. An artificial multi-view dataset will be generated by a design intended to capture the characteristics of the real-world imputed gene expression - SIF data and the association between the true sparse genes and phenotype of interest. Three canonical correlation methods - regularized CCA via shrinkage, regularized CCA via Cross-Validation and Sparse CCA, will be applied to the artificial data and their efficacy and performances will be evaluated.

- **Objective 2**. A multi-view dataset with two intrinsic clusters will be created, where each cluster has distinct gene-trait association, assembling an artificial dataset with correlation sub-structure. A high dimensional version of the correlation clustering algorithms proposed by Fern et al. (2005) will be constructed and applied to this two-cluster simulated data. The efficacy and performance of the correlation clustering algorithms in the high dimensional realm will be tested and evaluated.

- **Objective 3**. Canonical correlation analysis will be applied to the multi-view data combining the imputed gene expression via PrediXcan and SIF measures on the DCCT subjects. Correlation structure between the genetic and trait domain will be examined. The correlation clustering algorithm will be applied to the same dataset to investigate the existence of potential correlation sub-structure between the two domains.

Chapter 2

Evaluation of Canonical Correlation Analysis Methods

A series of evaluation studies were developed to test and compare the performances of the various canonical correlation analysis methods described earlier. Section 2.1 discusses the core design of the artificial test dataset and our experiment. In Section 2.2, regularized CCA methods (via shrinkage or cross-validation) and the sparse CCA were separately applied to the artificial data. In Section 2.3, the performances of these methods are evaluated and compared and the implications to the real-world application were discussed. All methods and experiments were performed using `R` software Version 3.4.3, the corresponding code scripts are in Appendix B.

2.1 Artificial Dataset and Design of Experiment

2.1.1 The core design of artificial test dataset

An test dataset was generated in an attempt to reflect the dimensional characteristics and the correlation structure (both inter-domain and intra-domain) of the real world DCCT-SIF multi-view dataset.

Consider a multi-view dataset with View 1 containing the subject gene expression profiles, and View 2 for the corresponding skin intrinsic fluorescence measurements. Let n denotes the sample size, p and q denote the dimension of the feature vector X of View 1 and Y of View 2, hence the View 1 and View 2 are matrices of size $n \times p$ and $n \times q$, respectively.

Assume the feature vectors X and Y follow multivariate normal distributions and for the core design of the artificial data, we assume that there is no correlation within the genetic nor the trait domain.

To create the artificial dataset, matrices consisting of the "background noise" were first generated for View 1 and View 2, then certain variables in each view were selected and designated as the "interactive variables", finally a mapping between the interactive variables across the two domains was introduced.

Hence the View 1 matrix for the genetic variables is generated by a multivariate normal generator with mean vector

$$\boldsymbol{\mu_1} = \begin{pmatrix} \mu_{1,1} \\ \mu_{2,1} \\ \vdots \\ \mu_{p,1} \end{pmatrix}$$

and a tridiagonal covariance matrix

$$\Sigma_1 = \begin{pmatrix} \sigma_{11,1} & & & 0 \\ & \sigma_{22,1} & & \\ & & \ddots & \\ 0 & & & \sigma_{pp,1} \end{pmatrix}$$

Similarly, the View 2 matrix for the trait variables is generated by a multivariate normal generator with mean vector

$$\boldsymbol{\mu_2} = \begin{pmatrix} \mu_{1,2} \\ \mu_{2,2} \\ \vdots \\ \mu_{q,2} \end{pmatrix}$$

and a tridiagonal covariance matrix

$$\Sigma_2 = \begin{pmatrix} \sigma_{11,2} & & & 0 \\ & \sigma_{22,2} & & \\ & & \ddots & \\ 0 & & & \sigma_{qq,2} \end{pmatrix}$$

In most genetic studies, only a very small collection of genes (even just one or two) are truly associated with the phenotype of interest. We tried to reflect this important characteristic in our simulated data by introducing a linear model depicting the gene-trait association mechanism. To achieve this, n_g genes were chosen from View 1 and designated as the "target genes", and n_t traits were chosen from View 2 as the "influenced traits". We portray the gene-trait association via the following linear mapping,

$$y_{ij} = \sum_{k=1}^{n_g} x_{ik}\beta_{jk} + \epsilon_{ij} \tag{2.1}$$

where $i = 1, 2, ..., n$ and $j = 1, 2, ..., n_t$. That is, for individual i, the value of the jth influenced trait is determined by a linear combination of expression levels of the n_g target genes in View 1, plus some random noise ϵ_{ij} which is assumed to follow a normal distribution with mean and variance later specified. Finally we replace the original values of the element $y_{i,j}$, $i = 1, 2, ..., n$ and $j = 1, 2, ..., n_t$ with the values resulted in Equation (2.1).

For simplicity, we set $\mu_{j,1}$ to be 3 and $\sigma_{jj,1}$ to be 0.1 for $j = 1, 2, ..., p$ in View 1 and we set $\mu_{j,2}$ to be 5 with $\sigma_{jj,2}$ 0.1 for $j = 1, 2, ..., q$ in the View 2. In the core design, we select the first 10 genes in View 1 and the first $n_t = 5$ variables in View 2 to be our "target genes" and "influenced traits" across the two domains respectively, this allow us to check the performance of the CCA methods more easily by plotting the canonical loadings.

2.1.2 The existence of intra-domain correlations

However the intra-domain correlation does exist in reality(e.g. co-expression of genes, correlation between SIF variables). To reflect this concern in the artificial data, we introduced the intra-domain correlation in the following way as a variation to our core design. The covariance matrix of the genetic domain will have the following tri-diagonal form,

$$\Sigma_1 = \begin{pmatrix} \sigma_{11,1} & \sigma_{12,1} & & & \\ \sigma_{21,1} & \sigma_{22,1} & \sigma_{23,1} & & \\ & \sigma_{32,1} & \ddots & \ddots & \\ & & \ddots & \ddots & \sigma_{(p-1)p,1} \\ & & & \sigma_{p(p-1),1} & \sigma_{pp,1} \end{pmatrix}$$

where $\sigma_{ij,1} = \rho_1\sigma_{ii,1}\sigma_{jj,1}$ for $i, j = 1, 2, ..., p$, ρ_1 is the correlation level between genetic variables to be specified later. Instead of picking the first 10 genetic variables as the target genes in the core design, we choose the #1, #2, #11, #12, #21, #22, #31, #32, #41 and #42 to be the truly associated genes in this variation design, such that each of them is correlated to one truly associated gene and one noise variable.

The covariance matrix of the trait domain will have the following compound symmetric form,

$$\Sigma_2 = \begin{pmatrix} \sigma_{11,2} & \sigma_{12,2} & \cdots & \sigma_{1q,2} \\ \sigma_{21,2} & \sigma_{22,2} & & \vdots \\ \vdots & & \ddots & \sigma_{(q-1)q,2} \\ \sigma_{q1,1} & \cdots & \sigma_{q(q-1),2} & \sigma_{qq,2} \end{pmatrix}$$

where $\sigma_{ij,2} = \rho_2 \sigma_{ii,2} \sigma_{jj,2}$ for $i, j = 1, 2, ..., q,\ \rho_2$ is the correlation level between trait variables to be specified later. We remain to choose the first 5 trait variables as the truly associated ones.

2.1.3 Design of Experiment

The primary interest of our evaluation study is assessing the performance of various CCA methods when they are applied to the high dimensional and high background noise multi-view type of dataset that often arise in genetic studies. Similar studies were carried out in order to demonstrate the effectiveness of the proposed sCCA approach in some other articles (Chu, Liao, Ng, & Zhang, 2013a; Hardoon & Shawe-Taylor, 2011; Waaijenborg et al., 2008; Witten et al., 2009), where high dimensional matrices existed in both domains. However as illustrated by our motivating question, "asymmetric high dimensional multi-view data" can rise naturally in many genetic studies, where we typically have a high dimensional data matrix for the genetic domain and low-dimensional matrix for the trait domain. Evaluation of the performance of CCA methods on this type of data was not found in existing literature.

The ultimate goal of applying CCA to high dimensional data is that the process could simultaneously detect and portray the inter-domain correlation while correctly identifying the truly associated variables across two domains among a large number of background noise feature variables. I am also be interested in comparing the efficiency of different approaches. To serve these objectives, I propose to adopt the following metrics to the CCA model output. Terminologically, let the "effect size" of a variable be the absolute value of its assigned canonical coefficient.

- **Distinctiveness of interactive variables (DIV)**

  Defined as the number of the truly associated variables successfully identified by the CCA method. More specifically, it is the number of truly associated variables with the effect size greater than the mean effect size of the noise variables. We are interested in whether the process assigns significantly non-zero canonical coefficient to the interactive variables. Ideally the true interactive features should be assigned with canonical coefficients with significant magnitude.

- **Level of Sparsity (S%)**

  Defined as the ratio of number of noise features with zero canonical loading versus the total number of noise features in a particular domain. This metric indicates how effectively a CCA method suppresses the noise features in a given domain.

- **Mean and Standard deviation of noise loadings (M, SD)**

  The mean and standard deviation of the canonical loadings assigned to all noise features, in order to examine how are these coefficients distributed.

- **Degree of Separation (DOS)**

  Defined as the quotient of the mean effect size of the truly associated features divided by the mean effect size of the noise features. The value of DOS ranges from 0 to ∞, the greater DOS value indicates stronger separation between the truly associated feature and the background noise.

- **Canonical Correlation (CC)**

  The correlation between the first order canonical covariates.

- **Running Time (RT)** Measurement in seconds of the amount of time each CCA method requires to run, within the same system computing environment.

It is crucial to point out that none of these metrics serves as a single-best metric to model performance.

2.2 Evaluation of CCA methods

2.2.1 Preparation of artificial test data

Our motivating questions suggest that in real world genetic studies we could be potentially required to handle feature vectors of dimensions approximately 10,000 for View 1 and 10-15 for View 2. For this evaluation study, I aimed to create an artificial dataset that is roughly 1/10 of the real-world dimension, and assess the performance of various canonical correlation analysis methods. The variation design of the artificial data considering existence of the intra-domain correlation is used here. The specifications of the artificial multi-view dataset are presented using Table 2.1, with the Gene-Traits association mapping coefficients presented in the following Table 2.2.

Table 2.1: Specifications of Simulated data for evaluation of CCA methods

Dimension	
Total Observations n	100
Dimension of View 1 features p	1000
Dimension of View 2 features q	10
Number of designated genes	10
Number of designated traits	5
Embedded Gene - Trait Association	
Target Genes (position in feature array)	View 1 variables #1, #2, #11, #12 #21, #22, #31, #32, #41 and #42
Influenced Traits (position in feature array)	View 2 variables #1 to #5
Correlation level of genetic variables ρ_1	0.5
Correlation level of trait variables ρ_2	0.85

Table 2.2: Specifications of simulated gene-trait mapping coefficients

Traits\Genes	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10
t1	0	0	1	1	1	1	-1	-1	-1	-1
t2	-1	-1	0	0	1	1	-1	-1	1	1
t3	1	-1	1	-1	0	0	1	-1	1	-1
t4	-1	-1	-1	1	1	1	0	0	-1	1
t5	1	1	1	-1	-1	-1	1	-1	0	0

2.2.2 Regularized Canonical Correlation Analysis

When the number of feature variables exceeds the sample size, a regularization step is required before conventional CCA can be applied, in order to avoid singularity and ensure invertibility. Existing literature provides two options for regularization, namely regularization through cross-validation or through shrinkage. In this study I examines both options.

rCCA via Shrinkage

Regularized CCA via shrinkage was carried out through the `rcc` function in the `mixOmics` package by setting the `method = ‘‘shrinkage’’` (Rohart et al., 2017).

The embedded shrinkage regularization process yielded regularization parameters of 0.9481 and 0.1066 for λ_1 and λ_2 respectively. Subsequently, we obtained the first three order canonical correlations to be 0.5999, 0.5932 and 0.5776 respectively.

Figure 2.1 shows a plot of the canonical coefficients for both genetic and trait domain against the corresponding variables, with loading coefficients associated to the “target genes” and “influenced traits” plotted in solid triangle symbol and marked in

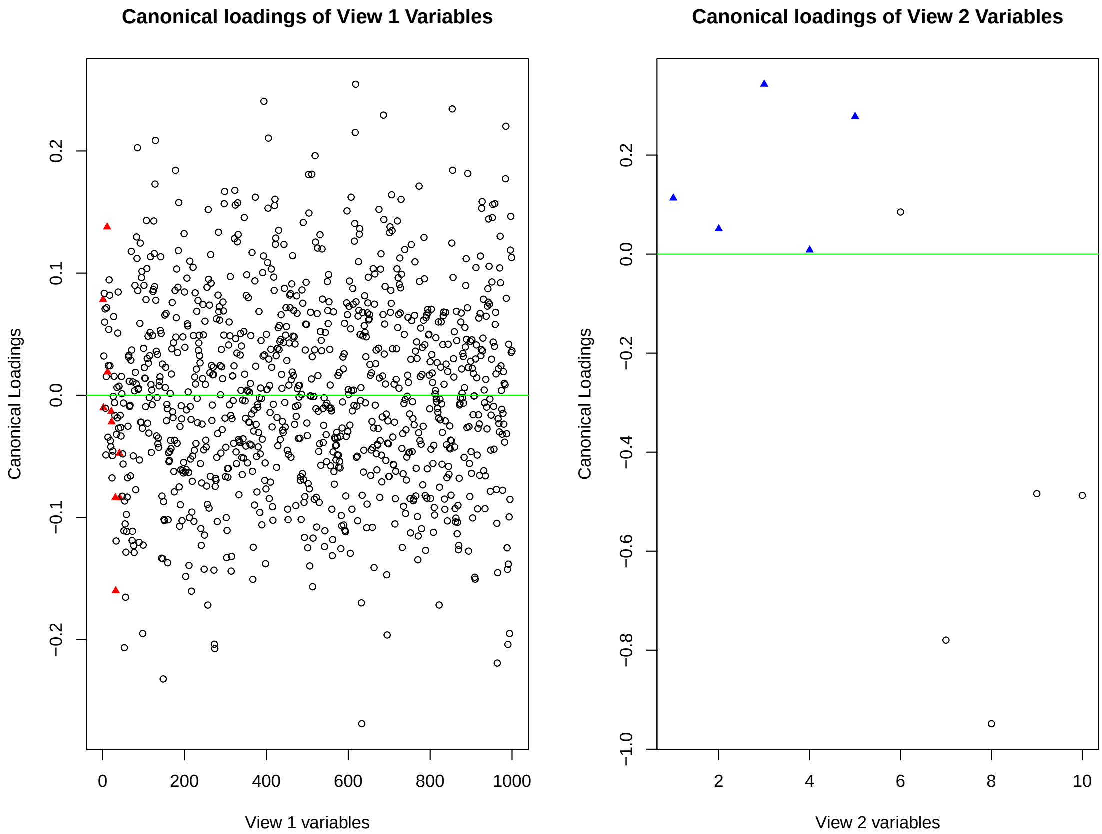

Figure 2.1: Plot of the canonical loadings vs. feature variables under the shrinkage-rCCA. Green horizontal line indicates the zero level. Truly associated variables in View 1 and View 2 are marked by red and blue triangle symbols respectively.

red and blue respectively, and the model output of the rCCA via shrinkage regularization are summarized by Table 2.3. In the genetic domain, the DIV scored 5/10, which means 5 out of 10 truly associated genes were successfully identified, where in the trait domain, the DIV scored 0/5, that is none of the truly associated traits were identified. In both domains, there are also many unassociated variables being assigned large coefficients as shown in Figure 2.1, this is also reflected in the relatively low score of DOS, 1.0312 and 0.2849 for the genetic and trait domain, respectively.

rCCA via Cross Validation

The regularization via cross-validation(cv-rCCA) was carried out by function `estim.regul()` in the `CCA` package, yielding λ_1 and λ_2 as the regularization parameters González and Djean (2012). The cross-validation process yielded regularization parameters 0.75025 for λ_1 and 0.001 for λ_2. The regularized CCA was subsequently performed by function `rcc` using the obtained regularization parameters.

Method of Regularization	Shrinkage
Regularization Parameters	
λ_1	0.9481
λ_2	0.1066
First three order Canonical Correlations	0.5999, 0.5932 and 0.5776
Non-Zero Parameters in View 1	1000
Non-Zero Parameters in View 2	10
Distinctiveness of Int. variables (DIV)	
View 1	5/10
View 2	0/5
Level of Sparsity (S%)	
View 1	0%
View 2	0%
Degree of Separation (DOS)	
View 1	1.0312
View 2	0.2849
Mean & Standard deviation of Canonical Loadings	
View 1	0.002822, 0.07951
View 2	0.5229, 0.3935
Running Time	1.3205 secs

Table 2.3: Summary of model output by Shrinkage rCCA

Figure 2.2 shows the plot of the canonical loadings versus the corresponding variables for both domains, and the model output of the cv-rCCA are presented in Table 2.4. The plot suggests that the the cv-rCCA resulted in lower effect sizes of the noise variables compare to the sh-rCCA. This is reflected in the performance metrics - in the genetic domain, 5 of 10 truly associated genes were successfully identified and in the trait domain, 2 of 5 truly associated traits were identified; in both genetic and trait domain, the cv-rCCA achieved higher DOS scores of 2.0277 and 1.2140, which indicates better separation of the true variables from the background noise. All coefficients naturally remain non-zero and contribute to the background noise as rCCA has no way to introduce any sparsity.

Method of Regularization	Cross Validation
Regularization Parameters	
λ_1	0.75025
λ_2	0.001
First three order Canonical Correlations	0.7802, 0.6924 & 0.7671
Non-Zero Parameters in View 1	1000
Non-Zero Parameters in View 2	10
Distinctiveness of Int. variables (DIV)	
View 1	5/10
View 2	2/5
Level of Sparsity (S%)	
View 1	0%
View 2	0%
Degree of Separation (DOS)	
View 1	2.0277
View 2	1.2140
Mean & Standard deviation of Canonical Loadings	
View 1	0.0005254, 0.02180
View 2	-0.2636, 0.6122
Running Time	18264.8394 secs

Table 2.4: Summary of model output by Cross-Validation rCCA

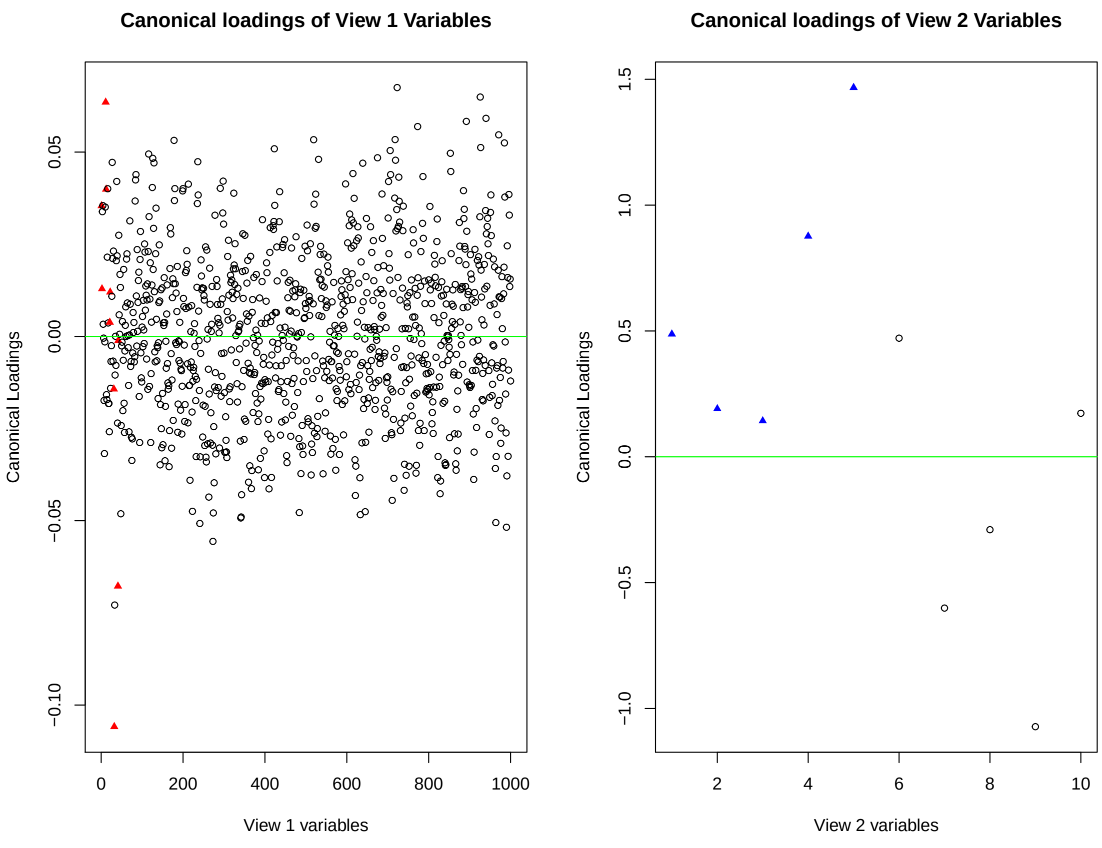

Figure 2.2: Plot of the canonical loadings vs. feature variables under the Cross Validation-rCCA. Green horizontal line indicates the zero level. Truly associated variables in View 1 and View 2 are marked by red and blue triangle symbols respectively.

2.2.3 Evaluation of Sparse CCA

The sparse canonical correlation analysis (sCCA) developed by Witten et al. (2009) is carried out by the `CCA` function within the R-packge `PMA` (Witten, Tibshirani, Gross, & Narasimhan, 2018).

The sCCA was applied to the same artificial data. Figure 2.3 shows the plot of the canonical loadings versus the corresponding variables for both domains, and the model output of the sCCA are presented in Table 2.5. The sCCA successfully identified 8 of 10 truly associated genes and only 1 of 5 truly associated traits. The sCCA successfully introduced sparsity to the output by setting the loading of most noise variables to zero or very close to it, the genetic and trait domain achieved 99.09% and 100% level of sparsity. The degree of separation of the true variables from the background noise was also improved, the genetic and trait domain achieved DOS score of 491.71 and `Inf` (which indicates perfect separation) respectively, which is a great improvement compare to the result of rCCA methods.

Num non-zeros u's:	9
Num non-zeros v's:	1
Penalty for x(L1 Bound):	0.1
Penalty for z(L1 Bound):	0.1
Cor(Xu,Zv):	0.9601
Distinctiveness of Int. variables (DIV)	
View 1	8/10
View 2	1/5
Level of Sparsity (S%)	
View 1	99.09%
View 2	100%
Degree of Separation (DOS)	
View 1	491.71
View 2	`Inf`
Mean & Standard deviation of Canonical Loadings	
View 1	5.9298×10^{-5}, 0.006632
View 2	0, 0
Running Time	6.7692 secs

Table 2.5: Summary of model output by Sparse CCA

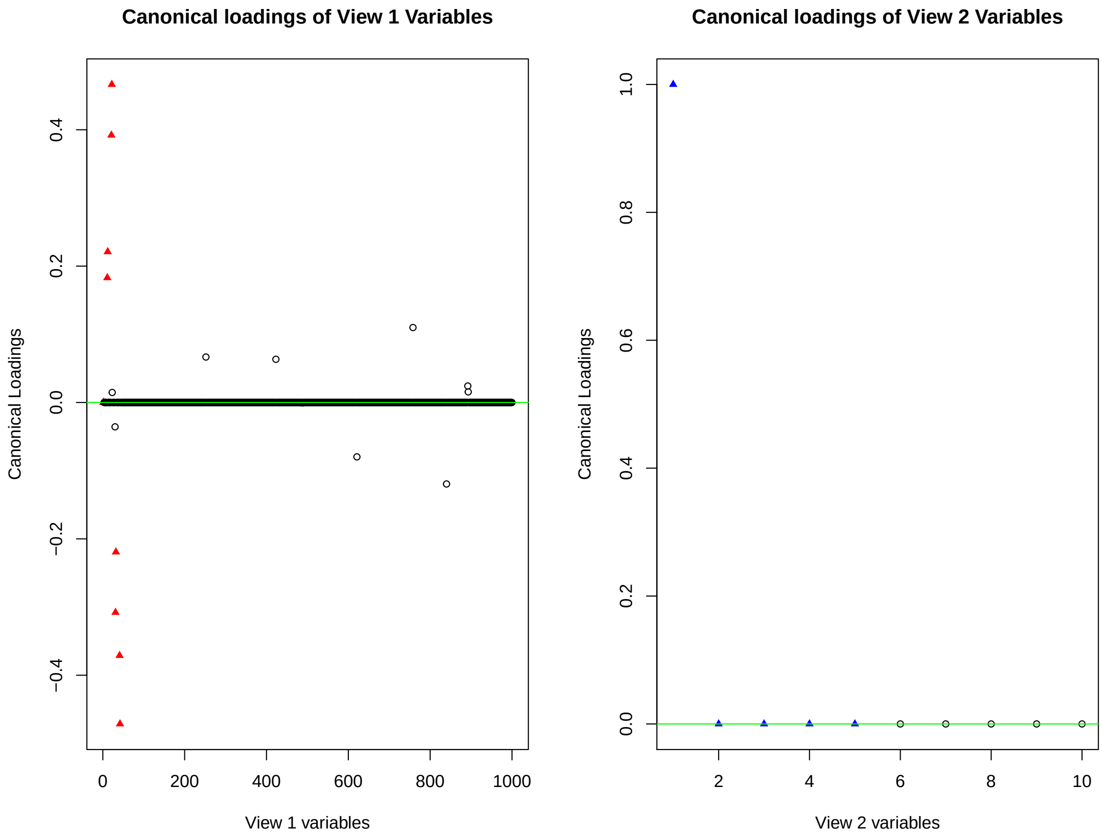

Figure 2.3: Plot of the canonical loadings vs. feature variables under the sparse CCA. Green horizontal line indicates the zero level. Truly associated variables in View 1 and View 2 are marked by red and blue triangle symbols respectively.

2.2.4 Comparison and Discussion

We draw the following conclusions from the output of our evaluation study,

- In our experiment the cv-rCCA outperformed the sh-rCCA in terms of the model quality. In View 1, despite both method identified the same number of truly associated variables, the cv-rCCA was able to yield model with lower effect sizes for the noise variables, this is confirmed by the higher DOS value achieved by the cv-rCCA. In View 2, the sh-rCCA failed to identified any of the truly associated trait variables due to large loadings assigned to the noise variables, the cv-rCCA identified 2 of 5 truly associated variables.

- The sparse CCA method appeared to outperform both regularized CCA methods. In View 1 the sCCA detected 8 of 10 truly associated genetic variables, however in View 2, the method identified only 1 of 5 truly associated trait variables, and excluded all other variables from the model. In both domain, the sCCA was able to effectively suppress the weights of the most non-interactive coefficients to near zero level, if not exactly zero. This fact is reflected by the DOS score. the sCCA achieved 491.71 and `Inf`(perfect separation) in View 1 and View 2 respectively. The sCCA process returned the first order canonical correlation of 0.9601, which also improved on both rCCA methods.

- Another important consideration is the computational cost of the proposed method. To carry out the same analytical task using the same machine, it took the sh-rCCA 1.32 seconds and the cv-rCCA 18264.83 seconds, that is cross validation regularization outperforms the shrinkage method in terms of the model quality at the cost of much greater computing time. On the other hand, sCCA is able to yield much more satisfying output with only 6.77 seconds, making sCCA the most favourable method among the three.

Chapter 3

Evaluation of Correlation Clustering

3.1 Overview

The efficacy and performance of the correlation clustering algorithm was tested and evaluated using an artificial multi-view dataset with two intrinsic clusters.

The artificial dataset was created for the purpose of evaluating the effectiveness and performance of the correlation clustering algorithm in the high dimensional setting. Fern et al. (2005) tested the proposed correlation algorithms using a simple two-cluster artificial dataset and found that the proposed method was able to correctly partition the testing data and assign instances to the appropriate cluster according to the underlying correlation sub-structure introduce by design. However the case of high dimensional and high background noise data were not addressed in Fern et al. (2005), where the dimension of the features greatly exceeds the sample size and the interested correlation structure exists amongst only a small collection of features in the presence of strong background noise. I believe that this algorithm should also be applicable to this type of dataset, as the two intuitions on which the algorithm was based should remain true under the new setting. However there remain questions and concerns regarding how the high dimensional feature and sparse nature of the target variables impact the clustering performance. In each iteration, the reassignment of individual to clusters relies on the quality of the local CCA model output of the interim clusters, which was greatly challenged by the high dimensional data with sparse target variables.

Two versions of the correlation clustering algorithms will be tested in this simulation study with one uses regularized CCA and one uses the sparse CCA. The primary modification to the correlation clustering algorithm was to replace the conventional

CCA component in Step 2 of the original algorithm with regularized CCA or sparse CCA. For the regularized CCA correlation clustering, cv-rCCA will be used as the previous simulation study had shown that sh-rCCA failed to effectively extract the correlation structure of a multi-view dataset in the high dimensional setting, despite it having a significant advantage in computational cost compared to the cv-rCCA. The code for these two correlation clustering algorithms have been written in R and provided in Sections B.8 and B.9 of Appendix B.
My experiment primarily investigate the following two aspects of the output:

- **Error rate of clustering output** - The assigned membership of two resulting clusters are compared with the true membership. Error rate is calculated as the percentage of subjects being assigned with false membership.
- **Local CCA model output** - The final local CCA models of the clustering algorithm were examined for whether they successfully captured the correlation structure of the original clusters by our design(e.g. whether the truly associated variables were correctly identified).

Section 3.2 discusses the design and specification of our artificial test data. The implementation of the two versions of the correlation clustering methods are in Sections 3.3 and 3.4 for rCCA and sparse CCA respectively. Results are discussed in Section 3.5.

3.2 Preparation of artificial data

The artificial data was created by stacking two multi-view datasets that differ by the target variables and mapping coefficients. The core design of the artificial data, which assumes no intra-domain correlation, is used here. For the convenience of visually examining the result, we select the first 10 and the last 10 View 1 variables to be the "target genes", and the first 5 and the last 5 View 2 variables to be the "influenced traits", for the Cluster 1 and Cluster 2 respectively. The mapping coefficients for Cluster 1 and Cluster 2 are given in the Table 3.2 and 3.3 respectively. The Cluster 1 and Cluster 2 datasets have distinct correlation structures, and the combined dataset has correlation sub-structure. A summary of two clusters in the multi-view dataset is in Table 3.1. The goal of the test is to examine the ability of the correlation clustering algorithm to identify the true clustering scheme and recover the local correlation structure within each cluster. The R-code for creating the simulated data is available in Section B.7.

Table 3.1: Specifications of the two-cluster simulated data

	Cluster 1	Cluster 2
Matrix Dimension		
View 1 Features	1000	1000
View 2 Features	10	10
Sample size	100	100
Target Variables		
View 1	#1, #2,...,#10	#991, #992,...., #1000
View 2	#1, #2,..., #5	#6, #7,..,#10

Table 3.2: Specifications of gene-trait mapping coefficients in Cluster 1

Traits\Genes	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10
t1	0	0	1	1	1	1	-1	-1	-1	-1
t2	-1	-1	0	0	1	1	-1	-1	1	1
t3	1	-1	1	-1	0	0	1	-1	1	-1
t4	-1	-1	-1	1	1	1	0	0	-1	1
t5	1	1	1	-1	-1	-1	1	-1	0	0

Table 3.3: Specifications of gene-trait mapping coefficients in Cluster 2

Traits\Genes	g991	g992	g993	g994	g995	g996	g997	g998	g999	g1000
t6	1	-1	1	-1	1	-1	1	-1	0	0
t7	-1	1	-1	1	-1	1	0	0	-1	1
t8	1	1	1	1	0	0	-1	-1	-1	-1
t9	-1	-1	0	0	-1	-1	1	1	1	1
t10	0	0	1	-1	1	-1	1	-1	1	-1

3.3 Evaluation of rCCA correlation clustering

The cross validation regularization was adopted for the rCCA clustering as the cv-rCCA exhibited stronger performance over the sh-rCCA in the previous evaluation study. A leave-one-out cross validation was performed over the entire simulated data (two clusters combined) in order to seek for the optimal regularization parameters. The process returned $\lambda_1 = 1$ and $\lambda_2 = 0.001$ with CV-score 0.1378.

The cv-rCCA based correlation clustering was then applied to the simulated data with the obtained regularization parameters. The number of clusters k was set to 2 and the clustering algorithm was run for 50 iterations. The clustering algorithm partitioned the simulated dataset into two clusters with error rate oscillating between 44.5% to 45.5%. 113 subjects were assigned to the Cluster 1 and 87 were assigned to the cluster 2. The clustering algorithm had not converged after 50 iterations, as the error rate started to oscillate after initial decline. The interim error rate of the iteration process is plotted in Figure 3.1. The local CCA model output is summarized in Table 3.4. The local loadings are plotted in Figure 3.2 for visual examination of the quality of local CCA models.

Table 3.4: Summary of local CCA model output for the rCCA-clustering

	Cluster 1		Cluster 2	
	View 1	View 2	View 1	View 2
Identified True Variables	7/10	2/5	8/10	4/5
Degree of Separation	1.5867	0.6142	1.5382	1.5463
Canonical Correlation	0.7258		0.7654	

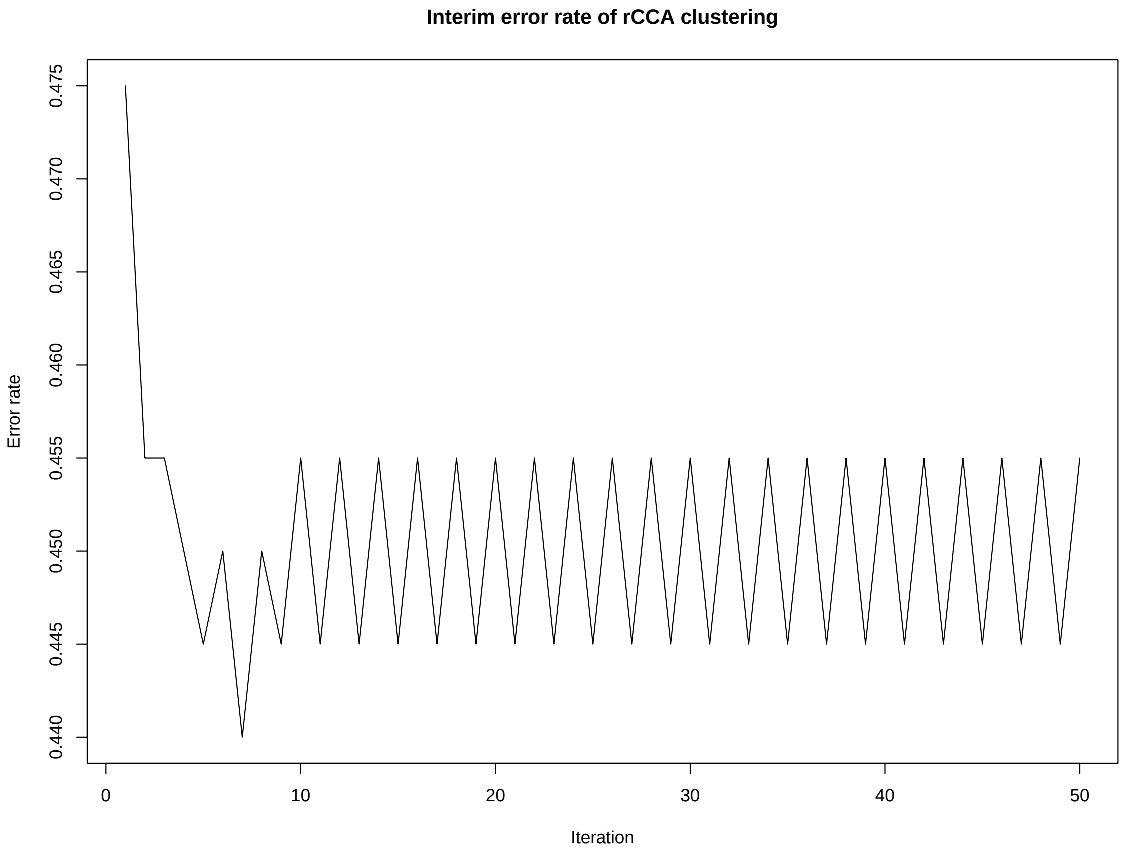

Figure 3.1: The interim error rate of the rCCA based correlation clustering.

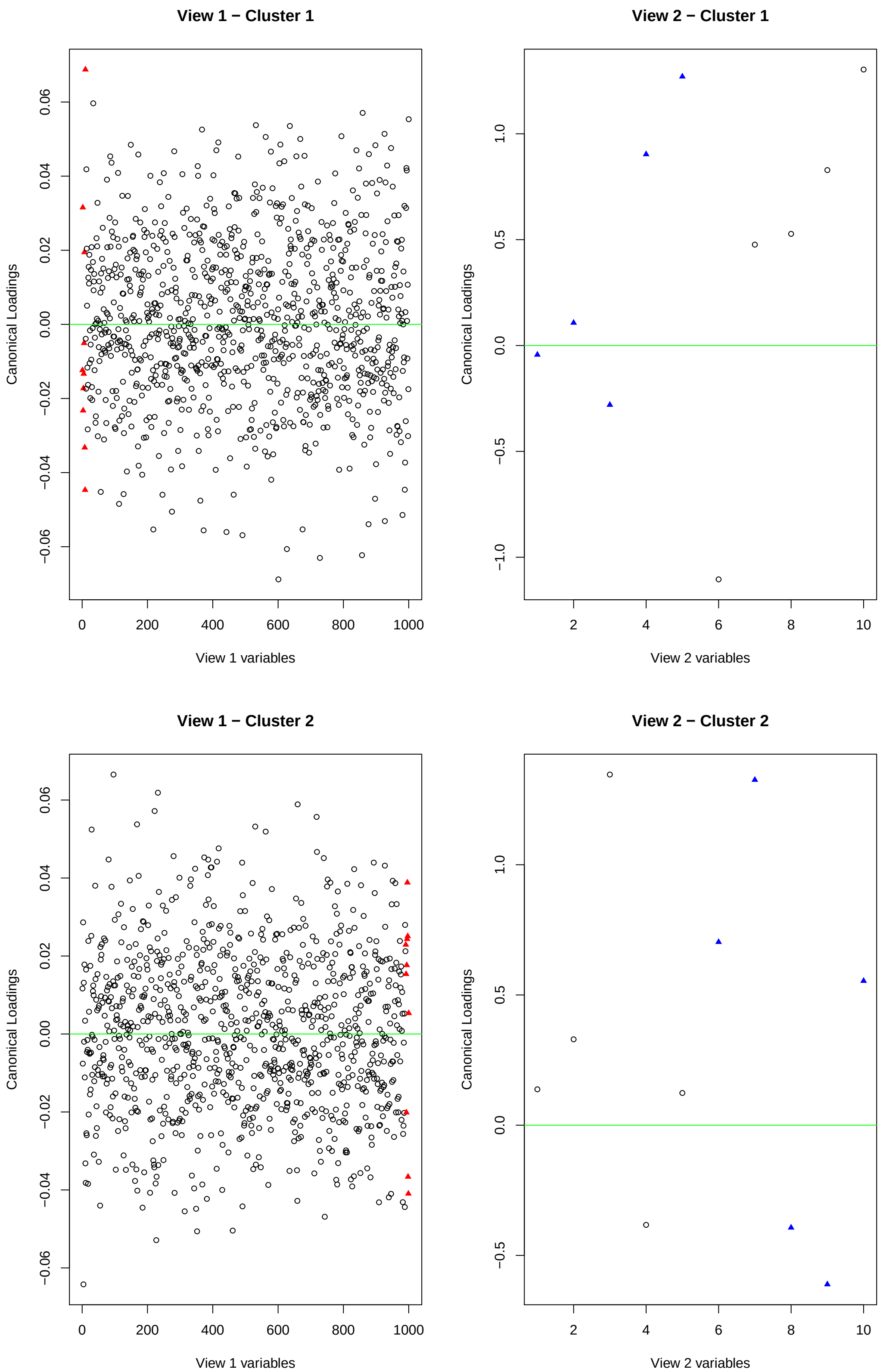

Figure 3.2: Plot of the canonical loadings vs. feature variables of both clusters under the rCCA-clustering. Green horizontal line indicates the zero level. Truly associated variables in View 1 and View 2 are marked by red and blue triangle symbols respectively.

3.4 Evaluation of sCCA correlation clustering

The sparse CCA correlation clustering was tested on the same simulated data with number of clusters k set to 2 and iteration set to 50. The clustering algorithms perfectly partitioned the simulated dataset and recovered the original clustering scheme with 0 % error rate, The clustering algorithm converged after 9 iterations. The interim error rate of the iteration process is plotted in Figure 3.3 The local CCA model is examined and summarized in Table 3.5. The loadings are plotted in Figure 3.4.

Table 3.5: Summary of local CCA model output for the sCCA-clustering

	Cluster 1		Cluster 2	
	View 1	View 2	View 1	View 2
Identified True Variables	6/10	1/5	7/10	1/5
Degree of Separation	171.3502	`Inf`	279.4076	`Inf`
Sparsity Level	98.69%	100%	98.58%	100%
Canonical Correlation	0.8620		0.9095	

3.5 Discussion

In this evaluation study, sCCA clustering demonstrated the capability of correctly recovering the original clustering scheme and producing meaningful local CCA models output that rCCA clustering lacks.

In our experiment, The rCCA correlation clustering failed to partition the subjects correctly and the local CCA models in the two clusters failed to meaningfully isolate the truly associated variables from the unassociated ones. The sCCA clustering, on the other hand, was able to perfectly recover the original clustering scheme. In both local clusters under the sCCA clustering, the unassociated variables were assigned zero or very close-to-zero coefficients. This fact is also reflected in the DOS score, the degree of separation of the truly associated variables from the background noise was significantly improved by sCCA clustering compare to rCCA clustering. However some of the truly associated variables were not included in the sCCA local models, 6 and 7 out of 10 truly associated variables in View 1 were identified in Cluster 1 and Cluster 2 respectively, only 1 out of 5 true variable was detected in View 2 for both clusters.

The same evaluation was repeated a number of times using different seeds for the artificial data generator. I have made the following observation through my repeated

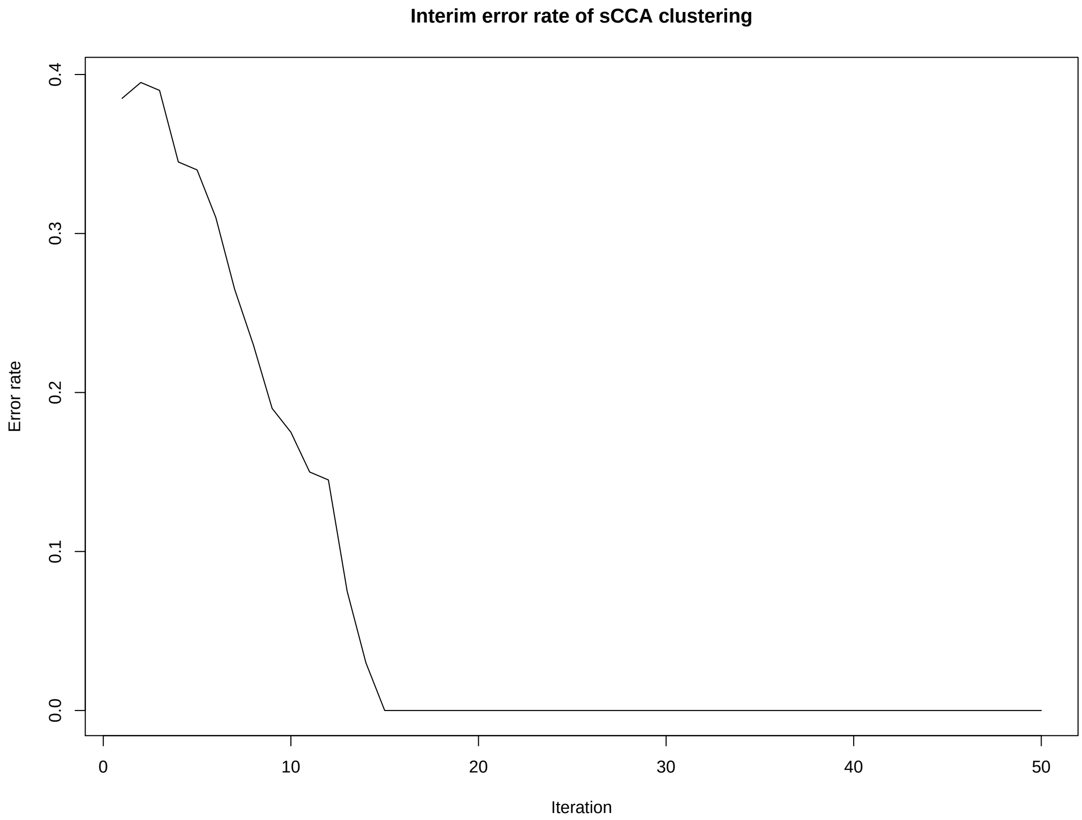

Figure 3.3: The interim error rate of the sCCA based correlation clustering. The error rate fairly consistently declined through the first 14 iterations. At the 15th iteration the error rate sharply jumped to 100% percent, despite the dramatic effect, this in fact indicates perfect segregation of the experiment data points - with the interim tags completely opposite to the original assignment.

trials, however for them to constitute an evidence-supported conclusion, a rigorous simulation study of a much larger scale is required in order to fully explore the capability and limitation of the correlation clustering algorithm in the high dimensional setting.

- There appears to be a trade-off between the strength of penalty used in the sCCA clustering versus the effectiveness of the clustering and the quality of the resulted local models - weak penalties tends to make the clustering fail as the truly associated variables could not be well separated from the background noise, while strong penalties tend to yield good clustering result at the cost of sacrificing some truly associated variables.

- The good performance of sCCA clustering is not guaranteed. While the sCCA clustering in general greatly outperforms the rCCA clustering, there are times sCCA clustering failed to output desired result.

These observations suggest that caution needs to be used when applying the sCCA clustering algorithm to the real world data.

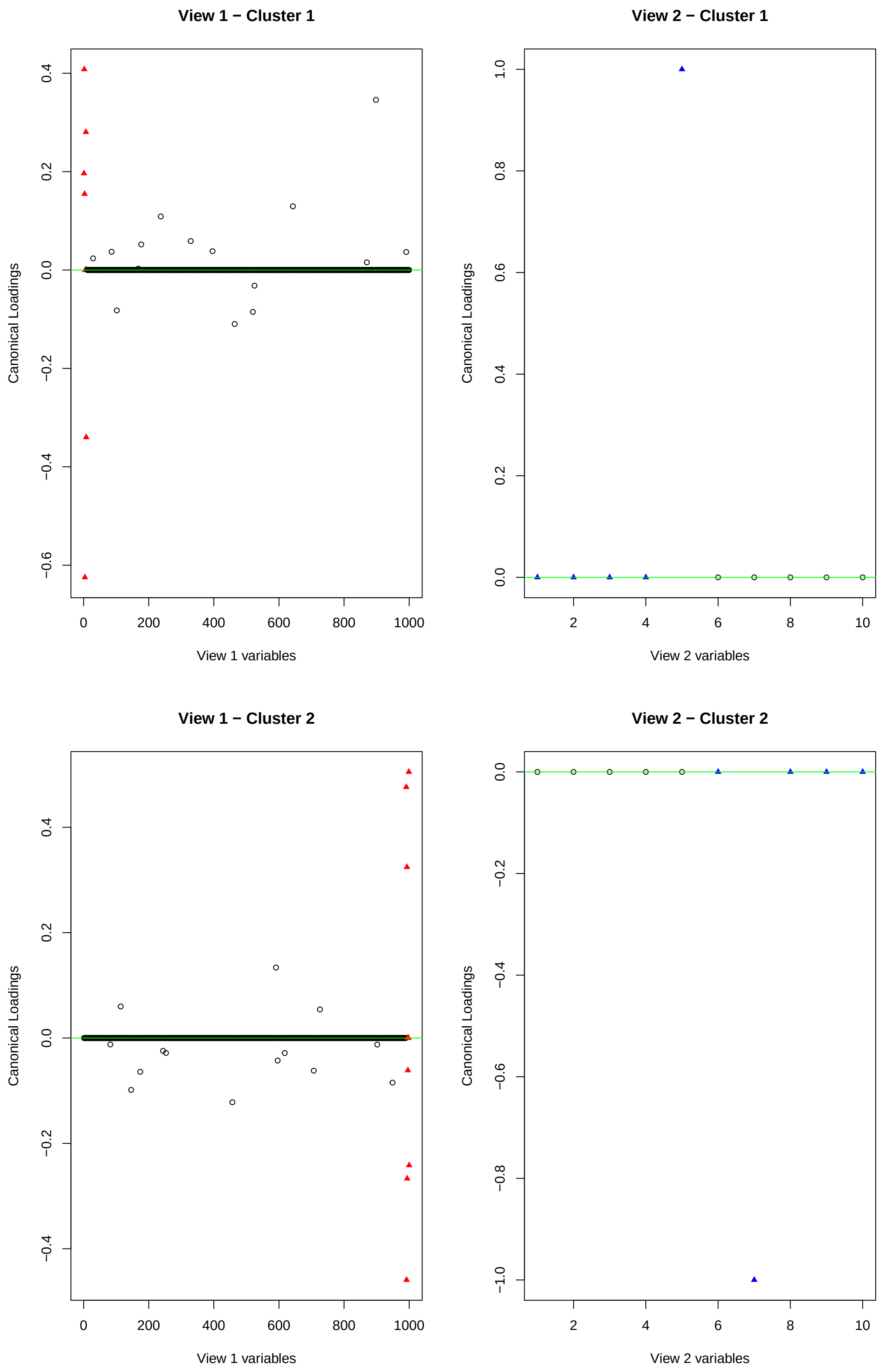

Figure 3.4: Plot of the canonical loadings vs. feature variables of both clusters under the sCCA-clustering. Green horizontal line indicates the zero level. Truly associated variables in View 1 and View 2 are marked by red and blue triangle symbols respectively.

Chapter 4

Applications

4.1 Expression-SIF multi-view data

Professor Sara Good modelled the mean logarithm of HbA1c level of DCCT participants against the PrediXcan-imputed genetically regulated expression plus other covariates in her recent study, where the gene expression profiles of DCCT subjects were imputed using the PrediXcan weights trained via the Depression Genes and Networks(DGN) Whole Blood model as well as the Version 6 GTEx tissue models. She generously granted me the permission to use the imputed gene expression data from her work for my thesis. Dr. Paterson and Dr. Roshandel from The Hospital for Sick Children, Toronto generously provided me access to the skin fluorescence(SIF) data of the DCCT subjects. In this thesis, for a demonstration of the sCCA method and correlation clustering algorithm and as an exploratory study, I will use the gene expression data as the View 1 matrix and the SIF data as the View 2 matrix, our goal is to examine the correlation structure of the Expression-SIF multi-view dataset, and investigate the existence of possible correlation sub-structure.

In this thesis, the imputed gene expression predicted by the DGN whole blood training will be used as it contains the highest number of genes and is not tissue-dependent. Gene expression profiles for 1304 DCCT subjects were imputed from their SNPs while the SIFs were measured only on a subset of 1082 subjects, and the two datasets presented the measurement of subjects in different order. To create a usable multi-view dataset, imputed expression table were inner-joined with the SIF table by the subject ID. View 1 and View 2 matrices were then further extracted from the joined table. Consequently View 1 matrix contains the imputed expression of 11538 genes of 1082 participants and View 2 contains 15 SIF measurements from the same participants. Data points were arranged in the same order by subject ID in both Views. The mean and variance of the SIF measurements were calculated and presented in Table 4.1, a

	Sample Mean	Sample Variance	Minimum	Maximum
SIF1	22.6678	22.8966	8.7043	53.9731
SIF2	25.2242	47.3764	11.1644	76.3693
SIF3	14.0543	7.8103	5.4363	28.8863
SIF4	8.4468	4.004	2.6774	19.5894
SIF5	8.4822	4.3003	3.0593	23.0109
SIF6	9.9712	6.2918	3.6186	28.4303
SIF7	6.4663	2.6045	2.3378	17.1020
SIF8	7.5909	3.7177	2.7486	20.9174
SIF9	3.7144	0.7909	1.3119	8.2732
SIF10	5.0766	1.7562	1.8972	13.7144
SIF11	2.8730	0.5312	1.0135	6.3975
SIF12	2.6472	0.4460	0.9626	5.9605
SIF13	2.0292	0.2558	0.9143	5.0389
SIF14	1.4757	0.1247	0.6725	3.2374
SIF15	1.3581	0.1039	0.6384	3.0007

Table 4.1: The mean and variance of SIF variables by SIF ID

box plot was created in Figure 4.1 and a correlogram of the SIF variables is available in Figure 4.2. A preliminary examination to the data suggest that the magnitude of the mean and variance of the SIF variables decline as SIF ID number increases, the box-plot also indicates that for all SIF variables appear to be right-skewed and fat-tailed, this is verified by the histograms of the SIF variables in Figure C.1. The correlogram suggests that the SIF variables are in general highly correlated to each other.

4.2 Application of Sparse Canonical Correlation Analysis

The Sparse CCA was applied to the expression-SIF multi-view data. The Lasso penalty was applied in the penalized matrix decomposition process to enforce sparsity of the canonical loadings. The penalty optimization function `CCA.permute()` yielded penalties 0.1 for both domains, which were subsequently applied to the sparse CCA model fitting. The sCCA model achieved correlation of 0.6585 between the genetic and SIF domains, with 193 genes from the expression domain and one (#7) from the SIF domain assigned with non-zero canonical loadings. The canonical loadings are plotted against variables by Figure 4.3.

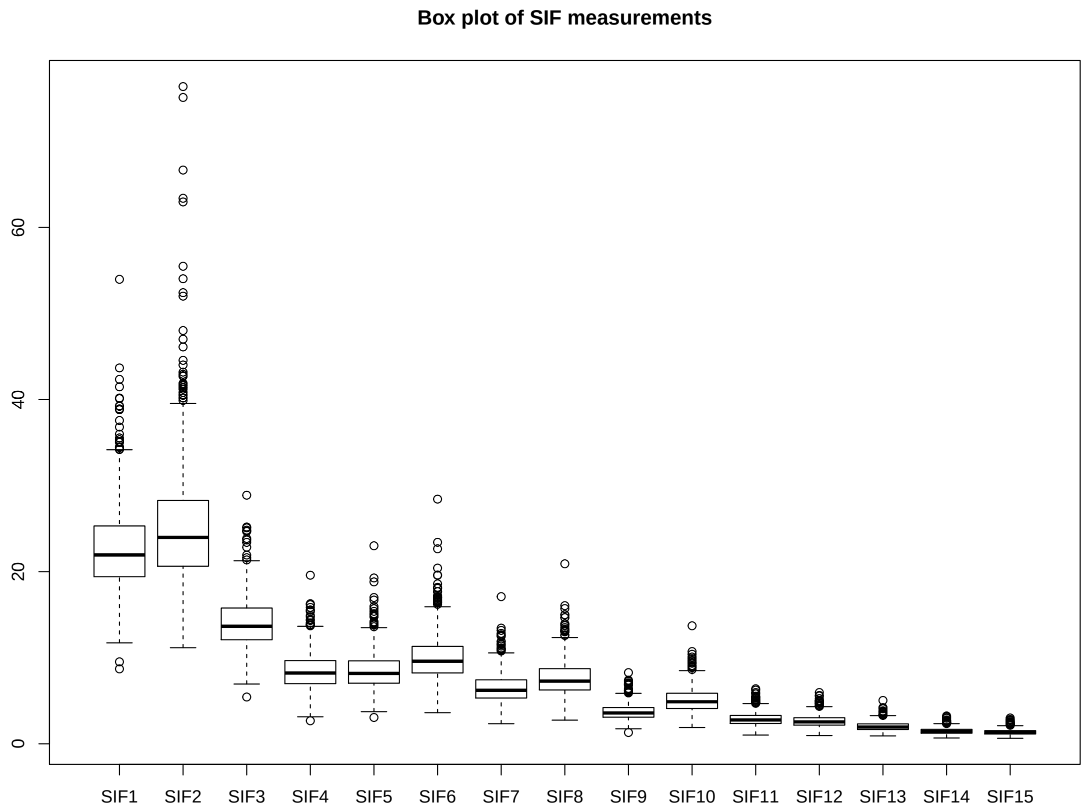

Figure 4.1: The box plots of SIF variables

In order to further investigate the association between the selected genes and the trait, a simple linear regression model (SLR) was fitted to the 193 identified genes individually, where the SIF #7 measurement was regressed against the PrediXcan imputed gene expression. T-tests were performed at significance level of $\alpha = 0.05$. To counteract the multiple testing problem, the significance level was adjusted by Bonferroni correction, that is, we test the null hypothesis of $\beta_i = 0$ at significance level $\tilde{\alpha} = \frac{0.05}{193} = 0.0002591$, for $i = 1, 2, \ldots, 193$, where β_i is the coefficient in the simple linear model for the ith gene.

A Manhattan plot is used to visualize the resulted p-values, where the $-log_{10}$ transformed p-value was plotted against the genes, as shown in Figure 4.4. Two genes, ENSG00000100281.9 and ENSG00000112787.8, were identified as significantly associated to the trait SIF #7.

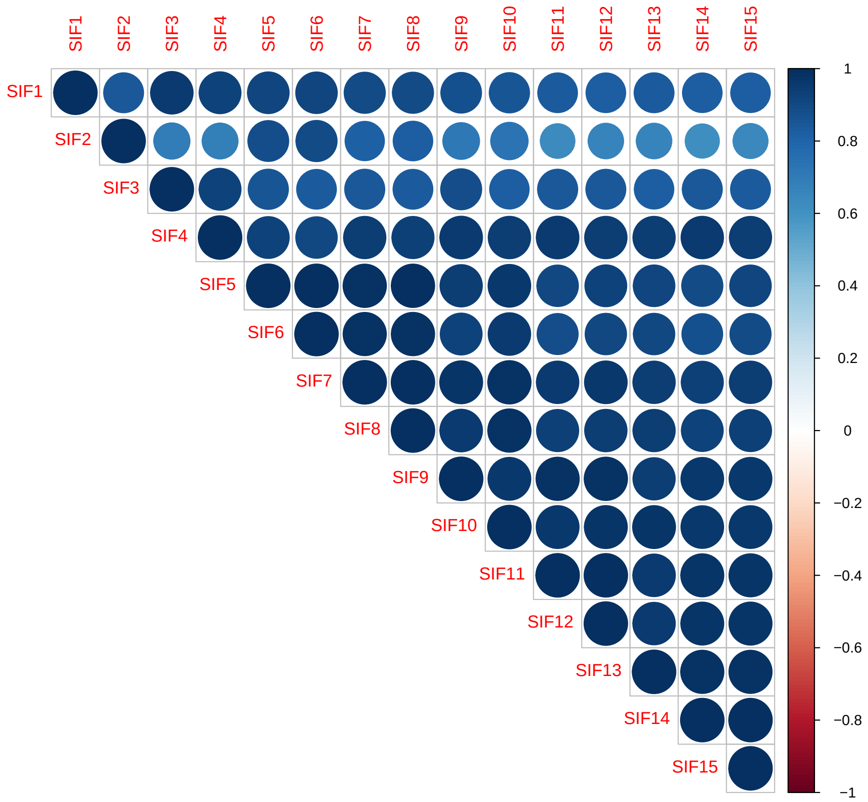

Figure 4.2: The Correlogram of SIF variables

4.3 Application of Correlation Clustering

The sCCA powered correlation clustering algorithm was then applied to the expression-SIF dataset. The number of clusters K was set to 2, 3, 4, 6 and 8, with 50 iterations in each case. The local cluster canonical correlations, the number of variables with non-zero loadings and the number of individuals in a cluster were extracted from the model output in order to examine the quality of clustering and local CCA models. The clustering output was summarized in Table 4.2.
Our result shows that all the local CCA models have some close-to-1 correlation at very low level of sparsity in the genetic domain. In these local models, the number of genetic variable with non-zero loadings greatly exceeded the number of elements in the cluster, indicating that the sparsity was not well enforced and result is likely implausible.

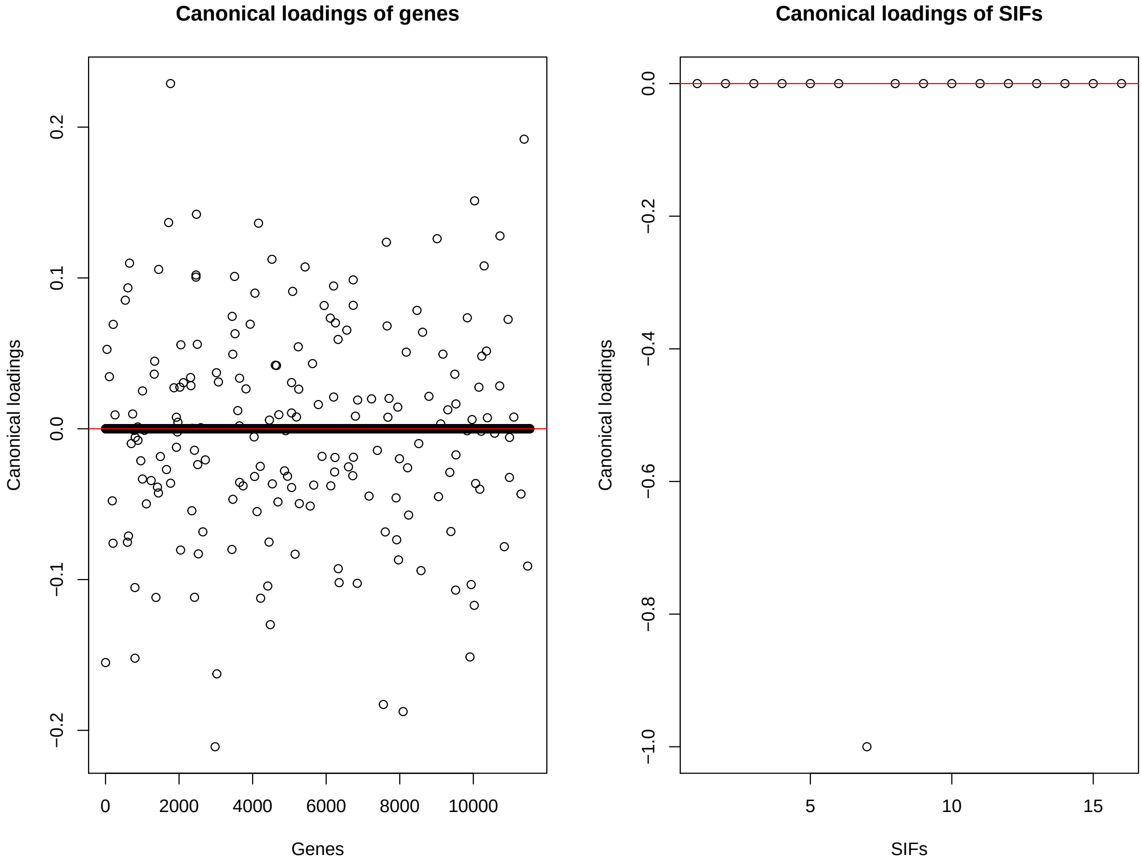

Figure 4.3: The canonical loadings of genetic and SIF variables of 1082 DCCT subjects. 193 genes and SIF#7 were identified and assigned non-zero canonical loadings. The horizontal red line indicates the zero level of efficacy.

Table 4.2: Summery of clustering output under different number of clusters

Num. of Clusters	Parameters	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8
K =2	Local correlations	0.9459	0.9795						
	Non-zero V1 Variables	9273	9299						
	Non-zero V2 Variables	10	11						
	Number of elements	727	355						
K =3	Local correlations	0.9891	0.9656	0.9567					
	Non-zero V1 Variables	9091	9680	9236					
	Non-zero V2 Variables	11	10	10					
	Number of elements	216	262	604					
K =4	Local correlations	0.9711	0.9557	0.9488	0.9945				
	Non-zero V1 Variables	8080	519	579	7655				
	Non-zero V2 Variables	9	1	1	10				
	Number of elements	401	270	296	115				
K =6	Local correlations	0.9709	0.9909	0.9303	0.9197	0.9938	0.9938		
	Non-zero V1 Variables	1223	9193	335	334	9136	9136		
	Non-zero V2 Variables	1	10	1	1	12	12		
	Number of elements	300	113	192	159	83	235		
K =8	Local correlations	0.9809	0.9363	0.9932	0.9934	0.9941	0.9174	0.9543	0.9954
	Non-zero V1 Variables	1182	339	3738	9303	9189	476	284	9261
	Non-zero V2 Variables	1	1	4	12	10	1	1	10
	Number of elements	175	201	93	79	82	232	156	64

4.4 Discussion

The sCCA method resulted in an interesting model output. 193 Genes were identified among 11538 candidates to be in association with SIF variable #7, under penalty 0.1

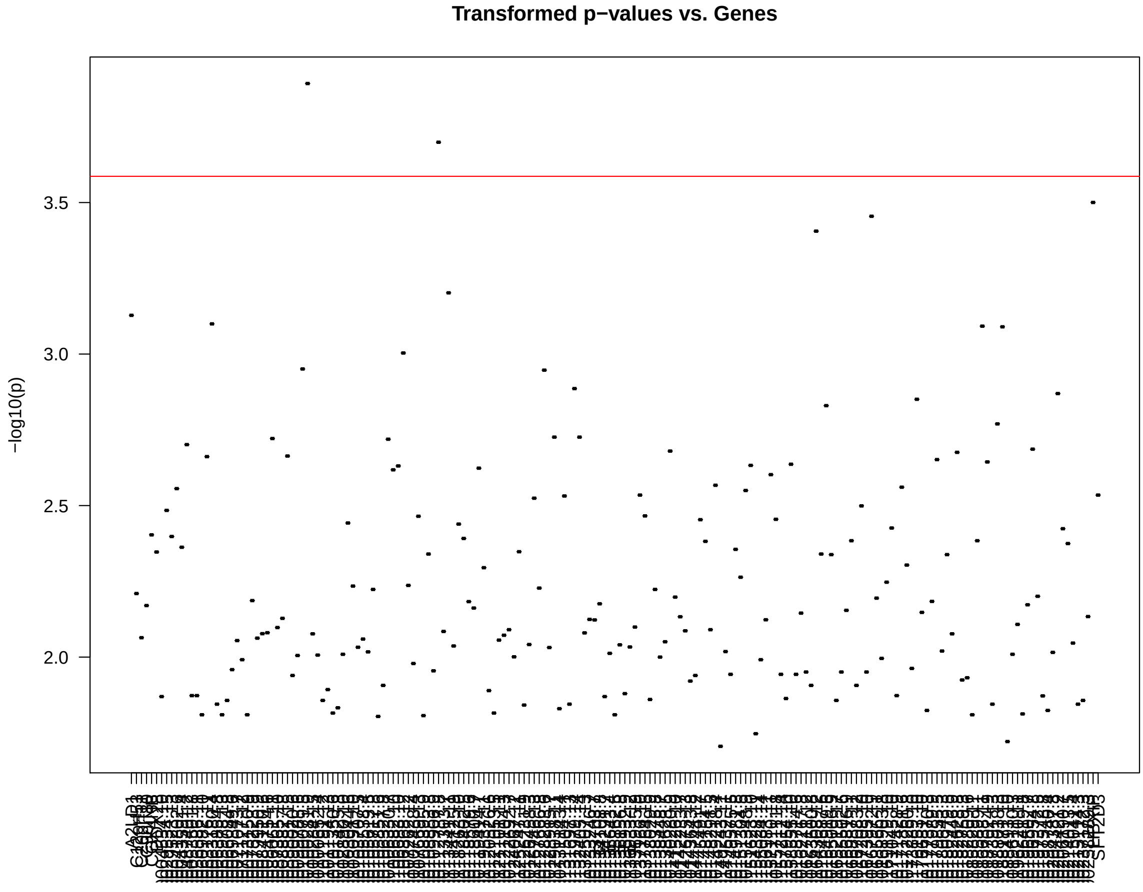

Figure 4.4: The Manhattan plot of the p-values of 193 individual t-tests to the identified genes, the red line indicates the significance threshold.

for both genetic domain and the SIF domain. To validate this result and explore how the value of the penalties impact the model output, a series of experiments were performed, which involved increasing and decreasing one penalties while keep another fixed. Our experiments confirmed that the penalty optimization function did yield the optimal result - increasing in penalty for the genetic domain will improve the sparsity in View 1 at the cost of reduced canonical correlation, while decreasing will do the opposite. Increasing the penalty for the trait domain does not exhibit any impact, however decreasing the penalty in the SIF domain will cause deterioration of the sparsity level in the genetic domain. Particularly, several attempts with strong penalty over View 1 (e.g. using 0.05, 0.025 and 0.0125) and relaxed penalty over View 2 (e.g 0.2 and 0.4) were made, in hope of obtaining a higher level of sparsity in the genetic domain while still including more SIF variables in the CCA model. Such ideal result was not attained as the correlation greatly suffered from these penalty values. The result of these experiments are presented in supplementary Table A.2.

The sparse CCA model identified a set of 193 genes that were correlated to the SIF #7. Further investigation of these 193 genes using simple linear regression revealed that only two of them, ENSG00000100281.9 and ENSG00000112787.8, are significantly associated to the trait SIF#7. Interestingly, the NAT2 (ensemble ID ENSG00000156006), the only gene found to be in association with the SIFs Eny et al. (2014), did not show up in the resulted CCA model, because in fact it was not included in the imputed expression predicted by the DGN training set.
I notice that the performance of the correlation clustering algorithm could be unstable through the earlier evaluation studies, however another possibility in this study is that the expression-SIF multi-view dataset has no intrinsic correlation sub-structure. The latter explanation is a more likely based on our observation, as among all local models under different clustering settings, No single local model significantly outperformed the output of the stand alone sparse CCA model (e.g. To achieve a higher level of correlation with the same/comparable level of sparsity.) Therefore in this study, we cannot claim to have found any plausible correlation sub-structure in the relationship between the imputed gene expression and the SIF measurements.

Chapter 5

Discussion

5.1 Conclusion

Multi-view datasets arise naturally in many disciplines of scientific research including genetic statistics, such as the motivating problem presented in Chapter 1, where the subject of study is portrayed by two sets of feature vectors. A widely used statistical method for investigating the correlation structure of multi-view datasets is the canonical correlation analysis, which seeks the projection coefficients to the features such that the resulted canonical variates are maximally correlated across two domains. However conventional CCA cannot be directly applied to the high dimensional data as the correlation matrix will be ill-conditioned in such case. To adapt the method to the high dimensional case, a regularization step is required before the conventional CCA can be performed. Two methods of regularization, Cross Validation and Shrinkage, were examined in this thesis. A combination of regularization and CCA creates the regularized CCA for high dimensional applications. Furthermore in most genetic studies, only a very small subset of genes across the genome are truly associated with the phenotype of interest. Therefore a sparse version of CCA may be useful. Multiple approaches for introducing sparsity to the CCA model have been developed and introduced in the literature. In this thesis, the sparse CCA developed by Witten et al (2009), was examined and presented.

An evaluation study was carried out to evaluate the performance of the regularized CCA and sparse CCA using artificially created high dimensional data with sparse truly associated variables intended to imitate real world genetic data. In this study sparse CCA demonstrated its suitability for the analysis of these datasets by successfully isolating most of the truly associated variables from the unassociated ones with high degree of sparsity and recovering the original correlation structure by design. For the rCCA, Cross validation outperforms the shrinkage method in terms of the

resulted model output, however such advantage came at much greater computational cost.
An important limitation to the canonical correlation analysis is that it is only designed to detect the global linear correlation structure between two domains and it does not perform well if some type of correlation sub-structure exists in the multi-view data. Fern et al (2005), developed a K-mean style correlation clustering algorithm to tackle this problem by incorporating a mixture of local linear CCA models each capturing the correlation sub-structure of a local cluster. The correlation clustering algorithm relies on recursively applying CCA to the local clusters and re-assigning subjects according to a correlation distance metric based on the local CCA model outputs. The original algorithm was developed based on the conventional CCA therefore unsuited for the high dimensional genetic data. To adapt the correlation clustering algorithm to the multi-view dataset of our interest, the conventional CCA steps in the original algorithm was replaced with the sparse or regularized CCA.
A second evaluation study was conducted to evaluate the efficacy and performance of the modified correlation clustering algorithm, where in the artificial two cluster test dataset, View 1 consisted a high dimensional matrix representing the genetic domain and View 2 consisted a regular matrix representing the trait domain. Our experiment showed that the rCCA based correlation clustering was completely ineffective in the high dimensional setting, as at each iteration, the lack of sparsity in the local CCA model output impaired the subsequent re-assignment and ultimately caused the clustering algorithm to fail. One the other hand, sCCA based clustering performed extremely well in our experiment, the clustering algorithm perfectly recovered the original clusters, and over each cluster, most of the truly associated variables in View 1 were identified with high degree of sparsity over the entire high degree domain, however we do noticed this came at the the sacrifice of losing most of the truly associated components in View 2 - only 1 out of 5 true trait variables were identified with all others assigned zero coefficient along with the unassociated variables. Through repeated trials using different seeds, I noticed that there appears to be a trade-off between the degree of penalty adopted by the local sCCA model and the effectiveness of clustering algorithm that - successful clustering scheme result were usually obtained under strong penalty, at cost of losing some true variables in the local models; and weak penalty tends to cause the clustering process to fail. I also noticed that the good performance of the sCCA clustering is not guaranteed, there are times sCCA clustering failed to generate desirable experiment result. However, to fully understand the capability and limitation of the correlation clustering algorithm

in the high dimensional setting require a massive scale of simulation study, which is beyond the scope of this thesis. This suggests the sCCA clustering needs to be applied with caution.
Both sCCA and sCCA based correlation clustering were applied to the expression-SIF multi-view dataset. Among 11538 candidate genes, 193 were originally identified to be in correlation with SIF#7 with canonical correlation 0.6585. Further examination of these identified genes using simple linear regression model suggests that only two genes, ENSG00000100281.9 and ENSG00000112787.8, are significantly associated to the trait SIF#7. Interestingly, NAT2, the only gene found to be in association with the SIFs by GWAS was not included in the result, as it was not included in the imputed expression predicted by the PrediXcan DGN whole blood training set. No plausible correlation sub-structure were discovered within the dataset using the correlation clustering method, as no single local cluster outperforms the application of a stand alone sCCA to the entire multi-view dataset.

5.2 Limitations of this Research and Future Research Directions

To fully explore the capability and limitation of the sCCA and the related correlation clustering algorithm, simulation study of a much larger scale, which incorporates greater variability, is required. Given the time and computational resource it demands, this was not possible for the scope of this thesis, however as a potentially powerful tool in statistical genetics, sCCA and the correlation clustering deserve a much more rigorous and comprehensive examination. For example, sCCA demonstrated its effectiveness on a multi-view dataset with intra-domain correlation in the way as it was introduced in our evaluation study, however such conclusion can not be well generalized as in reality the correlation pattern can be highly variable. The sCCA based correlation clustering demonstrated its capability of correctly partitioning the subjects and recover the true correlation sub-structure in our evaluation study. However, through repeated trials of the same experiments under different seeds, I noticed that in some few cases, the clustering did not return desirable clustering scheme. The underlying reason of this instability of performance was still unknown, investigating the global stability of the correlation clustering algorithm in the high dimensional setting requires tremendous amount of computing power, which unfortunately was not available to this study.

Our evaluation study assumed normality in the values of gene expression and SIF measurements, however a preliminary examination to the SIF data revealed significant positive skewness in the data, suggesting that the normality assumption was violated in the reality. For future study, one may consider repeat this evaluation study with some other distribution featuring fat-tail and positive skewness.
The PrediXcan imputed gene expression based on the DGN whole blood training set was used for this study for the purpose of demonstrating the application of sCCA and correlation clustering, as well as a brief exploratory study. It would be very interesting to extend the exploratory study to the imputed expression datasets based on other GTEx tissues training sets. Also, there is a distinction between PrediXcan imputed expression and the true gene expression - the imputed expression portrays the predicted variation of the expression level from the baseline level, therefore the imputed expression value can be positive or negative, where the true gene expression values are strictly positive. I used the PrediXcan imputed expression data in this study, because the actual transcriptome data is often unavailable as acquiring such data usually require invasive procedure. However in the case where the actual gene expression data is available, it would be very interesting to carry out the same study using the actual transcriptome data as the View 1 matrix and compare the result to that under PrediXcan imputed expression, this allows us to back test the effectiveness of the PrediXcan method. If sufficient computational resources and are available, one may even consider directly using the SNPs data as the View 1 matrix to investigate the correlation structure between genetic variation and trait of interests.
I modified the correlation clustering algorithm for the high dimensional setting based on the sCCA developed by Witten et al. (2009), however there are a few other approaches of sparse CCA in the existing literature (Parkhomenko et al., 2009; Waaijenborg et al., 2008). The performance of the correlation clustering algorithm based on these different sparse CCA methods deserves further examination. The correlation clustering algorithm examined in this thesis is a unsupervised method, that is, it does not make use of the trait measurements each observation. Appropriate use of the existing measurements allows us to potentially extend this clustering algorithm into a classification model, that is, a supervised learning method enable us to make prediction of trait measurement based on values of input variables. Given the time and resource it requires, this type of project is more suitable to a Ph.D level of study.

Appendix A

Supplementary Tables

A.1 Experiment of different Penalty parameters for sCCA method

Table A.1: sCCA model output under various penalty schemes

V1 Penalty	V2 Penalty	Correlation	Identified V1 variables	Identified V2 variables
0.4	0.1	0.8872	3163	1
0.2	0.1	0.797	828	1
0.1	0.1	0.6585	193	1
0.05	0.1	0.4326	56	1
0.025	0.1	0.2399	13	1
0.0125	0.1	0.1283	2	1
0.1	0.4	0.6603	206	4
0.1	0.2	0.6585	193	1
0.1	0.05	0.6585	193	1
0.1	0.025	0.6585	193	1
0.1	0.0125	0.6585	193	1
0.05	0.2	0.4326	56	1
0.025	0.2	0.2399	13	1
0.025	0.4	0.2409	13	5
0.0125	0.4	0.1304	2	4

Appendix B

Script

B.1 Simulated Data Generator

The following user-written function `SimGen()` produces the simulated multi-view dataset that reflect the core design. The inputs and outputs of this function are presented by Table C.2.

Function Inputs	
`n`	Sample size n
`p`	Dimension of View 1 Feature p
`q`	Dimension of View 1 Feature q
`ng`	Number of truly associated genes
`nt`	Number of truly associated traits
Function Ouputs	
`total_view`	The multi-view dataset in Matrix form
`view_1`	The Stand-alone View 1 matrix
`view_2`	The Stand-alone View 2 matrix
`Betas`	The artificial gene-trait mapping coefficients

Table B.1: Description of Inputs and Outputs of `SimGen` Function

```
#---------------The ‘‘Simgen()’’ function ---------------

# clear memory

rm(list = ls())

# MASS package
```

```
require(MASS)

#set seed

set.seed(2018)

# Specifications of input parameters

# n:  number of observations
# p:  number of features for view 1
# q:  number of features for view 2
# ng: number of genes that influences the interested traits
# nt: number of traits influenced the truly associated genes


SimGen = function(n, p, q)  {

  # view 1 feature means

  mu_1 = rep(3, p)

  # view 1 feature std dev

  vars_1 = rep(0.1,p)

  sig_1 = diag(vars_1)

  # view 1 data matrix

  v1 = mvrnorm(n, mu_1, sig_1, tol = 1e-6,
               empirical = FALSE,
               EISPACK = FALSE)


  # indexing the genes that influences the traits
```

```
s1 = c(1:ng)   # first ng variables as target


# indexing the traits actually under influence

s2 = c(1:nt)    # first nt variables as influenced



# artificial effect size matrix, each row represents the
# coefficient vector for one of the influenced traits


eff = matrix(c(0,-1,1,-1,1,
               0,-1,-1,-1,1,
               1,0,1,-1,1,
               1,0,-1,1,-1,
               1,1,0,1,-1,
               1,1,0,1,-1,
               -1,-1,1,0,1,
               -1,-1,-1,0,-1,
               -1,1,1,-1,0,
               -1,1,-1,1,0
               ), nrow = nt, ncol = ng)


# background noise of traits domain

# view 2 feature means

mu_2 = rep(5, q)

# view 1 feature std dev

vars_2 = rep(0.1, q)
```

```
sig_2 = diag(vars_2)

# view 2 data matrix

v2 = mvrnorm(n, mu_2, sig_2, tol = 1e-6,
             empirical = FALSE,
             EISPACK = FALSE)


# replace influenced traits under mapping

for (j in s2){


  v2[,j] = v1[,s1] %*% eff[match(j,s2),]
  + rnorm(n, 0, 0.1)

}

# combining view 1 and view 2 for the multi-view dataset

  total_view = as.data.frame(cbind(view1, view2))

  # adding row ID

  total_view$ID = seq.int(nrow(total_view))

  # Renaming variables

  s1 -> index_v1
  s2 -> index_v2
  eff -> betas
  v1 -> view_1
  v2 -> view_2

  # Function Output
```

```
  return(out = lsit(total_view, n, p, q
                    index_v1, index_v2, v1, v2, betas))

}
```

B.2 Data Preparation For Evaluation of Stand-Alone CCA

```
# -----Generating simulated multi-view data -----
# -----using ‘‘SimGen’’ function-----

# Specifications

# n_1 = 100 <- sample size
# m_1 = 1000 <- dimension of feature vector for view 1
# m_2 = 10 <- dimension of feature vector for view 2
# ng_1 = 3 <- number of target genes for view 1
# nt_1 = 2 <- number of influenced traits for view 2

set.seed(2018)

# generating simulated data with pre-specified parameters
# inserted random values for cluster 2 parameters
# as we only capture the cluster 1 for stand-alone CCA

sim = simgen(100, 1000 , 10)

# capture cluster 1 output for view 1 and view 2 data

v1 = as.data.frame(sim[7])
v2 = as.data.frame(sim[8])

# index of target genes and influenced traits

s1 = c(1:10)
s2 = c(1:5)

# the mapping cofficients matrix
```

```
betas = sim[[9]]

# View 1 matrix
v1

# View 2 matrix
v2

# dimension check

dim(v1)

dim(v2)
```

B.3 Evaluation of Regularized CCA - Shrinkage Regularization

```
#---- rCCA with shrinkage regularization-----

# This evaluation requires R- package ‘‘mixOmics’’
# and installation of XQuartz app in Mac OS
# or X11 in Windows

require(mixOmics)


start.time = Sys.time()


# Output of Regularized CCA

rcca_out = rcc(v1, v2, method = "shrinkage")

rcca_out

# Obtained canonical correlations
rcca_cor = rcca_out$cor

rcca_cor

# Obtained first degree projection vectors (loadings)

x_load = rcca_out$loadings$X[,1]
y_load = rcca_out$loadings$Y[,1]

# recall target genes and influenced traits

s1
```

```
s2

# plot layout specification

par(mfrow=c(1,2))

# Plot of view 1 loading with target genes
# marked in red

x = seq(1,1000,1)

plot(x, x_load, pch = ifelse(x%in%c(1:10), 17, 1),col =
ifelse(x%in%c(1:10),  "red", "black"),
     cex = 0.8, xlab = "View 1 variables",
     ylab = "Canonical Loadings",
     main = "Canonical loadings of View 1 Variables")

abline(h = 0, col = "green")


# Plot of absolute value of view 1 loading, target genes
# marked in blue

y = seq(1,10,1)

plot(y, y_load,pch = ifelse(x%in%c(1:5), 17, 1) ,col =
 ifelse(y%in%c(1:5),  "blue", "black"),
     cex = 0.8, xlab = "View 2 variables",
     ylab = "Canonical Loadings",
     main = "Canonical loadings of View 2 Variables")

abline(h = 0, col = "green")


end.time = Sys.time()
```

```
# -----Assessment of performance ------------

# view 1

# mean level of effect size of noise variable in view 1

mean_noise_1 = mean(abs(x_load[-s1]))

# return Distinctiveness of interactive variables

abs(x_load[s1])>mean_noise_1

# mean level of effect size of target variables in view 1

mean_int_1 = mean(abs(x_load[s1]))

# Degree of separation in View 1

dos_1 = mean_noise_1/mean_int_1

dos_1

any(x_load == 0)

# mean and standard deviation of noise variable in view 1

mean(x_load[-s1])

sd(x_load[-s1])


# view 2

# mean level of effect size of noise variable in view 2
```

```
mean_noise_2 = mean(abs(y_load[-s2]))

# return Distinctiveness of interactive variables

abs(y_load[s2])>mean_noise_2

# mean level of effect size of target variables in view 2

mean_int_2 = mean(abs(y_load[s2]))

# degree of separation in view 2

dos_2 = mean_noise_2/mean_int_2

any(y_load == 0)

# mean and standard deviation of noise variables in view 2

mean(y_load[-s2])

sd(y_load[-s2])

# running time

run.time = end.time-start.time
```

B.4 Evaluation of Regularized CCA - via Cross-Validation Regularization

```
#---- Regularized Canonical Correlation Analysis--------
#---- via Cross-Validation Regularization -----

require(CCA)

start.time = Sys.time()

# Cross-Validation Regularization via
# estim.regul() required for high dimensional data

# format - estim.regul(X, Y, grid1 = NULL,
# grid2 = NULL, plt = TRUE)

# grid: if NULL - grid1, grid2 vector use
# seq(0.001, 1, length = 5) as default otherwise specify
# grid values ie. c(0.01,0.5)

# plt: logic, whether the CV heatmap should be plotted

# Regularization parameters

# reg_par = estim.regul(v1, v2)

lam1 = reg_par$lambda1
lam2 = reg_par$lambda2


# Implement rCCA with previously obtained
# regularization parameters

rcca_out = rcc(v1, v2, lam1, lam2)
```

```
rcca_out

# Obtained canonical correlations

rcca_cor = rcca_out$cor
rcca_cor

# Obtained first degree projection vectors (loadings)

x_load = rcca_out$xcoef[,1]
y_load = rcca_out$ycoef[,1]

# recall target genes and influenced traits

s1

s2

# plot layout specification for visual inspection

par(mfrow=c(1,2))

# Plot of absolute value of view 1 loading, target genes
# marked in red

x = seq(1,1000,1)

plot(x, x_load, pch = ifelse(x%in%c(1:10), 17, 1),col =
 ifelse(x%in%c(1:10), "red", "black"),
     cex = 0.8, xlab = "View 1 variables",
     ylab = "Canonical Loadings",
     main = "Canonical loadings of View 1 Variables")

abline(h = 0, col = "green")
```

```
# Plot of absolute value of view 1 loading, target genes
# marked in blue

y = seq(1,10,1)

plot(y, y_load,pch = ifelse(x%in%c(1:5), 17, 1) ,col =
ifelse(y%in%c(1:5),   "blue", "black"),
     cex = 0.8, xlab = "View 2 variables",
     ylab = "Canonical Loadings",
     main = "Canonical loadings of View 2 Variables")

abline(h = 0, col = "green")


end.time = Sys.time()


# -----Assessment of performance ------------

# view 1

# mean level of effect size of noise variable in view 1

mean_noise_1 = mean(abs(x_load[-s1]))

# return Distinctiveness of interactive variables

abs(x_load[s1])>mean_noise_1

# mean level of effect size of target variables in view 1

mean_int_1 = mean(abs(x_load[s1]))

# Degree of separation in View 1

dos_1 = mean_noise_1/mean_int_1
```

```
dos_1

any(x_load == 0)

# mean and standard deviation of noise variable in view 1

mean(x_load[-s1])

sd(x_load[-s1])


# view 2

# mean level of effect size of noise variable in view 2

mean_noise_2 = mean(abs(y_load[-s2]))

# return Distinctiveness of interactive variables

abs(y_load[s2])>mean_noise_2

# mean level of effect size of target variables in view 2

mean_int_2 = mean(abs(y_load[s2]))

# degree of separation in view 2

dos_2 = mean_noise_2/mean_int_2

any(y_load == 0)

# mean and standard deviation of noise variables in view 2

mean(y_load[-s2])
```

```
sd(y_load[-s2])

# running time

run.time = end.time-start.time
```

B.5 Evaluation of Sparse CCA

```
require(PMA)

# timing starts

start.time = Sys.time()


# Sparse CCA output

sparse_out  = CCA(v1,v2,"standard","ordered",
                  standardize = TRUE)

#Obtained  canonical correaltions

sparse_cor = sparse_out$cors

sparse_cor

# Obtained first degree view 1 and view 2 loadings

x_load = sparse_out$u
y_load = sparse_out$v

# graph layout specification

par(mfrow=c(1,2))

# plot of absolute value of view 1 loading, target
# genes marked in red

x = seq(1,1000,1)
```

```
plot(x, x_load, col = ifelse(x %in% c(1:10),  "red", "black"),
     cex = 0.8, xlab = "Genes",
     ylab = "Canonical loadings",
     main = "Canonical loadings of View 1")

abline(h = 0, col = "green")


# plot of absolute value of view 2 loading, target
# genes marked in blue

y = seq(1,10,1)

plot(y, y_load, col = ifelse(y %in% c(1:5), "blue", "black"),
     cex = 0.8, xlab = "Traits",
     ylab = "Canonical loadings",
     main = "Canonical loadings of view 2")

# horizontal reference

abline(h = 0, col = "green")


#timing ends

end.time = Sys.time()


# -----Assessment of performance ------------

# view 1

# mean level of effect size of noise variable in view 1

mean_noise_1 = mean(abs(x_load[-s1]))
```

```
# return Distinctiveness of interactive variables

abs(x_load[s1])>mean_noise_1

# mean level of effect size of target variables in view 1

mean_int_1 = mean(abs(x_load[s1]))

# Degree of separation in View 1

dos_1 = mean_noise_1/mean_int_1

dos_1

any(x_load == 0)

# mean and standard deviation of noise variable in view 1

mean(x_load[-s1])

sd(x_load[-s1])


# view 2

# mean level of effect size of noise variable in view 2

mean_noise_2 = mean(abs(y_load[-s2]))

# return Distinctiveness of interactive variables

abs(y_load[s2])>mean_noise_2

# mean level of effect size of target variables in view 2

mean_int_2 = mean(abs(y_load[s2]))
```

```
# degree of separation in view 2

dos_2 = mean_noise_2/mean_int_2

any(y_load == 0)

# mean and standard deviation of noise variables in view 2

mean(y_load[-s2])

sd(y_load[-s2])

# running time

run.time = end.time-start.time
```

B.6 Data Preparation for Evaluation of Correlation Clustering

```
#----------Simulated data generation using SimGen() ---------------

# Simulated data generation engine for Cluster 1

# A Multi-view data matrix generator
# using MVN(mu, sig)


# MASS package
require(MASS)

# Specifications of input parameters

# n <- number of observations

# p <-  number of features for view 1
# q<-  number of features for view 2

# n_gene <- number of genes that influences
# the traits been studied

# n_trait number of traits actually influenced
# the selected genes


simgen = function(n, p, q)  {

  # renaming variables

  ng1 = 10
```

```
nt1 = 5

# view 1 feature means
mu_1 = rep(3, p)        # fix mu

# view 1 feature std dev

vars_1 = rep(0.1,p)

sig_1 = diag(vars_1)

# view 1 matrix

v1 = mvrnorm(n, mu_1, sig_1, tol = 1e-6,
             empirical = FALSE,
             EISPACK = FALSE)


# indexing the genes that influences the traits

s1 = c(1:ng1)   # first ng1 variables as target


# indexing the traits actually under influence

s2 = c(1:nt1)    # first nt1 variables as influenced


# artificial effect size matrix,

# each row represents the coefficient vector for one of

# the influenced traits


eff = matrix(c(0,-1,1,-1,1,
```

```
               0,-1,-1,-1,1,
               1,0,1,-1,1,
               1,0,-1,1,-1,
               1,1,0,1,-1,
               1,1,0,1,-1,
               -1,-1,1,0,1,
               -1,-1,-1,0,-1,
               -1,1,1,-1,0,
               -1,1,-1,1,0
), nrow = nt1, ncol = ng1)


# background noise of traits domiain

# view 2 feature means
mu_2 = rep(5, q)

# view 1 feature std dev

vars_2 = rep(0.1, q)

sig_2 = diag(vars_2)

# view 1 matrix

v2 = mvrnorm(n, mu_2, sig_2, tol = 1e-6,
             empirical = FALSE,
             EISPACK = FALSE)


# replace influenced traits under mapping

for (j in s2){


  v2[,j] = v1[,s1] %*% eff[match(j,s2),]
```

```
    + rnorm(n, 0, 0.1)

  }

  # capturing output

  view1 = v1
  view2 = v2

  total_view = as.data.frame(cbind(view1, view2))

  # adding ID column to each row

  total_view$ID = seq.int(nrow(total_view))

  # renaming variables

  s1 -> index_v1
  s2 -> index_v2
  eff -> betas

  # Function Output

  return(out = list(total_view, n, p, q,
                    index_v1, index_v2, view1, view2, betas))

}

# Data generation engine for Cluster 2


simgen_2 = function(n, p, q)  {

  # renaming variables

  ng1 = 10
```

```
nt1 = 5


# view 1 feature means
mu_1 = rep(3, p)        # fix mu

# view 1 feature std dev

vars_1 = rep(0.1,p)

sig_1 = diag(vars_1)

# view 1 matrix

v1 = mvrnorm(n, mu_1, sig_1, tol = 1e-6,
             empirical = FALSE,
             EISPACK = FALSE)


# indexing the genes that influences the traits

s1 = c((p-ng1+1):p)   # last ng1 variables as target


# indexing the traits actually under influence

s2 = c((q-nt1+1):q)    # last nt1 variables as influenced


# artificial effect size matrix,

# each row represents the coefficient vector for one of

# the influenced traits
```

```
eff = matrix(c(1,-1,1,-1,0,
               -1,1,1,-1,0,
               1,-1,1,0,1,
               -1,1,1,0,-1,
               1,-1,0,-1,1,
               -1,1,0,-1,-1,
               1,0,-1,1,1,
               -1,0,-1,1,-1,
               0,-1,-1,1,1,
               0,1,-1,1,-1

), nrow = nt1, ncol = ng1)


# background noise of traits domiain

# view 2 feature means
mu_2 = rep(5, q)

# view 1 feature std dev

vars_2 = rep(0.1, q)

sig_2 = diag(vars_2)

# view 1 matrix

v2 = mvrnorm(n, mu_2, sig_2, tol = 1e-6,
             empirical = FALSE,
             EISPACK = FALSE)


# replace influenced traits under mapping

for (j in s2){
```

```
    v2[,j] = v1[,s1] %*% eff[match(j,s2),]
    + rnorm(n, 0, 0.1)

  }

  # capturing output

  view1 = v1
  view2 = v2

  total_view = as.data.frame(cbind(view1, view2))

  # adding ID column to each row

  total_view$ID = seq.int(nrow(total_view))

  # renaming variables

  s1 -> index_v1
  s2 -> index_v2
  eff -> betas

  # Function Output

  return(out = list(total_view, n, p, q
                    index_v1, index_v2, view1, view2, betas))

}



# Data generation for a two cluster multi-view dataset

set.seed(2018)
```

```
# Specifying multi-view dataset dimensions

n = 100
p = 1000
q = 10

ng = 10
nt = 5

# intrinsic cluster 1

sim_1 = simgen(n, p , q)

# capturing output view_1, view_2 matrices for cluster 1

v1_1 = as.data.frame(sim_1[7])
v2_1 = as.data.frame(sim_1[8])

mv_1 = cbind(v1_1, v2_1)

# capture the indices of target genes and influenced traits

s1_1 = c(1:ng)

s2_1 = c(1:nt)

# intrinsic cluster 2

sim_2 = simgen_2(n, p , q)

# capture output view_1, view_2 matrices for cluster 2

v1_2 = as.data.frame(sim_2[7])
v2_2 = as.data.frame(sim_2[8])
```

```
mv_2 = cbind(v1_2, v2_2)

# capture the indices of target genes and influenced traits

s1_2 = c((p-ng+1):q)   # last ng1 variables as target

s2_2 = c((q-nt+1):q)    # last nt1 variables as influenced


# combining two clusters to obtain simulated multi-view data

mvdata = as.data.frame(rbind(mv_1, mv_2))

# add tag for the intrinsic cluster of each row

mvdata[, "ID"] = c(1:nrow(mvdata))

# dimension check

dim(mvdata)
```

B.7 Evaluation of rCCA correlation clustering

```
#------ Evaluation of rCCA correlation clustering ---------

require(CCA)

# regularization via cross validation

# Extracting View 1 and View 2 matrices

   v1 = mvdata[, 1:1000]

   v2 = mvdata[, 1001:1010]

# Cross validation function

  reg_par = estim.regul(v1, v2)

# Regularization parameters

 lam1 = reg_par$lambda1
 lam2 = reg_par$lambda2

# rCCA correlation clustering engine

reg_clust = function(mvdata, p, q, k, iter){

  # The true clustering scheme

  truth = c(rep(1,100), rep(2,100))

  # creating array for error_rate

  error_rate = rep(0, iter)
```

```
mvdata$tg = sample(1:k, nrow(mvdata), replace = TRUE )

group = vector("list",length = k)

cca_out = vector("list",length = k)

U = vector(list, length = k)

V = vector(list, length = k)

slr_out = vector("list", length = k)

# the slope of V~U fit
a = vector("list", length = k)

# the intercept of V~U fit
b = vector("list", length = k)


for (i in c(1:iter)){

  for (j in c(1:k)){

    group[[j]] =  mvdata[which(mvdata$tag ==j), ]

    cca_out[[j]] =  rcc(group[[j]][,1:p], group[[j]][
    (p+1):(p+q)], lam1, lam2)

    U[[j]] = as.matrix(group[[j]][, 1:p]) %*%
    cca_out[[j]]$xcoef[,1]

    V[[j]] = as.matrix(group[[j]][,(p+1):(p+q)]) %*%
    cca_out[[j]]$ycoef[,1]

    slr_out[[j]] = lm(V[[j]] ~ U[[j]])
```

```
  b[[j]] =  as.numeric(cef(slr_out[[j]])[1])  # intercept

  a[[j]] =  as.numeric(cef(slr_out[[j]])[2]) # slope

}

for (id in c(1:nrow(mvdata))){

  item_v1 = mvdata[id,1:p]

  item_v2 = mvdata[id, (p+1):(p+q)]

  item_U = vector("list", length = k)

  item_V = vector("list", length = k)

  V_hat = vector("list", length = k)

  dist = vector("list", length = k)

  for (cl in c(1:k)){

    item_U[[cl]] = as.matrix(item_v1) %*%
    as.matrix( cca_out[[cl]]$xcoef[,1] )

    item_V[[cl]] = as.matrix(item_v2) %*%
    as.matrix( cca_out[[cl]]$ycef[,1] )

    V_hat[[cl]] = a[[cl]]%*%item_U[[cl]] + b[[cl]]

    dist[[cl]] = (V_hat[[cl]]- item_V[[cl]] )^2
```

```
      }

      new_tag_id = match( min(unlist(dist)), dist)

      mvdata[id, ncol(mvdata)] = new_tag_id

    } # reassign end

    label_out = table(mvdata$tag == truth)

    error_rate[i] = as.vector(label_out)[1]/nrow(mvdata)

  }   # iteration end

  return(list(mvdata, cca_out, mvdata$tag, error_rate))

}

# Specifications of function output

# mvdata - the original multi-view dataset
# cca_out - the local cca models on each cluster
# mvdata$tag - the resulted clustering scheme
# error_rate - the error rate of clustering scheme
```

B.8 Evaluation of sCCA correlation clustering

```
# Spase CCA clustering with optimized penalties


require(PMA)

sparse_clust_calipena = function(mvdata, p, q, k, iter){

 # the Truth clustering scheme

  truth = c(rep(1,100), rep(2,100))

  # creating array for error_rate

  error_rate = rep(0, iter)


  #1 randomly assign instances into k clusters

  mvdata$tag = sample(1:k, nrow(mvdata), replace = TRUE)


  group = vector("list",length = k)

  cca_out = vector("list",length = k)

  U = vector("list", length = k)   # canonical covariates

  V = vector("list", length = k)

  slr_out = vector("list", length = k)

  a = vector(list, length = k)  # the slope of V~U fit
  b = vector(list, length = k)  # the intercept of V~U fit
```

```
for (i in c(1:iter)){

  for(j in c(1:k)){


    group[[j]] =  mvdata[which(mvdata$tag ==j),  ]

    par = CCA.permute( group[[j]][,1:p], group[[j]][ ,(p+1):(p+q)] )

    cca_out[[j]] =  CCA(group[[j]][,1:p],group[[j]][,(p+1):(p+q)]
                        penaltyx = par$bestpenaltyx,
                        penaltyz = par$bestpenaltyz,
                        standardize = TRUE )

    U[[j]] = as.matrix(group[[j]][ , 1:p]) %*% cca_out[[j]]$u

    V[[j]] = as.matrix(group[[j][[ (p+1):(p+q)]) %*% cca_out[[j]]$v

    slr_out[[j]] = lm(V[[j]] ~ U[[j]])

    b[[j]] =  as.numeric(coef(slr_out[[j]])[1])  # intercept

    a[[j]] =  as.numeric(coef(slr_out[[j]])[2])  # slope


  }

  # reassignment

  for (id in c(1:nrow(mvdata))){

    item_v1 = mvdata[id, 1:p]

    item_v2 = mvdata[id, (p+1):(p+q)]
```

```
    V_hat = vector("list", length = k)

    item_U = vector("list", length = k)

    item_V = vector("list", length = k)

    dist = vector("list", length = k)


    for (cl in c(1:k)){

      item_U[[cl]] = as.matrix(item_v1) %*% cca_out[[cl]]$u

      item_V[[cl]] = as.matrix(item_v2) %*% cca_out[[cl]]$v

      V_hat[[cl]] = a[[cl]]%*%item_U[[cl]] + b[[cl]]

      dist[[cl]] = (V_hat[[cl]]- item_V[[cl]] )^2

    }

    new_tag_id = match( min(unlist(dist)), dist)

    mvdata[id, ncol(mvdata)] = new_tag_id


  } # reassignment end

  label_out = table(mvdata$tag = truth)

  error_rate[i] = as.vector(label_out)[1]/nrow(mvdata)


} # iteration end
```

```
  return(list(mvdata, cca_out, mvdata$tag, error_rate))

} # function end


# Specifications of function output

# mvdata - the original multi-view dataset
# cca_out - the local cca models on each cluster
# mvdata$tag - the resulted clustering scheme
# error_rate - the error rate of clustering scheme
```

Appendix C

Supplementary Figures

C.1 Histograms of SIF variables

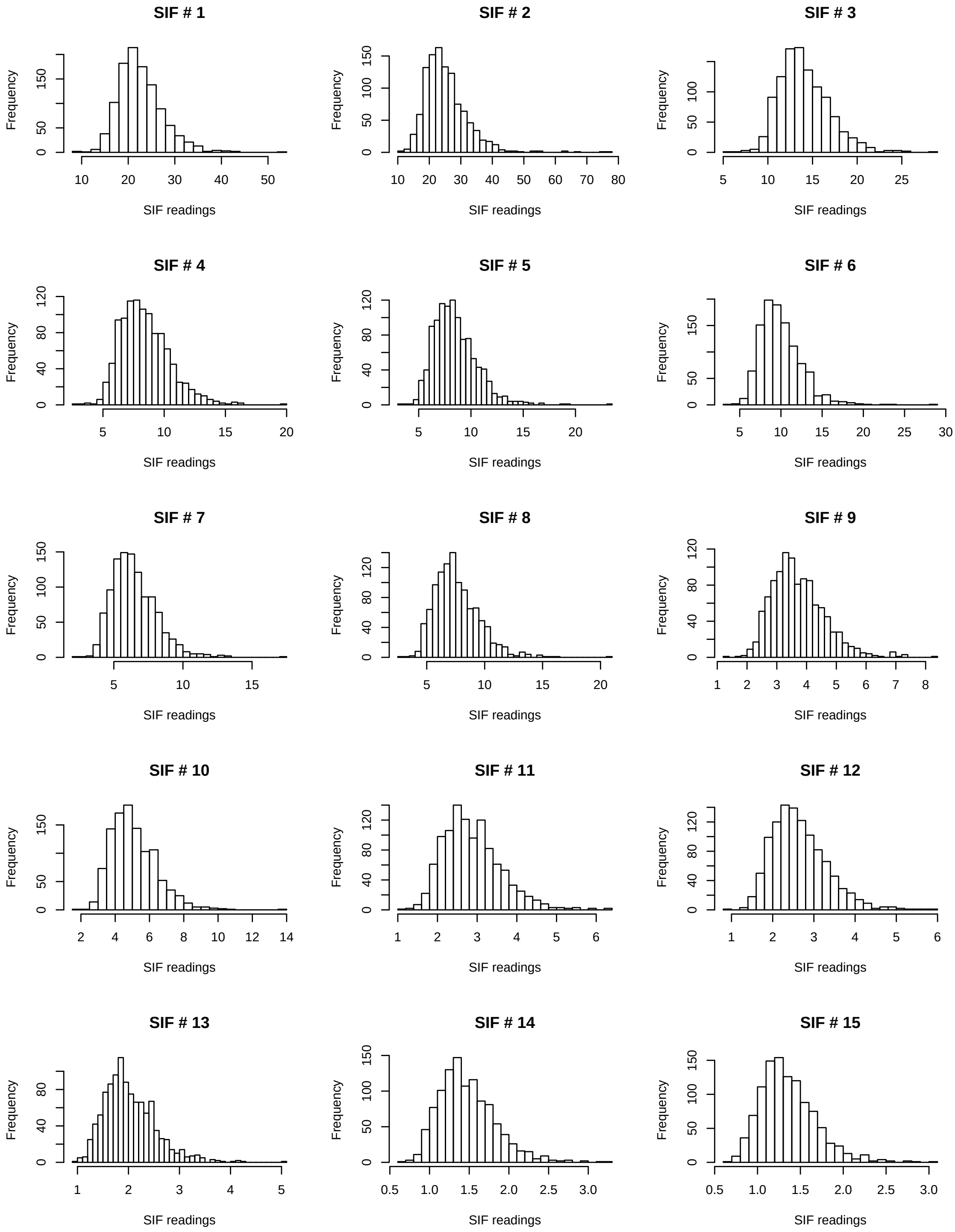

Figure C.1: The histograms of SIF variables

References

Ben-Dor, A., Shamir, R., & Yakhini, Z. (1999). Clustering gene expression patterns. *Journal of Computational Biology*, *6*(3-4), 281–297.

Canty, A. J., Srestha, N. M., Sylvestre, M.-P., Paterson, A. D., Boright, A. P., Brunzell, J. D., & Bull, S. B. (2016). A genome-wide association study for lipoprotein profiles using an empirically fitted null distribution. *Manuscript*.

Chen, X., Han, L., & Carbonell, J. (2012). Structured sparse canonical correlation analysis. In *Artificial intelligence and statistics* (pp. 199–207).

Chi, E. C., Allen, G. I., Zhou, H., Kohannim, O., Lange, K., & Thompson, P. M. (2013). Imaging genetics via sparse canonical correlation analysis. *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on*, 740–743.

Chu, D., Liao, L., Ng, M., & Zhang, X. (2013b). Sparse kernel canonical correlation analysis. In *Proceedings of international multiconference of engineers and computer scientists.*

Chu, D., Liao, L.-Z., Ng, M. K., & Zhang, X. (2013a). Sparse canonical correlation analysis: New formulation and algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(12), 3050–3065.

DCCT/EDIC Research Group. (2014). The diabetes control and complications trial/epidemiology of diabetes interventions and complications study at 30 years: Overview. *Diabetes Care*, *37*(1), 9–16.

D'haeseleer, P. (2005). How does gene expression clustering work? *Nature Biotechnology*, *23*(12), 1499.

Dudoit, S., Fridlyand, J., & Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, *97*(457), 77–87.

Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, *95*(25), 14863–14868.

Eny, K. M., Lutgers, H. L., Maynard, J., Klein, B. E. K., Lee, K. E., Atzmon, G., ... Paterson, A. D. (2014). GWAS identifies a NAT2 acetylator status tag single nucleotide polymorphism to be a major locus for skin fluorescence. *Diabetologia*, *57*(8), 1623–1634.

Fern, X. Z., Brodley, C. E., & Friedl, M. A. (2005). Correlation clustering for learning mixtures of canonical correlation models. In *Proceedings of the 2005 SIAM International Conference on Data Mining* (pp. 439–448).

Friedman, J. (1989). Regularized discriminant analysis. In *Journal of the American*

Statistical Association (pp. 165–175).

Gamazon, E. R., Wheeler, H. E., Shah, K. P., Mozaffari, S. V., Aquino-Michaels, K., Carroll, R. J., ... Im, H. K. (2015). A gene-based association method for mapping traits using reference transcriptome data. *Nature Genetics*, *47*(9), 1091.

Genz, A., & Bretz, F. (2009). *Computation of multivariate normal and t probabilities.* Heidelberg: Springer-Verlag.

González, I., Déjean, S., Martin, P. G., Baccini, A., et al. (2008). CCA: An R package to extend canonical correlation analysis. *Journal of Statistical Software*, *23*(12), 1–14.

González, I., & Djean, S. (2012). CCA: Canonical correlation analysis [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=CCA` (R package version 1.2)

Guo, Y., Hastie, T., & Tibshirani, R. (2006). Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, *8*(1), 86–100.

Hardoon, D. R., & Shawe-Taylor, J. (2011). Sparse canonical correlation analysis. *Machine Learning*, *83*(3), 331–353.

Herrero, J., Valencia, A., & Dopazo, J. (2001). A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics*, *17*(2), 126–136.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, *28*(3/4), 321–377.

Hull, E. L., Matter, N. I., Olson, B. P., Ediger, M. N., Magee, A. J., Way, J. F., ... Maynard, J. D. (2014). Noninvasive skin fluorescence spectroscopy for detection of abnormal glucose tolerance. *Journal of Clinical & Translational Endocrinology*, *1*(3), 92–99.

Jiang, D., Tang, C., & Zhang, A. (2004). Cluster analysis for gene expression data: a survey. *IEEE Transactions on Knowledge and Data Engineering*, *16*(11), 1370–1386.

Johnson, R. C., Nelson, G. W., Troyer, J. L., Lautenberger, J. A., Kessing, B. D., Winkler, C. A., & O'Brien, S. J. (2010). Accounting for multiple comparisons in a genome-wide association study (GWAS). *BMC Genomics*, *11*(1), 724.

Ledoit, O., & Wolf, M. (2003). Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, *10*(5), 603–621.

Lei, E., Miller, K., & Dubrawski, A. (2017). Learning mixtures of multi-output regression models by correlation clustering for multi-view data. *arXiv preprint arXiv:1709.05602*.

Leurgans, S. E., Moyeed, R. A., & Silverman, B. W. (1993). Canonical correlation analysis when the data are curves. *Journal of the Royal Statistical Society. Series B (Methodological)*, 725–740.

Li, B., Verma, S., Veturi, Y., Verma, A., Bradford, Y., Haas, D., & Ritchie, M. (2018). Evaluation of predixcan for prioritizing gwas associations and predicting gene expression. In *Pacific symposium on biocomputing. pacific symposium on biocomputing* (Vol. 23, pp. 448–459).

Li, Y., Willer, C., Sanna, S., & Abecasis, G. (2009). Genotype imputation. *Annual Review of Genomics and Human Genetics*, *10*, 387–406.

Lin, D., Calhoun, V. D., & Wang, Y.-P. (2014). Correspondence between fmri and snp data by group sparse canonical correlation analysis. *Medical Image Analysis*, *18*(6), 891–902.

Marzetta, C., Foster, D., & Brunzell, J. (1989). Relationships between LDL density and kinetic heterogeneity in subjects with normolipidemia and familial combined hyperlipidemia using density gradient ultracentrifugation. *Journal of Lipid Research*, *30*(9), 1307–1317.

Miller, R. G. (1981). Normal univariate techniques. In *Simultaneous Statistical Inference* (pp. 37–108). Springer.

Noble, W. S. (2009). How does multiple testing correction work? *Nature Biotechnology*, *27*(12), 1135.

Parkhomenko, E., Tritchler, D., & Beyene, J. (2007). Genome-wide sparse canonical correlation of gene expression with genotypes. In *Bmc proceedings* (Vol. 1, p. S119).

Parkhomenko, E., Tritchler, D., & Beyene, J. (2009). Sparse canonical correlation analysis with application to genomic data integration. *Statistical Applications in Genetics and Molecular Biology*, *8*(1), 1–34.

Paterson, A. D., Waggott, D., Boright, A. P., Hosseini, S. M., Shen, E., Sylvestre, M.-P., … others (2009). A genome-wide association study identifies a novel major locus for glycemic control in type 1 diabetes, as measured by both hba1c and glucose. *Diabetes*.

Rockman, M. V., & Kruglyak, L. (2006). Genetics of global gene expression. *Nature Reviews Genetics*, *7*(11), 862.

Rohart, F., Gautier, B., Singh, A., & LeCao, K.-A. (2017). mixomics: An R package for omics feature selection and multiple data integration. *PLoS Computational Biology*, *13*(11), e1005752. Retrieved from `http://www.mixOmics.org`

Roshandel, D., Klein, R., Klein, B. E., Wolffenbuttel, B. H., van der Klauw, M. M., van Vliet-Ostaptchouk, J. V., … Paterson, A. D. (2016). A new locus for skin intrinsic fluorescence in type 1 diabetes also associated with blood and skin glycated proteins. *Diabetes*, db151484.

Schäfer, J., & Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, *4*(1).

Subramanian, V., Chidester, B., Ma, J., & Do, M. N. (2018). Correlating cellular features with gene expression using cca. In *Biomedical imaging (isbi 2018), 2018 ieee 15th international symposium on* (pp. 805–808).

Sun, J., Lu, J., Xu, T., & Bi, J. (2015). Multi-view sparse co-clustering via proximal alternating linearized minimization. In *International conference on machine learning* (pp. 757–766).

Tibshirani, R., Hastie, T., Narasimhan, B., & Chu, G. (2003). Class prediction by nearest shrunken centroids, with applications to DNA microarrays. *Statistical Science*, 104–117.

Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with S* (Fourth ed.).

New York: Springer. Retrieved from http://www.stats.ox.ac.uk/pub/MASS4 (ISBN 0-387-95457-0)

Vinod, H. D. (1976). Canonical ridge and econometrics of joint production. *Journal of Econometrics*, *4*(2), 147–166.

Waaijenborg, S., de Witt Hamer, P. C. V., & Zwinderman, A. H. (2008). Quantifying the association between gene expressions and dna-markers by penalized canonical correlation analysis. *Statistical Applications in Genetics and Molecular Biology*, *7*(1).

Wiesel, A., Kliger, M., & Hero III, A. O. (2008). A greedy approach to sparse canonical correlation analysis. *arXiv preprint arXiv:0801.2748*.

Witten, D., & Tibshirani, R. (2009). Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical Applications in Genetics and Molecular Biology*, *8*(1), 1–27.

Witten, D., Tibshirani, R., Gross, S., & Narasimhan, B. (2018). PMA: Penalized multivariate analysis [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=PMA (R package version 1.0.11)

Witten, D., Tibshirani, R., & Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, *10*(3), 515–534.

Yeung, K. Y., & Ruzzo, W. L. (2001). Principal component analysis for clustering gene expression data. *Bioinformatics*, *17*(9), 763–774.