<Sparse Statistical Methods for Data Integration>

CONTRIBUTIONS TO SPARSE STATISTICAL METHODS FOR DATA INTEGRATION

BY

ASHLEY JOEL BONNER,

B.Sc., M.Sc.

A THESIS
SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY

MCMASTER UNIVERSITY

McMaster University DOCTOR OF PHILOSOPHY (2018)

Hamilton, Ontario (Health Research Methodology)

TITLE: Contributions to Sparse Statistical Methods for Data Integration

AUTHOR: Ashley Joel Bonner,
B.Sc. (McMaster University),
M.Sc. (McMaster University)

SUPERVISOR: Dr. Joseph Beyene

PAGES: xvii, 118

Lay Abstract:

Due to rapid advances in technology, many areas of scientific research are measuring multiple sources of massive, complex, and diverse data in hopes to better understand the principles underpinning puzzling phenomena. Now, more than ever, advancement and discovery relies upon sophisticated and robust statistical and computational methods that reduce the data complexity, harness variability, and integrate multiple sources of information. In this thesis, I test and validate the 'sparse' class of multivariate statistical methods that is becoming a promising, fresh solution to these data-driven challenges. Using publicly available data from genetic toxicology as motivation, I demonstrate the utility of these methods, find where they work best, and explore the possibility of improving their scientific interpretability. The work in this thesis contributes to both biostatistics and genomic literature, by meshing together rigorous statistical methodology with real-world data applications.

Abstract:

Background: Scientists are measuring multiple sources of massive, complex, and diverse data in hopes to better understand the principles underpinning complex phenomena. Sophisticated statistical and computational methods that reduce data complexity, harness variability, and integrate multiple sources of information are required. The 'sparse' class of multivariate statistical methods is becoming a promising solution to these data-driven challenges, but lacks application, testing, and development.

Methods: In this thesis, efforts are three-fold. Sparse principal component analysis (sparse PCA) and sparse canonical correlation analysis (sparse CCA) are applied to a large toxicogenomic database to uncover candidate genes associated with drug toxicity. Extensive simulations are conducted to test and compare the performance of many sparse CCA methods, determining which methods are most accurate under a variety of realistic, large-data scenarios. Finally, the performance of the non-parametric bootstrap is examined, determining its ability to generate inferential measures for sparse CCA.

Results: Through applications, several groups of candidate genes are obtained to point researchers towards promising genetic profiles of drug toxicity. Simulations expose one sparse CCA method that outperforms the rest in the majority of data scenarios, while suggesting the use of a combination of complimentary sparse CCA methods for specific data conditions. Simulations for the bootstrap conclude the bootstrap to be a suitable means for inference for the canonical correlation coefficient for sparse CCA but only when sample size approaches the number of variables. As well, it is shown that aggregating sparse CCA results from many bootstrap samples can improve accuracy of detection of truly cross-correlated features.

Conclusions: Sparse multivariate methods can flexibly handle challenging integrative analysis tasks. Work in this thesis has demonstrated their much-needed utility in the field of toxicogenomics and strengthened our knowledge about how they perform within a complex, massive data framework, while promoting the use of bootstrapped inferential measures.

Acknowledgements:

Many people supported me during my education and work for this thesis.

I would like to thank my supervisor, Dr. Joseph Beyene, who paved the way for me to embark on many new educational adventures. His encouragement and guidance was extremely helpful and I could not have done this without him.

I would like to thank my committee members, Dr. Jemila Hamid and Dr. Angelo Canty, for providing excellent feedback and encouragement along the way, as well as my mentor Dr. Gordon Guyatt, who taught me invaluable lessons in writing and organization.

My journey would not have been possible without the comradery of my fellow students from the Statistics for Integrative Genomics and Methods Advancement (SIGMA) group and beyond. Sathish Pichika, Taddele Kibret, Zelalem Firisa Negeri, Mateen Shaikh, Binod Neupane, Ahmed Hossain, Shofique Islam, Regina Kampo, Joycelyne Ewusie, Paul Alexander, Akram Alyass, et al.

Lastly, but most importantly, I could not have done this without the support of my selfless wife, Stephanie Santamaria, and loving parents, Linda and Mark Bonner, who gave me the much needed encouragement to persevere.

Table of Contents:

List of Tables:

List of Figures:

List of Abbreviations and Symbols:

General:

PCA:	principal component analysis
PC:	principal component
CCA:	canonical correlation analysis
CV:	canonical variate
EVD:	eigen-value decomposition
SVD:	singular-value decomposition
PLS:	partial least-squares
OLS:	ordinary least squares
LASSO:	least absolute shrinkage and selection operator
GAW:	Genetic Analysis Workshop
CAMDA:	Critical Assessment of Massive Data Analysis
TGP:	Japanese Toxicogenomics Project
DEG:	differentially expressed gene
DEPC:	differentially expressed principal component
DILI:	drug-induced liver injury
FARMS:	factor analysis for robust microarray summarization
SNP:	single-nucleotide polymorphism

Declaration of Academic Achievement:

I was the main contributor and first author for all published manuscripts, drafted manuscripts, and all other content contained within this thesis. An account of co-author contributions is listed in the preface. Apart from the manuscripts contained within this thesis, my supervisory committee, consisting of Drs. Joseph Beyene, Jemila Hamid, and Angelo Canty, provided critical feedback for content within this thesis (Chapters 1 through 5).

Preface:

This sandwich-style thesis includes five chapters. Chapter 1 provides an introduction to the research topic covered. Chapters 2, 3, and 4, include ***three manuscripts*** that constitute the bulk of work presented in this thesis. I am the first author and primary contributor for each of these manuscripts. Below are brief descriptions for the three manuscripts that form the basis of this thesis. Chapter 5 summarizes the work and contributions, as well as highlights future work and conclusions. All work was completed between September, 2012 and October, 2018.

Manuscript 1 (Chapter 2): This published paper includes a novel application of sparse principal component analysis to real toxicogenomic data. I conceived the idea, developed the methods, coded and conducted analysis, transcribed results, and drafted the manuscript. The manuscript was published in a peer-reviewed journal called 'Systems Biomedicine' in 2014. I had one co-author: Dr. Joseph Beyene (supervisor). Dr. Beyene guided me to fine-tune the methods and results sections, and to edit writing in the manuscript.

Manuscript 2 (Chapter 3): This manuscript includes an extensive simulation study that identifies sparse canonical correlation analysis methods that best perform in high-dimensional data, and an analysis of real toxicogenomic data. I conceived the idea and methods, designed and coded the simulation experiments and real-data analysis, summarized the results, and drafted the manuscript. The manuscript has been prepared for submission to a peer-reviewed journal. I currently have three co-authors in the following order: Dr. Jemila Hamid (committee member), Dr. Angelo Canty (committee member), and Dr. Joseph Beyene (supervisor). Dr. Hamid provided critical feedback regarding methods and analysis, and helped to revise the manuscript. Dr. Canty provided guidance regarding methods and reviewed the manuscript. Dr. Beyene helped improve the methods section and to edit writing in the manuscript.

Manuscript 3 (Chapter 4): This manuscript includes an extensive simulation study that investigates the performance of the non-parametric bootstrap with sparse canonical correlation analysis. I conceived the idea and methods, designed and coded the simulation experiments, summarized the results, and drafted the writing. The manuscript has been prepared for submission to a peer-reviewed journal. I have three co-authors in the following order: Dr. Angelo Canty (committee member), Dr. Jemila Hamid (committee member), and Dr. Joseph Beyene (supervisor). Dr. Hamid provided critical feedback regarding methods and results, and helped revise the manuscript. Dr.

Canty provided guidance regarding methods and reviewed the manuscript. Dr. Beyene helped improve the methods section and to edit writing in the manuscript.

CHAPTER 1

1. INTRODUCTION

1.1. Big data integration in health research

The continued development of technology allows us to measure and store data at a growing rate and capacity in both academic and private sectors alike.[1] From marketing to financial transactions, social media to mobile downloads, and from the high-resolution imaging of the brain to the microscopic examination of the human genome, the flow of data has become prolific.[2]

The health research sector has experienced this surge in data acquisition at a particularly rapid pace.[3,4] Tools in the clinical setting have evolved to automate the acquisition of patient information. The development of biotechnologies, such as the DNA microarray, mass-spectrometry, and next-generation sequencing, has sparked a genomic revolution by allowing the measurement of genetic activity with remarkable precision and granularity.[5]

We can now detect genes ("genomics"), mRNA ("transcriptomics"), proteins ("proteomics"), and metabolites ("metabolomics"), providing a holistic view of an organism's biological system.[6] Investigators studying diseases, conditions, and phenomena of all kinds are simultaneously capturing 'omic' data sources like these alongside more conventional data (e.g., clinical assessments, demographics). By bringing together complementary views, it is hypothesized that we can extract new insights into the biological mechanisms underpinning complex conditions such as cancers, neurodevelopmental disorders, diabetes, and gene-diet interactions.[4,7]

Though a marvel on its own, the measurement of such data is just the first step to generating new knowledge. Extracting succinct, comprehensible output from a sea of complex data requires thoughtful analytic work and the right tools to do-so. As such, the challenging task of simultaneously analyzing multiple data types has been given the spotlight. The undertakings to tackle this challenging task can be identified through both publications and larger collaborative initiatives, indexed by popularized terms such as 'data integration', 'data fusion', and 'integrative analysis'.[8–10]

To explore common approaches and illustrate activity in this area, I searched the OVID Medline database for published articles that were fundamentally concerned with the integration of omic data (see Appendix A for a description of my search strategy). Figure 1 displays the number of publications found per year, updated September 14th, 2018. The search returned 1858 publications between the years 1998 and 2018, of

which 931 (50%) occurred from 2015 onward. With continued technological advancement and the ever pressing demand for data-driven breakthroughs, I expect this trend to continue for years to come.

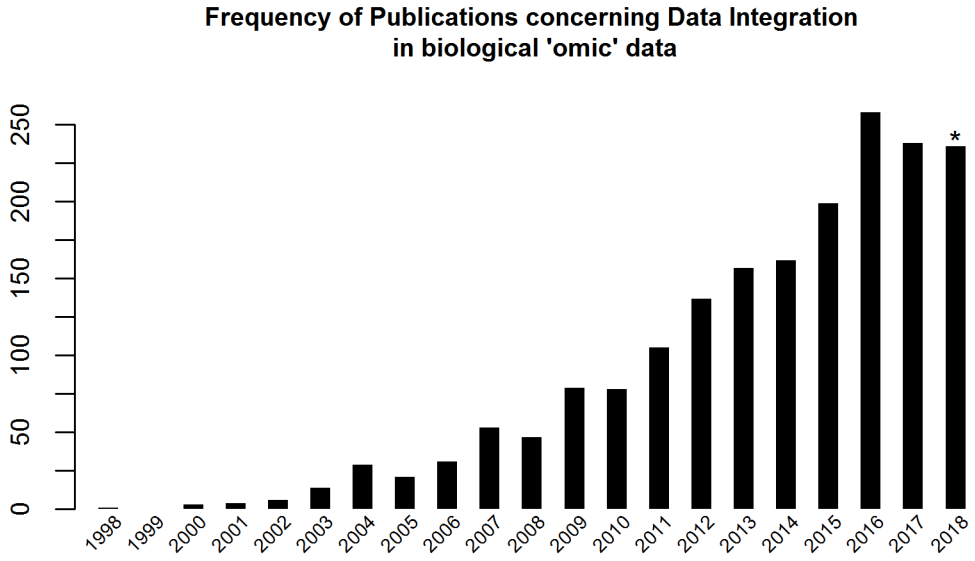


*Figure 1: A bar plot displaying the number of publications per year retrieved by my search of OVID Medline (documented in Appendix). *Since the search was last updated September 14th, 2018, the frequency for the year 2018 is expected to be larger by the end of the year.*

Skimming through articles, one quickly gains an appreciation for the cross-disciplinary skillsets required to understand, process, integrate, and analyze massive omic data. Collaboration between a variety of health experts and statisticians, biostatisticians, bioinformaticians, and computer scientists is a more crucial requirement than ever before. Efforts have gone beyond in-house team building. The sharing of data and assembly of conferences and workshops have profoundly impacted the collaboration landscape and greatly facilitated my learning and work within this thesis.

Data sharing projects have promoted secondary research and maximized the return on investment for large-scale data capture initiatives. The Gene Expression Omnibus (GEO), for example, is a massive repository that centralizes both data submission and acquisition for a wide range of health contexts.[11] Databases like The

Cancer Genome Atlas (TCGA) provide a hub for more area-specific data mining (in this example, cancer).[12]

Conferences and workshops are also fostering cross-disciplinary learning, sometimes with publicly available omic data. To improve my knowledge, I participated in two conferences themed around 'analysis challenges'. Both the 2012 Genetic Analysis Workshop (GAW) and the 2013 International Conference on the Critical Assessment of Massive Data Analysis (CAMDA) attracted attendees to share innovative analyses of challenge datasets. The integration of massive omic data with conventional clinical outcomes was the primary focus for both events.[13]

By reading data integration articles and engaging with real data, one can appreciate how diverse the term 'integrate' can be. There are many reasons and ways to integrate data. In 2009, Hamid et al, proposed a conceptual framework for categorizing an omic data integration task by considering three important questions regarding the context from which it arises.[10] In short, the questions are: 1. *Why* are we integrating the data (i.e., what are we hoping to improve)? 2. *What* type of data are we integrating? 3. *When* are we integrating the data?

In this thesis, I narrow my focus to addressing issues involving a particular, yet omnipresent, setting for data integration found in health research. The choice of this particular setup was motivated due to the compelling health research questions involved, the state of the statistical methodology prepared to answer them (summarized in subsequent sections), and existing collaborations involving analysis of real data using the statistical methods that are the focus of this thesis. Figure 2 presents the data integration set up that I concentrate on hereafter.

The objective of data integration in this thesis is, therefore, to better detect meaningful relationships between omic data and clinical measures, with emphasis on finding complex relationships within and between data domains. From a health research perspective, the goal for integrating large sets of conventional, clinical measures and omic measures is to find deep connections between them that are more meaningful than simply observing relationships between two variables at a time.

The type of data considered in this thesis is heterogeneous, meaning that the multiple sources of data to be integrated have been measured on the *same* samples (individuals, participants, observations) but contain *different* types of variables (e.g., clinical vs. omic measurements). This contrasts the homogeneous data scenario, in which the multiple sources of data to be integrated are different samples (e.g., cohorts) with the same set of variables.

The stage at which data integration is performed is intermediate, occurring intrinsically within the statistical methodology I use rather than by concatenating data prior to analysis (early) or aggregating analysis results (late). Though there are some benefits to intermediate integration, the stage is a consequence of the methods I have chosen rather than a deliberate choice.

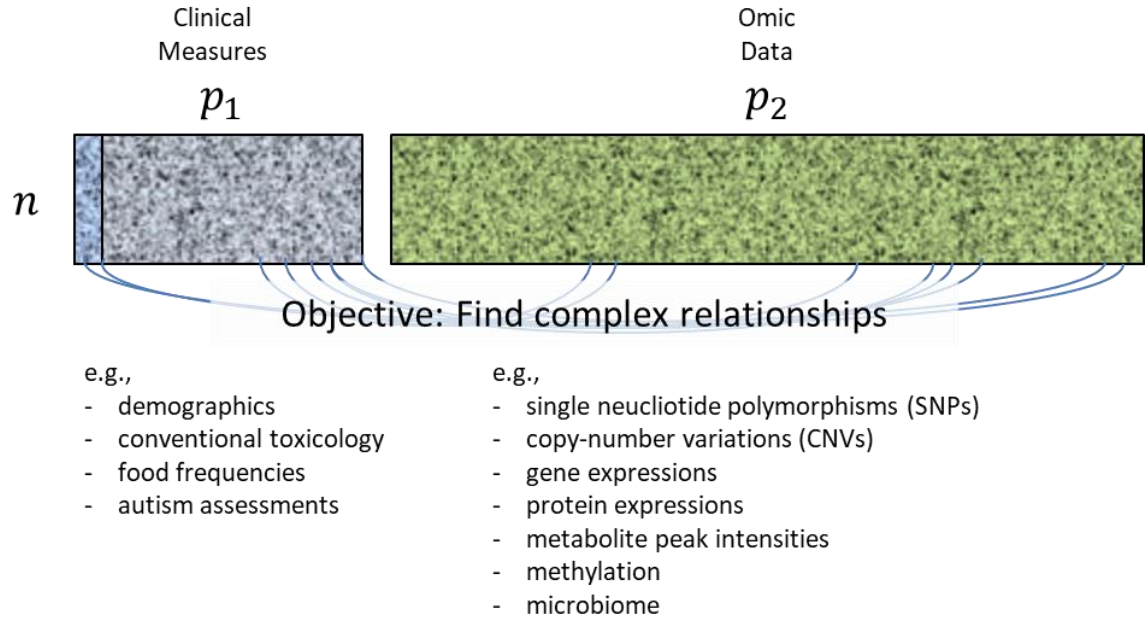


Figure 2: A depiction of the data integration scenario addressed within this thesis.

1.2. Statistical analysis challenges and possible statistical approaches

Detecting complex relationships between large heterogeneous data types with one or more set of omic data poses significant statistical challenges. Here I describe three significant technical hurdles to analyzing big data to justify and motivate the choice of a particular class of statistical methods from which the bulk of this thesis work is based.

The first major challenge is due to the *correlation between variables*. Phenotypes such as autism spectrum disorder, cancers, nutrition and diet, and drug toxicity are deemed 'complex' because they can be measured in a variety of ways and are affected by a combination of many genes and environmental factors at once.[14] Genetic processes are intricate by nature. Transcription and translation are measured from and modified by a multitude of factors. Therefore, both clinical/conventional data and omic data will tend to have correlation within and between variables and it is essential for a statistical analysis to harness it.

The fact that we capture so much correlated information proclaims an importance to use it jointly. The diagnosis of some conditions cannot be made without consulting a myriad of symptoms. For example, questionnaires to capture child behaviour or diet typically culminate in the calculation of composite scores as an attempt to classify cognitive function or level of nutrition. On the genetic end, individual biomarkers tend to contribute very small, hard to detect signals on their own and, with rare exception, work in concert.[15] Finding relationships within and between data via groups of variables or composite variables could increase probability of detection and better represent the underlying biological mechanisms.

Certain conventional multivariate methods are available to attempt such tasks, by parameterizing and estimating the correlation structure while searching for latent, combined effects. For example, principal component analysis (PCA)[16], partial least squares (PLS)[17], and canonical correlation analysis (CCA)[18] are classic multivariate methods that can investigate the variation within and between features.

The second major challenge is due to *the dimensions of the data*. Technologies used to measure molecular-level features are capable of doing-so at a high resolution. There has been movement away from candidate gene studies and toward whole-genome sequencing.[19] As a result, data will consist of thousands, up to millions of variables (p). However, such technologies are typically expensive to acquire and use, demanding a large budget from investigators. This limits the sample size (n) that is feasible for research projects; an issue that is accentuated when studying rare conditions without access to many participants anyways. This creates the 'small n, large p' scenario, which poses significant mathematical obstructions for multivariate approaches like PCA, PLS, and CCA, especially when $n < p$.[20] The term high-dimensional is often used to refer to this scenario, though it does not highlight the important small n aspect. Thankfully, regularization adaptations[21] to multivariate methods solve some of the mechanical challenges involved in estimation and allow them to operate as dimension reduction tools.[22]

The third major challenge is *variable selection and model interpretation*. With thousands of variables, it is extremely difficult to come up with accurate and concise multivariate models. As the number of variables in each data domain increases, the number of possible interactions becomes almost innumerable. Though able to estimate complex relationships, multivariate approaches tend to be poor at variable selection. Typically, they retain all variables in the estimated models. Amid thousands of variables, it is essential to adopt statistical methods that eliminate unimportant features entirely; separating the signal from the noise, so to speak.

A special subset of regularization approaches called 'sparse' methods have emerged within the last decade and are the focus and highlight of this thesis. Sparse adaptations of PCA, PLS, and CCA not only bring the dimension reduction qualities to the $n < p$ scenarios, but their unique sparsity mechanic simultaneously eliminates unimportant variables, producing models that express relationships between small subsets of the original sets of variables. This provides investigators superior insight compared to their non-sparse and often less accurate counterparts.

1.3. Sparse multivariate methods

In this thesis, I focus on the application, testing, and development of sparse PCA and sparse CCA; sparse regression approaches are also applied. In this section, I introduce the key aspects of these methods, including their generic mathematical infrastructure, and highlight the major developments preceding the work. There may be overlap between background methodology presented in this section and that within the manuscripts corresponding to my contributions, purely due to their need to stand alone as publishable material.

1.3.1. Conventional PCA:

Conventional PCA was introduced in 1901 by Karl Pearson.[23] It explores the latent multivariate structure of *one set of variables* and provides a transformative view of the data at a reduced dimension that can better articulate the information (variation) within.[16,24]

Let $\boldsymbol{X} = (X_1, X_2, \dots, X_p)'$ be a vector of p random variables. PCA first seeks a linear combination $Z_1 = \boldsymbol{v}_1'\boldsymbol{X}$ that, over all possible choices of the vector $\boldsymbol{v}_1 = (v_{11}, v_{12}, \dots, v_{1p})'$, has maximum variance $Var(\boldsymbol{v}_1'\boldsymbol{X})$. The objective function for this first step in PCA can be written as

$$\underset{\boldsymbol{v}_1}{\text{maximize}}\{\boldsymbol{v}_1'\Sigma\boldsymbol{v}_1\}$$
$$subject\ to\ \|\boldsymbol{v}_1\|_2^2 = 1,$$

where $\Sigma = Var(\boldsymbol{X})$, $\|.\|_l$ is the L_l-norm[†], and the added scaling constraint on $\boldsymbol{v}_1$ dictates that coefficients in the vector must be between -1 and 1. Upon solving this optimization problem, the resulting linear combination Z_1 is called the first *principal component (PC)* and the vector $\boldsymbol{v}_1$ is called the first *loading (coefficient, weight) vector*. The components

[†] For $\boldsymbol{a} = (a_1, a_2, \dots, a_b)'$, the L_l-norm is defined as: $\|\boldsymbol{a}\|_l = \sqrt[l]{|a_1|^l + |a_2|^l + \cdots + |a_b|^l}$

$v_{11}, v_{12}, \dots, v_{1p}$ of $\boldsymbol{v}_1$ are called *loadings* and, along with other measures, relay the extent to which each $X_1, X_2, \dots, X_p$ contribute to Z_1 (i.e., what Z_1 represents).

Subsequent PCs $Z_2, Z_3, \dots, Z_p$ and their loading vectors $\boldsymbol{v}_2, \boldsymbol{v}_3, \dots, \boldsymbol{v}_p$ can be obtained such that the PCs are orthogonal to Z_1 and one-another, and represent a monotone decreasing amount of unique variation in the data. The full set of PCs $\boldsymbol{Z} = (Z_1, Z_2, \dots, Z_p)'$ have a sum of variances equal to that of the original variables but, as is the goal of PCA, should be more heavily weighted towards the first few PCs. If this is the case, investigators may choose to keep only the first few PCs for analysis as they represent the majority of information in the data, hence the power of PCA as a dimension reduction tool.

Given an $n \times p$ matrix $\mathbb{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_p)$ representing n observations of the random vector $\boldsymbol{X}$, where $\boldsymbol{x}_j$ is the observed data for variable X_j (for $j = 1, \dots, p$), then the sample version of PCA can be conducted by solving the above optimization problem with sample covariance matrix $\hat{\Sigma}$ instead of Σ. Sample PCA returns estimated loading vectors $\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2, \dots, \hat{\boldsymbol{v}}_p$, where $\hat{\boldsymbol{v}}_j = (\hat{v}_{j1}, \hat{v}_{j2}, \dots, \hat{v}_{jp})'$ for $j = 1, \dots, p$, and their corresponding PCs $\mathbb{Z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_p)$, where $\boldsymbol{z}_j = \mathbb{X}\hat{\boldsymbol{v}}_j$ is the j^{th} PC.

Being deeply rooted in linear algebra, classic algorithms, including eigen-value decomposition (EVD) and singular value decomposition (SVD), can be used to conduct PCA on sample data and obtain full sets of estimated PCs and estimated loading vectors. Eigen vectors correspond to the loading vectors and the eigen values can be used to calculate the percentage of total variance explained by each PC.[24]

1.3.2. Conventional CCA:

Conventional CCA was introduced in 1936 by Harold Hotelling.[25] CCA has many of the same attributes that define PCA – it examines the latent multivariate structure and provides a unique view of the data at a reduced dimension. The two methods are also very closely related mathematically.[26] However, CCA aims to describe *correlation between two sets of variables*.[24]

Let $\boldsymbol{X}_1 = (X_{1,1}, X_{2,1}, \dots, X_{p_1,1})'$ be a vector of p_1 random variables and $\boldsymbol{X}_2 = (X_{1,2}, X_{2,2}, \dots, X_{p_2,2})'$ be a vector of p_2 random variables[‡]. CCA first seeks a *pair of*

[‡] PCA and CCA are closely related and extensions to CCA have been made to accommodate more than two sets of variables. In turn, I chose to reflect these facts in the notation via an additional layer of subscripts that differentiate sets of variables when moving to CCA. The data-defining subscripts are always placed at the end and are separated from others by a comma. For example: X_2 (from PCA; no commas) is the second random variable from the only possible set of variables $\boldsymbol{X}$, whereas $X_{1,2}$ (from CCA; comma and subscript at the end) is the first random variable from the second set of variables $\boldsymbol{X}_2$.

linear combinations $Z_{1,1} = \boldsymbol{w}_{1,1}'\boldsymbol{X}_1$ and $Z_{1,2} = \boldsymbol{w}_{1,2}'\boldsymbol{X}_2$ that, over all possible choices of vectors $\boldsymbol{w}_{1,1} = (w_{11,1}, w_{12,1}, \dots, w_{1p_1,1})'$ and $\boldsymbol{w}_{1,2} = (w_{11,2}, w_{12,2}, \dots, w_{1p_2,2})'$, have maximum correlation $Corr(\boldsymbol{w}_{1,1}'\boldsymbol{X}_1, \boldsymbol{w}_{1,2}'\boldsymbol{X}_2)$. The objective function for this first step in CCA can be written as

$$\underset{\boldsymbol{w}_{1,1}, \boldsymbol{w}_{1,2}}{\text{maximize}} \left\{ \frac{\boldsymbol{w}_{1,1}'\boldsymbol{\Sigma}_{12}\boldsymbol{w}_{1,2}}{\sqrt{\boldsymbol{w}_{1,1}'\boldsymbol{\Sigma}_{11}\boldsymbol{w}_{1,1}}\sqrt{\boldsymbol{w}_{1,2}'\boldsymbol{\Sigma}_{22}\boldsymbol{w}_{1,2}}} \right\}$$
$$subject\ to\ \left\|\boldsymbol{w}_{1,1}\right\|_2^2 = 1, \left\|\boldsymbol{w}_{1,2}\right\|_2^2 = 1,$$

where $\boldsymbol{\Sigma}_{11} = Var(\boldsymbol{X}_1)$, $\boldsymbol{\Sigma}_{22} = Var(\boldsymbol{X}_2)$, and $\boldsymbol{\Sigma}_{12} = Cov(\boldsymbol{X}_1, \boldsymbol{X}_2)$. Upon solving this optimization problem, the resulting linear combinations $Z_{1,1}$ and $Z_{1,2}$ are called the first *canonical variate (CV) pair* and have first *canonical correlation* $\rho_{1,12} = Corr(Z_{1,1}, Z_{1,2})$, and the vectors $\boldsymbol{w}_{1,1}$ and $\boldsymbol{w}_{1,2}$ are called the first *canonical (loading, coefficient, weight) vectors*. Similar to the loadings in PCA, the *loadings* $w_{11,1}, w_{12,1}, \dots, w_{1p_1,1}$ in $\boldsymbol{w}_{1,1}$ and $w_{11,2}, w_{12,2}, \dots, w_{1p_2,2}$ in $\boldsymbol{w}_{1,2}$ express the extent that each variable contributes to the first (highest) cross-correlation between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$.

Subsequent pairs of CVs $Z_{2,1}, Z_{3,1}, \dots, Z_{\min(p_1,p_2),1}$ and $Z_{2,2}, Z_{3,2}, \dots, Z_{\min(p_1,p_2),2}$, with their correlations $\rho_{2,12}, \rho_{3,12}, \dots, \rho_{\min(p_1,p_2),12}$, as well as corresponding canonical vectors $\boldsymbol{w}_{2,1}, \boldsymbol{w}_{3,1}, \dots, \boldsymbol{w}_{\min(p_1,p_2),1}$ and $\boldsymbol{w}_{2,2}, \boldsymbol{w}_{3,2}, \dots, \boldsymbol{w}_{\min(p_1,p_2),2}$, can be obtained such that the CVs are, respectively, orthogonal to $Z_{1,1}$ and $Z_{1,2}$ (and, respectively, one-another) and represent a monotone decreasing amount of unique cross-correlation in the data. From the full set of CVs $\boldsymbol{Z}_1 = (Z_{1,1}, Z_{2,1}, \dots, Z_{\min(p_1,p_2),1})'$ and $\boldsymbol{Z}_2 = (Z_{1,2}, Z_{2,2}, \dots, Z_{\min(p_1,p_2),2})'$, the investigator might elect to keep just the first few to inspect cross-correlations at a reduced dimension.

Given an $n \times p_1$ matrix $\mathbb{X}_1 = (\boldsymbol{x}_{1,1}, \boldsymbol{x}_{2,1}, \dots, \boldsymbol{x}_{p_1,1})$ representing n observations of the random vector $\boldsymbol{X}_1$, where $\boldsymbol{x}_{j_1,1}$ is the observed data for variable $X_{j_1,1}$ (for $j_1 = 1, \dots, p_1$), and an $n \times p_2$ matrix $\mathbb{X}_2 = (\boldsymbol{x}_{1,2}, \boldsymbol{x}_{2,2}, \dots, \boldsymbol{x}_{p_2,2})$ representing the same n observations but of the random vector $\boldsymbol{X}_2$, where $\boldsymbol{x}_{j_2,2}$ is the observed data for variable $X_{j_2,2}$ (for $j_2 = 1, \dots, p_2$), then the sample version of CCA can be conducted by solving the above optimization problem with sample covariance matrices $\hat{\Sigma}_{11}$, $\hat{\Sigma}_{22}$, and $\hat{\Sigma}_{12}$ instead of Σ_{11}, Σ_{22}, and Σ_{12}, respectively. Sample CCA returns estimated canonical vectors $\widehat{\boldsymbol{w}}_{1,1}, \widehat{\boldsymbol{w}}_{2,1}, \dots, \widehat{\boldsymbol{w}}_{\min(p_1,p_2),1}$ and $\widehat{\boldsymbol{w}}_{1,2}, \widehat{\boldsymbol{w}}_{2,2}, \dots, \widehat{\boldsymbol{w}}_{\min(p_1,p_2),2}$, where $\widehat{\boldsymbol{w}}_{m,1} = (\widehat{w}_{m1,1}, \widehat{w}_{m2,1}, \dots, \widehat{w}_{mp_1,1})'$ and $\widehat{\boldsymbol{w}}_{m,2} = (\widehat{w}_{m1,2}, \widehat{w}_{m2,2}, \dots, \widehat{w}_{mp_2,2})'$ for $m = 1, \dots, \min(p_1, p_2)$, as well as their corresponding CVs

$\mathbb{Z}_1 = (\boldsymbol{z}_{1,1}, \boldsymbol{z}_{2,1}, \dots, \boldsymbol{z}_{\min(p_1,p_2),1})$ and $\mathbb{Z}_2 = (\boldsymbol{z}_{1,2}, \boldsymbol{z}_{2,2}, \dots, \boldsymbol{z}_{\min(p_1,p_2),2})$, where $\boldsymbol{z}_{m,1} = \mathbb{X}\widehat{\boldsymbol{w}}_{m,1}$ is the m^{th} CV for $\mathbb{X}_1$ and $\boldsymbol{z}_{m,2} = \mathbb{X}\widehat{\boldsymbol{w}}_{m,2}$ is the m^{th} CV for $\mathbb{X}_2$, for $m = 1, \dots, \min(p_1, p_2)$. Similar to PCA, with sample data, components can be calculated using the EVD or SVD of certain matrices involving sample covariance matrices.[24]

1.3.3. Adding regularization and sparsity

The objective of PCA is to find linear combinations that express *maximum variation* in one set of data, whereas the objective of CCA is to find linear combinations that express *maximum correlation* between two sets of data. Regardless, obtaining solutions to their objective functions require the calculation of an inverse of a covariance matrix. This becomes challenging, and often impossible, in the presence of multicollinearity or $n < p$ data scenarios;[21] both conditions are prevalent with omic data integration. A covariance matrix for such data becomes ill-conditioned under these circumstances and its inverse does not exist, meaning the computation of PCA and CCA becomes compromised.

These issues can be solved by *regularization* techniques. Regularization was developed in regression settings before being ported over to solve issues experienced by PCA, CCA, and other multivariate methods. When regressing an outcome $\boldsymbol{y}$ on a set of variables contained in design matrix $\mathbb{X}_d$, the objective function for ordinary least squares (OLS) regression can be written as:

$$\underset{\boldsymbol{\beta}}{\text{minimize}}\{\|\boldsymbol{y} - \mathbb{X}_d\boldsymbol{\beta}\|_2^2\},$$

where $\boldsymbol{\beta}$ is a vector of regression coefficients (parameters) to be estimated. Estimation involves taking the inverse of $\mathbb{X}_d'\mathbb{X}_d$ in the OLS estimating equations $\widehat{\boldsymbol{\beta}} = (\mathbb{X}_d'\mathbb{X}_d)^{-1}\mathbb{X}_d'\boldsymbol{y}$, which is not possible when $n < p$. In 1970, Hoerl and Kennard presented 'ridge regression'[27] which adds a *penalty function* to OLS objective function:

$$\underset{\boldsymbol{\beta}}{\text{minimize}}\{\|\boldsymbol{y} - \mathbb{X}_d\boldsymbol{\beta}\|_2^2\}$$
$$subject\ to\ \|\boldsymbol{\beta}\|_2^2 < t,$$

where t is a constant to be chosen by the user referred to as a *tuning parameter* and, paired with the *penalty function* $\|\boldsymbol{\beta}\|_2^2$ that it constrains, accomplishes two important things. First, it allows estimation even when $n < p$. Second, even though the ridge estimator is biased, it has potential to have greatly reduced variance, thereby achieving

a lower mean squared error (MSE) and becoming arguably superior to OLS.[21,27] A key mechanical aspect to ridge regression is that the tuning parameter $shrinks$ estimates since the sum of their squared values must be less than t.

In 1996, Robert Tibshirani presented least absolute shrinkage and selection operator (LASSO) regression[28], which has the following objective function:

$$\underset{\boldsymbol{\beta}}{\text{minimize}}\{\|\boldsymbol{y} - \mathbb{X}_d\boldsymbol{\beta}\|_2^2\}$$
$$subject\ to\ \|\boldsymbol{\beta}\|_1 < t,$$

where constraining the L_1-norm penalty function not only shrinks the parameter estimates but also, due to its geometry, shrinks some to exactly 0. By using the LASSO penalty, one can obtain regression solutions that have less non-zero parameter estimates than there are variables in $\mathbb{X}_d$, which greatly improves variable selection and model interpretation.[28] This started the term 'sparse' regression.

In 2005, Zou and Hastie presented 'elastic net' regression that included both ridge and LASSO penalties at the same time, allowing sparse estimation in $n < p$ scenarios.[29] A multitude of developments have emerged over the years[30] to improve penalty-based regression, including the grouped LASSO, where group structure between variables can be specified and incorporated to the selection process;[31] the fused LASSO, where parameter estimates corresponding to comparable variables are influenced towards similar values;[32] and the adaptive LASSO, where weights can be given on a per-variable basis to further control the estimate shrinking process.[33]

In 2006, Zou, Hastie, and Tibshirani published the first sparse PCA method.[34] The authors cast PCA as a regression problem and ported over the concepts from sparse regression. The same penalty functions could be applied to the loading vectors from PCA, resulting in the following generalized objective function for a sparse PCA method:

$$\underset{\boldsymbol{v}_1}{\text{maximize}}\{\boldsymbol{v}_1'\Sigma\boldsymbol{v}_1\}$$
$$subject\ to\ \|\boldsymbol{v}_1\|_2^2 = 1$$
$$subject\ to\ P(\boldsymbol{v}_1) < t,$$

where the penalty function P could take on one of a number of penalties, like the ridge or LASSO, and there could be more than one penalty function; Zou et al., 2006 used a naïve elastic net, for example.[34] Solving this objective function results in sparse loading vectors, meaning concise groups of variables are attributed to the variation explained by

PCs. This contrasts loading vectors estimated from non-sparse PCA, whereby all loading values are non-zero, leaving limited means to judge which variables are responsible for variation in the data.

More sparse PCA approaches have been published since the first.[35–43] For instance, Witten et al., 2009 published a sparse PCA method based on the penalized matrix decomposition (PMD) and LASSO or fused LASSO penalty functions[36,44], and Lee et al., 2010 published a sparse PCA method based on the non-iterative partial least squares algorithm with random effects penalty functions.[38]

In 2009, Waaijenborg et al., Parkhomenko et al., and Witten et al., published the first sparse CCA methods.[36,44–47] Similarly, they ported in penalty functions and were able to cast the following generalized objective function:

$$\underset{\boldsymbol{w}_{1,1},\boldsymbol{w}_{1,2}}{\text{maximize}}\left\{\frac{\boldsymbol{w}_{1,1}'\boldsymbol{\Sigma}_{12}\boldsymbol{w}_{1,2}}{\sqrt{\boldsymbol{w}_{1,1}'\boldsymbol{\Sigma}_{11}\boldsymbol{w}_{1,1}}\sqrt{\boldsymbol{w}_{1,2}'\boldsymbol{\Sigma}_{22}\boldsymbol{w}_{1,2}}}\right\}$$
$$subject\ to\ \left\|\boldsymbol{w}_{1,1}\right\|_2^2 = 1, \left\|\boldsymbol{w}_{1,2}\right\|_2^2 = 1,$$
$$subject\ to\ P_1(\boldsymbol{w}_{1,1}) < t_1, P_2(\boldsymbol{w}_{1,2}) < t_2$$

where the penalty functions P_1 and P_2 could be of different form if desired. Much like sparse PCA, a variety of sparse CCA methods have emerged.[48–57] For example, a few authors utilized the group LASSO to prioritize the grouping of genes during CCA estimation,[53,54,58] Wilms and Croux, 2015 solved an iterative prediction-based approach to estimating CCA components,[52] and Hao et al., 2017 developed a sparse CCA method for longitudinal data. [57]

1.3.4. Notes on solving objective functions, penalties, and tuning parameter selection

A key difficulty for any regularized or sparse method is solving the objective function. The above objective functions represent just the starting point for many of the final expressions and algorithms designed by authors in order for them to be solvable. The technical particulars are not the focus of this thesis but it is important to mention here that a number of algorithms tend to be iterative and computationally intensive.

The choice of penalty function can vary based on the objective of regularization and feasibility of implementation (i.e., depending on the way the method is formulated and the algorithm used to solve the corresponding objective function). Regardless of the penalty function chosen, tuning parameters greatly govern their influence on the results.[46]

There are several strategies to select tuning parameters. Typically, the sparse method is performed many times, each time using a different set of tuning parameters from a suitable range, and results across runs are compared based on some criteria defined by the user. Additionally, a cross-validation procedure whereby the data is split into training and testing sets is popular to help avoid overfitting, a modeling issue which is increasingly problematic as the number of variables increases (especially so for multivariate methods). For example, several authors have chosen tuning parameters that maximize the average (across cross-validation folds) test-sample canonical correlation, or that minimize the average (across cross-validation folds) difference between training-sample and test-sample canonical correlations (emphasizing the model reliability).[36,47,52,59]

In this thesis, I use several sparse methods and tuning parameter selection approaches. In each case, I define and reference which method and selection approach I am using.

1.4. Objective of this thesis and its organization

The objective of this thesis is to apply, test, and expand our knowledge of the sparse class of multivariate statistical methods. Three contributions in the form of manuscripts are included in Chapters 2, 3, and 4, respectively. Full context for each project precedes the manuscripts included in Chapters 2, 3, and 4, as well as in the introductory sections of these manuscripts.

Chapter 2 presents an in-depth analysis with real data from a toxicogenomics study. By applying sparse PCA and sparse regression, groups of genes are estimated to be jointly associated with drug toxicity, outlining for the toxicogenomics community a new a new way to observe relationships within their data. Chapter 3 presents an extensive simulation study that compares the performance of several popular sparse CCA methods for extracting accurate sets of cross-correlated features between high-dimensional data. By finding which methods work best in a vast range of conditions, and by providing a simulation design to do it, new knowledge of how these methods perform is generated. Chapter 4 presents an investigation of the non-parametric bootstrap approach to constructing confidence intervals and probability estimations for which variables are truly cross-correlated. Through simulation experiments, the performance of the strategy under $n < p$ conditions is demonstrated, showing its ability to improve our means for inference when using the otherwise exploratory sparse CCA.

Discussion and conclusions regarding these contributions are embedded within the final sections of their corresponding manuscripts. In Chapter 5 of this thesis, overall

key findings are highlighted, common strengths and limitations of the work is summarized, and future research ideas valuable given the current state of sparse multivariate methodology and needs in health research are discussed.

Chapter 2

2. APPLICATION OF SPARSE PCA TO TOXICOGENOMIC DATA

Context

My first project is in the form of a published, peer-reviewed manuscript. The work presented in this manuscript is an application of sparse PCA to real toxicogenomic data. The work was initially motivated by my experiences with two 'analysis challenge' conferences.

In the first year of my Ph.D. studies, I participated in the 18th Genetic Analysis Workshop (GAW18) hosted in Portland, Oregon, United States on October 13-17, 2012. Having just finished my M.Sc. Statistics degree, for which I completed a simulation study comparing sparse PCA methods,[60] I found a reasonable way to apply sparse PCA to the type-2 diabetes genomic data supplied by the conference. After attending and presenting my approach at the conference, I submitted it for publication.[61] This work, along with my collaboration with the data mining group during and after the conference,[62] created the backbone for me to hone my approach for future analyses with large omic data.

Leveraging my experiences, I participated in the 2013 International Conference on the Critical Assessment of Massive Data Analysis (CAMDA2013), held in Berlin, Germany on July 19-20, 2013. It was structured similarly to GAW18 but this time attendees could analyze one of three challenge data bases. One challenge data base was from a large toxicogenomic project.

The field of toxicogenomics has emerged to handle the merger between conventional drug toxicity assessment studies and large genomic, transcriptomic, and proteomic data. Such omic data have been incorporated to better predict toxic drugs at an early stage of the drug-development process, making the process more efficient and the final products safer for human consumption.[63,64] In some instances, data from large initiatives like the Japanese Toxicogenomics Project (TGP),[65,66] DrugMatrix,[67] PredTox,[68,69] and eTox[70,71] have been made available to spark research and development. The CAMDA2013 organizers linked attendees to processed gene expression data from the TGP data base.

After some background reading of toxicogenomics literature, I found there was a great need for data integration, multivariate analyses, and variable selection.[72] With guidance from my supervisor, I designed and presented at the conference an analysis pipeline involving sparse PCA for extracting groups of genes associated with drug

toxicity. My initial analysis was well received and, afterwards, we pursued peer-reviewed publication. The resulting manuscript is presented in the following section and constitutes the first contribution to my thesis.

We secured peer-reviewed publication in the *Systems Biomedicine* journal. Below is the full citation and acknowledgement for our manuscript. Then, starting on the next page, I include our manuscript. Please note that mathematical notation in this manuscript deviates slightly from the notation in the rest of the thesis. This is because it was published before finalizing the notation for the thesis. That said, notation is fully explained within the manuscript.

Readers who have read Chapter 1 of this thesis might elect to skip the section called "Sparse PCA and regression methodology" within the manuscript. Although a difference in notation, the differences are intuitive and the rest of the manuscript is understandable without reading that section.

CITATION:

- Bonner, A., & Beyene, J. (2014). Detecting networks of genes associated with human drug induced liver injury (DILI) concern using sparse principal components. *Systems Biomedicine*, *2*(1), e29413. http://dx.doi.org/10.4161/sysb.29413

ACKNOWLEDGEMENT:

-

MANUSCRIPT BEGINS ON THE NEXT PAGE…

Detecting networks of genes associated with human drug induced liver injury (DILI) concern using sparse principal components.

Ashley J. Bonner[1], Joseph Beyene[12§]

[1] Department of Clinical Epidemiology and Biostatistics, McMaster University, 1280 Main St. West, Hamilton, ON, L8S 4L8, Canada
[2] Department of Mathematics and Statistics, McMaster University, 1280 Main Street West, Hamilton, ON, L8S 4L8, Canada.

§Corresponding author, Dr. Joseph Beyene (beyene@mcmaster.ca)

Conflict-of-Interest and Financial Disclosure Statements:
None.

Keywords:
statistics, toxicogenomics, drug toxicity, drug-induced liver injury, sparse principal component analysis, sparse principal components, loadings, groups of genes, differentially expressed.

Abbreviations:

CAMDA:	Critical Assessment of Massive Data Analysis
DEG:	differentially expressed gene
DEPC:	differentially expressed principal component
DILI:	drug-induced liver injury
FARMS:	Factor Analysis for Robust Microarray Summarization
PC:	principal component
PCA:	principal component analysis
TGP:	Japanese Toxicogenomics Project

Abstract

Background: The 12th Annual International Conference on the Critical Assessment of Massive Data Analysis (CAMDA) used data from together the massive Japanese Toxicogenomics Project (TGP) to predict drug-induced liver injury (DILI) concern provided by the U.S. Food and Drug Administration (FDA. The challenge was to predict DILI concern by means of gene expression data. Analysis of this high-dimensional toxicogenomic data requires statistical methodologies that can detect the transcriptomic associations with toxicity. **Methods:** We propose an analysis technique that involves sparse principal component analysis to efficiently reduce the dimension of the analysis problem. Sparse principal component variables are composed of groups of expressed genes. Associations between DILI concern and sparse principal component variables were tested and further scrutinized with sparse regression methodology to identify concise transcriptomic structures potentially responsible for and predictive of drug toxicity. **Results:** Working with a subset of the TGP data with FDA DILI concern classification, we identified 5 transcriptomic structures (sparse principal component variables) statistically associated with DILI concern. The most statistically significant structure consists of the genes *ZBTB16, FLVCR2*, *TNS3*, and *ASB13*. **Conclusion:** Sparse statistical methods offer a new way to handle analysis issues with massive omic data. Sparse PCA can efficiently extract groups of transcriptomic markers that may indicate drug toxicity.

2.1. Introduction

The current estimated cost in US dollars to develop a drug is $1.8 billion, but most drugs never make it to market, largely due to toxicity levels deemed unsafe for the human liver.[1] As it is unethical to test the effects of compounds in humans, toxicity has classically been tested in animal-based experiments. Hence, conclusions may not be generalizable to the human population, especially when the effects of toxicity are revealed after prolonged exposure. The testing inaccuracy could result in passing unsafe or discarding useful drugs, which hinder public health efforts and squander resources. In an attempt to improve testing accuracy, the field of toxicology has embraced animal and now human 'omic' data (genomics, transcriptomics, proteomics, metabolomics) to help identify biological material that predict the toxicity of compounds at an early stage. Therefore, resources are saved from developing drugs that will ultimately fail in the public domain. The use of omic data to inform toxicology is commonly referred to as *toxicogenomics* and is now the focus of several research initiatives. One such initiative is the Japanese Toxicogenomics Project (TGP).[2] With human *in vitro,* rat *in vitro*, and rat *in vivo* experimental models, 131 compounds were applied to liver samples and microarray gene expression profiles were obtained using Affymetrix GeneChip® technology.

To facilitate prediction of toxicity in new drugs, the Food and Drug Administration (FDA) developed a classification system of human drug-induced liver injury (DILI) concern (most, less, and no DILI concern) for drugs currently on the market.[3] The drugs they classified had been on market for a minimum of 10 years, allowing sufficient public interaction to obtain updated and realistic DILI concern information, not otherwise attainable due to ethical reasons. This new classification of human-based drug toxicity can be linked to toxicogenomic databases, facilitating the search for omic markers that predict toxicity based on this indicator. The 12th Annual International Conference on Critical Assessment of Massive Data Analysis (CAMDA 2013) linked FDA classification labeling to the TGP data and proposed analysis challenges involving prediction of drug toxicity. Discovering novel biomarkers associated with DILI concern within the TGP data may aid in our understanding of mechanisms of toxicity and enhance our ability to assess the toxicity of new compounds. However, the breadth and complexity of omic data makes analysis challenging and to extract key information it is important to enrich statistical methodology with biological context.[4]

Commonly, simple statistical methods are used to test associations between drug toxicity and each marker, one at a time, developing a list of top candidate genes. Although this approach is easy to implement, it involves conducting thousands of statistical tests and is prone to spurious associations (i.e. the 'multiple testing problem').

Perhaps more concerning is that by testing markers independent of one-another, this analysis does not acknowledge the fact that genes may act in concert to influence a phenotype. Complex networks of omic material will likely be the underlying indicator of drug toxicity. Identifying and characterizing these indicators requires more sophisticated statistical methods. We believe that recent advancement in a new class of so-called *sparse* statistical methods validates the use of principal component analysis (PCA) as a primary analysis tool to detect these complex networks.

PCA is a commonly used multivariate method for both dimension reduction and data visualization. It assembles a new set of variables, called principal components (PCs), from linear combinations of original variables. These PCs are uncorrelated and ordered by maximal variance, giving the analyst an easier dataset to work with if they can interpret what the PCs mean. Unfortunately, this is a major challenge with high-dimensional data, as found in toxicogenomics, since the PCs are formed by linear combinations of *all* original variables (a weighted sum of 1000 genes, for example) and are uninterpretable. To overcome this limitation, *sparse* principal component analysis methods [5,6,7] have been developed by combining PCA with *sparse* regression methodology.[8,9] Sparse PCA restricts principal components to be formed by *interpretable* linear combinations of smaller (sparse) subsets of the original variables (a weighted sum of 10 genes, for example). Tuning parameters control the level of sparseness induced, making sparse PCA procedures very flexible. *Sparse* principal components reflect concise and interpretable groups of original variables that remain in the linear combinations.

In this paper, we present an analysis strategy built around using sparse PCA to extract groups of related genes from a toxicogenomic database, testing their associations with drug toxicity. We apply this analysis strategy to subsets of the TGP database with the FDA's classification of human DILI concern as the measure of drug toxicity. We hope to convey that this new class of *sparse* statistical methods may prove beneficial to toxicogenomic analyses, as they specialize at separating the 'signal' from 'noise' amidst high-dimensional data.

The rest of our paper is organized as follows. In the Materials and Methods section, we describe the data we used, the essentials of sparse methodology, and our analysis strategy. We then present our Results and provide a Discussion on the potential merits and limitations of sparse methodology in toxicogenomic analyses.

2.2. Materials and Methods

Description of Data:

Data source: Completed in 2007, the TGP database was the result of a 5-year collaborative effort between government and private companies to obtain gene expression data using Affymetrix GeneChip® technology to characterize the response to various drugs across a variety of experimental units.[2] The completed TGP database provided by CAMDA 2013 contains thousands of results from experiments, involving 131 drugs applied at 4 dose levels (control, low, middle, and high) to human and rat *in vitro* hepatocytes, and administered to rat *in vivo* with either a single dose or repeated doses. Transcriptomic material was extracted from experimental units at several time points after the drugs had been given. Full details regarding the study design and protocol are given by Uehara et al.[2] and complete data, along with the portion we obtained including FDA human DILI concern labels, are available through the CAMDA 2013 conference website (http://dokuwiki.bioinf.jku.at/doku.php).

Samples used: We considered only those drugs with FDA human DILI concern classification; 93 of the 119 drugs tested on human samples, and 101 of the 131 drugs tested on rat samples. We primarily focused on the 93 samples from human in vitro experiments that specifically received high dose levels and had gene expression measured at 8 hours; a single subset of this rich database. However, to compare results across different experimental conditions, we also repeated our analysis on another 15 subsets of the database. Anticipating that higher doses result in more robust gene expression measurements [10] and due to many drugs not being administered at low doses in the human in vitro samples, we considered only samples that received middle or high dose levels. Likewise, since measurements of gene expression in the human samples were not taken at 2 hours for many drugs, we considered only gene expression values measured at later time points.

Transcriptomic markers used: Human gene expression data were obtained with the Affymetrix GeneChip® Human Genome U133 Plus 2.0 Array and rat gene expression data were obtained with Affymetrix GeneChip® RAE 230A 2.0 Array.[2] Although raw gene expression data was available, we chose to use preprocessed data that had undergone batch-effect correction and the Factor Analysis for Robust Microarray Summarization (FARMS) [11], that was also provided. A total of 18988 probesets were available for replicate-collapsed human samples, and 12088 probesets were available for replicate-collapsed rat samples. Almost all probesets had gene names provided and all gene names were unique; as such we refer to probesets and genes interchangeably unless the gene name was not provided. We used inter-quartile range to filter genes that did not vary much and kept the top 1000 genes to simplify analysis.

Classification variable for human DILI concern: Human DILI concern ('most', 'less', and 'no' DILI concern in humans) was provided to detect if gene expression measurements differed across DILI classification. However, since only 8 drugs are classified as 'no DILI concern', we reclassified the human DILI concern variable to be binary ('most' vs. 'less or no' DILI concern) to obtain a more balanced and simple categorical variable. Of the 93 drugs tested in human tissue, 40 are labeled as 'most DILI concern' and 53 are 'less or no DII concern' and of the 101 drugs tested in rat specimens, 41 are 'most DILI concern and 60 are 'less or no DILI concern'.

Sparse PCA and Sparse Regression Methodology:

Here we give an intuitive description of sparse PCA and sparse regression and refer the reader to refs. [5,6,7,8,9] for mathematical details.

Given a data matrix $\boldsymbol{X} = (X_1, X_2, \dots, X_p)$, where p variables (e.g., probsets or other omic markers) are measured for n observations (samples; drugs), classical PCA constructs linear combinations (weighted sums) from the variables in the form

$$Z_i = v_{i,1}X_1 + v_{i,2}X_2 + v_{i,3}X_3 + \cdots + v_{i,p-1}X_{p-1} + v_{i,p}X_p, \quad i = 1,2, \dots, \min(n,p),$$

where Z_i is the i^{th} principal component (PC) variable and the $v_{i,j}$'s are the loadings (coefficients; weights) for the j^{th} variable in the i^{th} PC. By utilizing the correlation structure among X's, PCA constructs PCs in such a way that they are uncorrelated, ordered by maximal variance, and the total amount of information (variance) in the new data matrix $\boldsymbol{Z} = (Z_1, Z_2, \dots, Z_{\min(n,p)})$ is equal to the total amount of information in the original data matrix $\boldsymbol{X}$. We may now choose to analyze the Z's instead of X's, as these defining properties potentially simplify analysis. For example, analyzing uncorrelated Z's allows for easier regression analysis as there is no issues with colinearity. Even more appealing, if $n < p$ then the number of Z's to analyze will be less than the number of X's. Likewise, since Z's are ordered by variance, the first few PCs will typically hold a large portion of the information and we can discard the trailing ones without losing much. For these last two reasons, PCA is an excellent so-called *dimension-reduction* tool, able to compress large amounts of data into a few important components. In some cases, PCA can be an excellent *exploratory* tool, since groups of highly correlated X's will bear substantial weight in the same PC and we can detect groups of variables through the magnitude of their loadings ($v_{i,j}$'s). However, since PCs are linear combinations of *all* X's, this makes them nearly impossible to interpret when dealing with thousands of correlated X's, as is common with high-dimensional toxicogenomics data.

S*parse* PCA is an extension to classical PCA that forces the loadings for some X's to be exactly 0, creating *sparse* linear combinations in the form (for example)

$$Z_i = v_{i,1}X_1 + v_{i,2}X_2 + 0X_3 + \cdots + 0X_{p-1} + v_{i,p}X_p, \quad i = 1,2,\ldots,\min(n,p),$$

where now the $v_{i,j}$'s that do not have much relation to a PC are removed. This results in a more interpretable set of PCs, having small groups of X's (genes, for example) creating new variables to work with. This is objectively achieved through integrating constrained (*sparse*) regression methodology to the derivation of PCA. For example, finding classical PCA solutions through the singular value decomposition (SVD) $\boldsymbol{X} = \boldsymbol{UDV}'$, where $\boldsymbol{V}$ holds all the loadings and $\boldsymbol{Z} = \boldsymbol{UD}$ are the PCs, Witten et al. [6] applied constraints to elements of $\boldsymbol{V}$, shrinking some of them to 0. A so-called *tuning parameter* (λ) controls the number of loadings that are forced to 0, providing a flexible framework from which to obtain *sparse* PCs. As a trade-off for acquiring interpretable PCs by introducing sparseness, a proportion of the total information (variance) is lost.

Finally, sparse regression works in a similar way, in that the coefficients estimates ($\hat{\beta}$'s) of a regression model of the form

$$Y = \beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_3 + \cdots + \beta_{p-1}X_{p-1} + \beta_pX_p + \varepsilon,$$

are constrained, shrinking a number of them, dictated by tuning parameters, directly to 0. It has been integrated to many extensions of the linear model, such as generalized linear models, including binary logistic regression.

Analysis Strategy:

The novel component of our analysis strategy is the use of sparse PCA to automatically select groups of variables, however we suggest a more broad analysis pipeline as visualized in Figure 1.

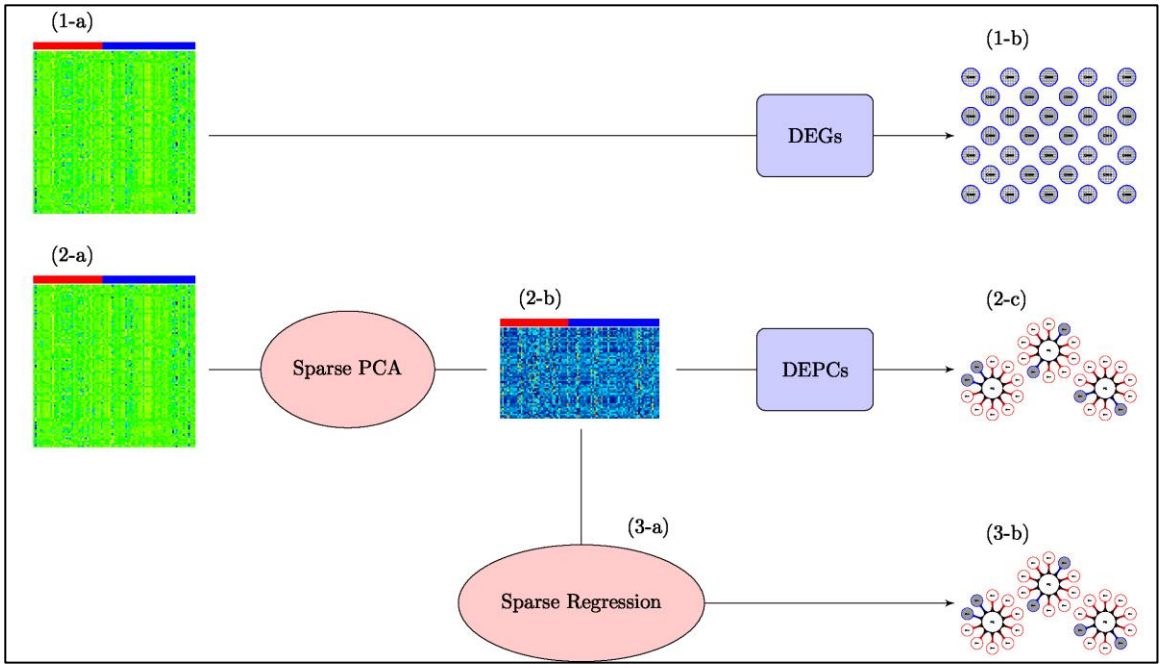


Figure 1: A visual of our analysis strategy applied to human in vitro, high dose, 8 hour sampling time data. We begin in the top row (1-a), by conducting a Differentially Expressed Genes (DEGs) analysis on the gene expression matrix; columns represent the 93 samples (40 'most' and 53 'less or no' DILI concern), rows represent 1000 expressed genes. This returns a list of top DEGs (1-b); the genes that are most significantly associated with DILI concern. We then move to the middle row (2-a), using sparse PCA on the same gene expression matrix to obtain new sparse principal component (PC) variables (2-b) to work with; columns for this new data matrix again represent the 93 samples, but rows represent the 93 new sparse PC variables (we have reduced the dimension from 1000 to 93). Then, we conduct a Differentially Expressed PCs (DEPCs) analysis on the PC expression matrix to obtain a list of top DEPCs (2-c); the sparse PCs that are most significantly associated with DILI concern. At this point, we examine the genes that contribute to these DEPCs to makes sense of what the structures mean and make note of those genes in these structures that were also identified as differentially expressed in the DEGs analysis. As a final validation step (3-a), we apply sparse regression to the same 93 sparse PC variables to identify a concise list of sparse PCs that are potentially related to DILI concern (3-b).

We implemented this analysis for each of the 16 subsets of the TGP data we considered but refer only to the human in vitro, high dose level, 8 hour gene expression sampling

time subset while explaining the steps in detail. All analysis was conducted with R statistical software (version 3.0.1).[12]

Step 1: Identify genes independently associated with DILI concern with a Differentially Expressed Gene (DEG) analysis: This step can be visualized with 1-a and 1-b in Figure 1. Starting with the 1000 most variable genes, we first investigated if any genes are independently associated with DILI concern. Instead of using the two independent samples t-test for each gene, we used the moderated t-test [13] from the R package 'limma' [14] which yields more conservative p-values; without this correction, when conducting many tests, suspiciously small standard errors can inflate test statistics. We obtained 'moderated t' test statistics with associated p-values and claimed that genes with p-value < 0.05 were significantly associated with DILI concern. Additionally, using 'limma', we obtained q-values (p-values adjusted to account for false-discovery rates) [15,16] to eliminate genes likely to be claimed falsely significant due to conducting multiple tests. With this list of differentially expressed genes likely to be independently associated with DILI concern, we moved to search for more complex relationships between the gene expression data and DILI concern.

Step 2: Identify groups of genes jointly associated with DILI concern using sparse PCA and a Differentially Expressed Principal Component (DEPC) analysis: This step can be visualized with 2-a, 2-b, and 2-c in Figure 1. Starting with the 1000 most variable genes, we used the sparse PCA method by Witten et al. [6] executed with the R package 'PMA' [17], to obtain sparse principal component variables. We chose this sparse PCA method over certain other formulations [5,7] as a result from our previous work [18] that found it to perform better in computer simulations with high-dimensional data. For each of a range of tuning parameters (λ = 2, 3, 5, 7, 10, 15), we ran sparse PCA and examined the trade-off between adjusted percentage explained variance [18] and sparseness among PCs with the goal to select a tuning parameter that delivered principal components with extremely sparse loading vectors while keeping a large proportion of variance. We chose tuning parameter λ = 3 because while compared to λ = 5, 7, 10, and 15 it resulted in loading vectors that were drastically more sparse at a small cost of information loss and loading vectors were almost as sparse as those from λ = 2 while comparatively keeping a substantial amount of information. Additionally, and important from a practical view, this delivered small and interpretable linear combinations so we could look at the principal components in clear detail. We then statistically tested associations between each sparse PC variable and DILI concern using a permutation approach claiming sparse PCs with p-values < 0.05 to be significantly associated with DILI concern. With this list of 'differentially expressed' sparse PCs (DEPCs) likely to be independently associated with

DILI concern, we then recorded and examined the genes that made up the PCs along with their respective loadings in the PC linear combination.

Step 3: Validate DEPCs with Sparse Regression: This step can be visualized with 3-a and 3-b in Figure 1. We simultaneously entered the sparse principal component variables into a binary logistic regression model framework with DILI concern as the outcome, where the coefficient estimates were obtained with sparse regression methodology; penalized maximum likelihood with a least absolute shrinkage and selection operator (LASSO) penalty [8] with tuning parameter selected via a cross validation procedure that maintains good prediction from the model, executed with the R package 'glmnet'.[19] From this sparse regression model, we obtained the list of sparse PC variables that remained, expecting overlap with the list of sparse PCs identified to be significantly related to DILI concern in Step 2, then recorded and examined the genes and loadings in these PCs. Sparse PCs found both in Step 2 and Step 3 were deemed to be most likely to be associated with DILI concern.

2.3. Results

We now present the results from our analysis strategy, reporting details on the human in vitro, high dose, 8 hour gene expression sampling time subset but also including summaries of findings for the other 15 subsets.

Of the 1000 genes, 54 had significantly different expression between 'most' and 'less or no' DILI concern samples, according to the moderated t tests with p-value < 0.05. Table 1 displays their gene names, effect size, and associated p-values.

*Table 1: Differentially expressed genes (DEGs; independently associated with DILI concern) from our analysis of the human in vitro, high dose, 8 hour gene expression sampling time subset. *Difference in Means is 'most' – 'less or no' DILI concern; a positive value indicates a larger gene expression value for the most DILI concern group. **p indicates the p-value obtained from the moderated t test.*

Rank	Gene Name	Diff. in Means*	p**	Rank	Gene Name	Diff in means*	p**
1	SNAPC1	0.228	0.0001	28	TNFRSF1B	-0.096	0.0263
2	TSLP	0.221	0.0005	29	BBS12	0.109	0.0268
3	ANKRD1	0.277	0.0027	30	FKBP5	-0.155	0.0271
4	FHL2	0.191	0.0050	31	MIR22HG	-0.189	0.0273
5	CTH	0.126	0.0055	32	FSTL1	-0.156	0.0280
6	HEXIM1	-0.178	0.0057	33	POR	-0.088	0.0282
7	NFIL3	0.195	0.0076	34	TUFT1	0.153	0.0289

8	ETS1	0.117	0.0078	35	LPIN2	-0.098	0.0304
9	MIR3682	-0.130	0.0090	36	RBMXL1	-0.092	0.0306
10	TUBE1	0.157	0.0098	37	CEBPD	-0.124	0.0330
11	BLNK	0.147	0.0117	38	EXT1	0.106	0.0331
12	CIDEC	-0.095	0.0132	39	SLC7A5	0.122	0.0336
13	HAS3	-0.092	0.0134	40	LMCD1	0.187	0.0337
14	ANGPTL4	-0.401	0.0163	41	ERBB3	-0.107	0.0352
15	ELL2	-0.093	0.0173	42	PDLIM5	0.066	0.0352
16	SRSF6	-0.104	0.0179	43	PDK4	-0.138	0.0352
17	INHBE	0.197	0.0190	44	MT1F	0.072	0.0361
18	TXNIP	-0.137	0.0202	45	GEM	0.206	0.0378
19	F3	0.240	0.0205	46	RBMX	-0.111	0.0380
20	FOXA1	-0.134	0.0209	47	ZBTB43	0.127	0.0381
21	MTHFD2	0.173	0.0218	48	HHEX	-0.139	0.0385
22	SLC25A20	-0.125	0.0220	49	TMEM158	0.074	0.0439
23	HSD17B2	-0.136	0.0240	50	ASNS	0.110	0.0446
24	FOXQ1	-0.134	0.0242	51	ATP8B1	0.095	0.0458
25	DUSP6	-0.125	0.0244	52	TNFAIP3	0.147	0.0464
26	CHAC1	0.125	0.0258	53	C11orf96	0.124	0.0474
27	SERTAD2	0.141	0.0261	54	RTP3	0.106	0.0496

None of these 54 genes remained statistically significant when using the FDR-adjusted p-value (q-value) with cut-off $q < 0.10$. Table 2 displays counts of the number of DEGs found in each of the 16 subsets we analyzed.

*Table 2: Counts of DEGs across all 16 subsets analysed. The total number of DEGs, along with the number of those that are up-regulated ('most DILI concern' has larger gene expression than 'less or no DILI concern') in brackets and those that are down-regulated in square brackets. * 's.Dose' means rats received only a single dose of drug. ** 'r.Dose' means rats received repeated doses over time.*

Data Source	Dose Level	Gene Expression Sampling Time	# genes p < 0.05 (mod. t) Total (up-reg) [down-reg]	# genes q < 0.10 (FDR) Total
Human In Vitro	High	8 hour	54 (29) [25]	0
Human In Vitro	High	24 hour	26 (12) [14]	0

Human In Vitro	Middle	8 hour	74 (32) [42]	0
Human In Vitro	Middle	24 hour	2 (2) [0]	0
Rat In Vitro	High	8 hour	29 (8) [21]	0
Rat In Vitro	High	24 hour	153 (1) [152]	17
Rat In Vitro	Middle	8 hour	42 (39) [3]	0
Rat In Vitro	Middle	24 hour	34 (20) [14]	0
Rat In Vivo (s.Dose*)	High	9 hour	53 (36) [17]	0
Rat In Vivo (s.Dose)	High	24 hour	82 (49) [33]	0
Rat In Vivo (s.Dose)	Middle	9 hour	228 (193) [35]	202
Rat In Vivo (s.Dose)	Middle	24 hour	140 (133) [7]	5
Rat In Vivo (r.Dose**)	High	15 day	50 (27) [23]	0
Rat In Vivo (r.Dose)	High	29 day	108 (21) [87]	1
Rat In Vivo (r.Dose)	Middle	15 day	53 (31) [22]	0
Rat In Vivo (r.Dose)	Middle	29 day	103 (69) [34]	0

The rat in vivo samples receiving single dose at middle dose levels with gene expression measured at 9 hours had a notably large amount of DEGs compared to the rest; 228 with 202 remaining statistically significant after the more strict FDR-adjustment.

Applying sparse PCA with tuning parameter λ=3 to the gene expression data, we obtained the 93 ($\min(n = 93, p = 1000)$) sparse principal components. The minimum, median, and maximum number of genes that contributed to a sparse principal component with non-zero loadings was 11, 18, and 33, respectively. Compared to classical PCA, which would return principal components built from all 1000 genes, this is a substantial amount of sparseness and, therefore, interpretability gained. As a sacrifice for simplifying the data in this way, we lost 29.97% of total variance (information) contained in the original 1000 genes. Considering that sparse PCA simplified our focus to a median of just 18 (1.8% of the 1000) genes per component, therefore giving us groups of genes to explore, and reduced our analysis burden from 1000 gene tests to 93 principal component tests, we consider this a massive gain for a comparably small loss. Similar data reduction statistics were observed for the remaining 15 subsets.

Of the 93 sparse principal component variables, 5 had significantly different cumulative expression values between ‘most’ and ‘less or no’ DILI concern samples, according to the permutation test with $p < 0.05$. Table 3 provides a summary of these

DEPCs, including p-values, a list of contributing genes, and how many of the contributing genes were found in the DEG analysis.

Table 3: Differentially expressed PCs (DEPCs; associated with DILI concern) from our analysis of the human in vitro, high dose, 8 hour gene expression sampling time subset. Also provided are counts of probesets that contribute to respective PCs, along with the number that were individually associated with DILI concern (DEGs) and gene names. p-values were obtained from permutation tests.

PC	p-value	# of contributing probesets (# that are DEGs)	Contributing genes in order of absolute loading value
PC49	0.0074	12 (1)	ZBTB16, FLVCR2, TNS3, ASB13, MLPH, CUX2, FKBP5, SHROOM3, PANK1, TGFBR2, MYLK, ORC6
PC15	0.0213	16 (0)	PCK1, ID1, RRAGC, HSPA1B, SOWAHC, NPC1, PLEKHB2, RNF19B, STX3, CCNE2, KANSL1, PIM3, FNIP2, GNA13, OTUD1, IRS2
PC13	0.0310	13 (8)	DDIT4, MTHFD2, FGF21, DDIT3, INHBE, TRIB3, TUBE1, HSPA13, CHAC1, SLC7A5, TSLP, ASNS, CTH
PC7	0.0382	15 (2)	NUAK2, F3, C8orf4, ENC1, EDN1, CCL2, LMCD1, PLK2, G0S2, KRT7, C1orf63, TRIM6, FILIP1L, THBS1, FASTKD3
PC72	0.0389	18 (3)	SLC40A1, SNAI2, KRCC1, IRS2, LPIN2, CYP1A1, CBLB, BCL6, HMGB2, CEBPD, ERBB3, HECA, RB1CC1, GNA13, EFNA1, TBC1D8, C6orf203

Figure 2 provides a visual representation of the top 3 of these PCs, showing the different types of groups of genes that are associated with DILI concern, in terms of size and composition.

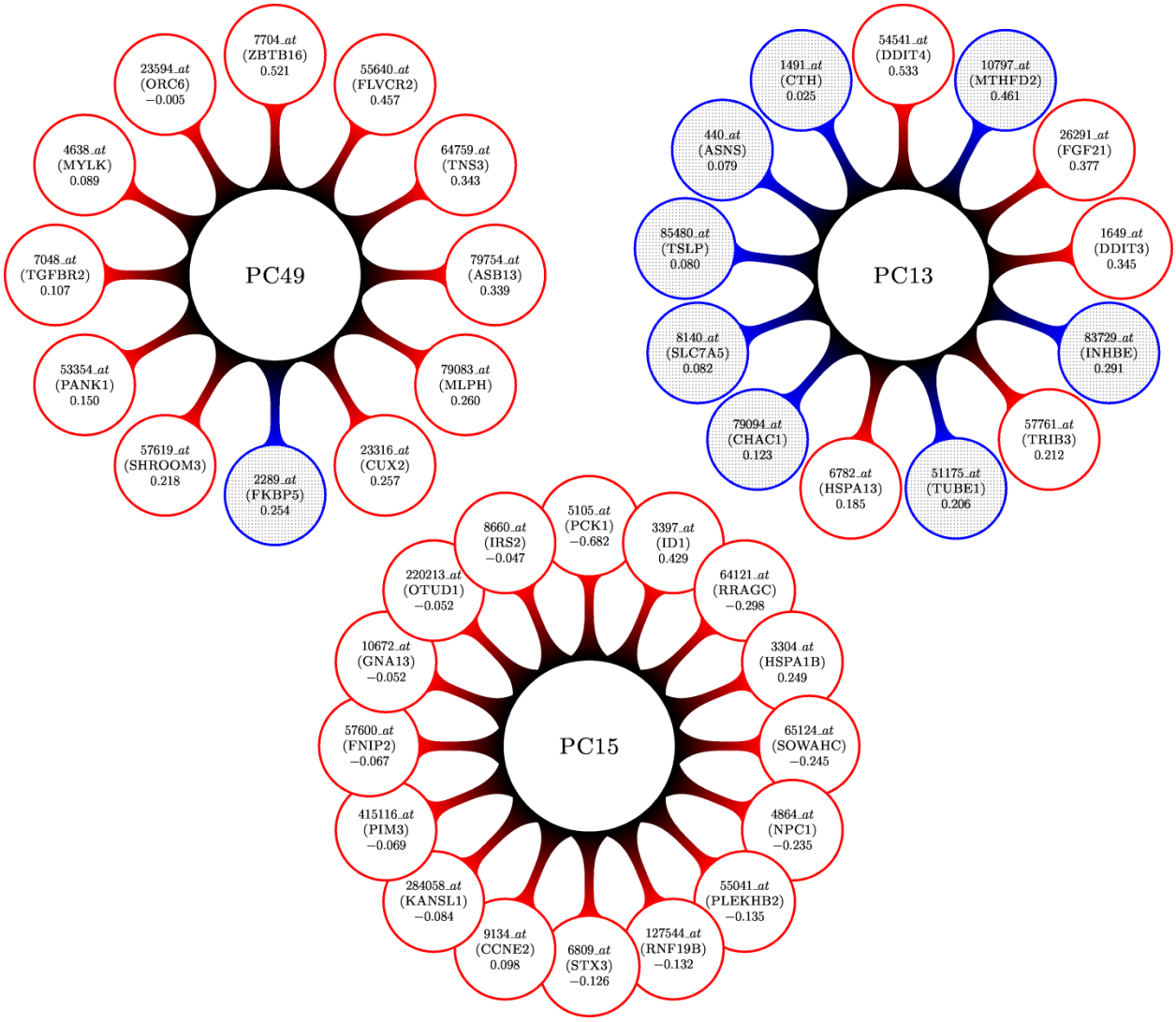


Figure 2: A visual display for the top 3 differentially expressed (most associated with DILI concern) sparse principal components (DEPCs) from our analysis of the human in vitro, high dose, 8 hour gene expression sampling time subset. Larger central circles represent the principal components. Attached to each are the genes that form the linear combinations; probesets (gene names) and loading values are inside the outer circles. Shaded circles represent genes that were found to be independently associated with DILI concern (DEGs), whereas non-shaded circles contain genes that were not. PC15 might bring forth a network of transcriptomic material that is associated with DILI concern, not otherwise being found with more simple statistical tests. PC13 shows us that some marginally associated genes behave similarly.

PC49 consists predominantly of probesets (genes), 7704_at (*ZBTB16*), 55640_at (*FLVCR2*), 64759_at (*TNS3*), and 79754_at (*ASB13*), with lesser contribution from eight more correlated genes, meaning these may have a joint relationship with drug toxicity. One of these eight, 2289_at (*FKBP5*), was identified as significantly associated with DILI concern in the DEG analysis. PC15 is a slightly larger network comprised entirely of genes which were not identified in the DEG analysis, meaning these genes might only be identified when accumulated into a composite variable like this sparse principal component. PC15 is driven by probesets (genes) 5105_at (*PCK1*), 3397_at (*ID1*), and 64121_at (*RRAGC*). In contrast, 8 out of 13 genes influencing PC13 had independent associations with DILI concern. Its top 4 weighted probesets (genes), 54541_at (*DDIT4*), 10797_at (*MTHFD2*), 26291_at (*FGF21*), and 1649_at (*DDIT3*), had only one DEG, meaning these genes are likely related to drug toxicity although they did not independently surface. Table 4 reports the number of sparse PCs that were significantly related to DILI concern across all 16 subsets of data we analyzed.

*Table 4: Counts of DEPCs across all 16 subsets analysed. The total number of DEPCs, along with the number of those which are up-regulated ('most DILI concern' has larger PC cumulative expression values than 'Less or No DILI concern') in brackets and those which are down-regulated in square brackets. * 's.Dose' means rats received only a single dose of drug. ** 'r.Dose' means rats received repeated doses over time.*

Data Source	Dose Level	Gene Expression Sampling Time	# PCs p < 0.05 (perm.) Total (up-reg) [down-reg]
Human In Vitro	High	8 hour	5 (3) [2]
Human In Vitro	High	24 hour	1 (1) [0]
Human In Vitro	Middle	8 hour	12 (8) [4]
Human In Vitro	Middle	24 hour	0 (0) [0]
Rat In Vitro	High	8 hour	2 (2) [0]
Rat In Vitro	High	24 hour	18 (8) [10]
Rat In Vitro	Middle	8 hour	3 (1) [2]
Rat In Vitro	Middle	24 hour	2 (1) [1]
Rat In Vivo (s.Dose*)	High	9 hour	3 (1) [2]
Rat In Vivo (s.Dose)	High	24 hour	5 (3) [2]
Rat In Vivo (s.Dose)	Middle	9 hour	15 (5) [10]
Rat In Vivo (s.Dose)	Middle	24 hour	19 (12) [7]
Rat In Vivo (r.Dose**)	High	15 day	6 (4) [2]
Rat In Vivo (r.Dose)	High	29 day	12 (6) [6]
Rat In Vivo (r.Dose)	Middle	15 day	6 (3) [3]

Rat In Vivo (r.Dose)	Middle	29 day	12 (5) [7]

Lastly, sparse regression was used to narrow down a more concise list of DEPCs. Entering the 93 sparse PCs, the model selected PC49 and PC15. These are the top 2 sparse PCs we identified to be most significantly associated with human DILI concern in the DEPCs analysis, bringing some statistical validation to these findings.

2.4. Discussion

In this paper, we identified sparse PCs that are differentially expressed between groups of 'most' and 'less or no' DILI concern, providing small networks of genes for further study in relation to drug toxicity. Apart from being able to investigate groups of variables, there are several components that make sparse PCA attractive. Since the number of principal components returned is $\min(n, p)$ (i.e., minimum of n samples and p variables), it will always reduce the number of variables to analyze when applied to high-dimensional ($n < \mathrm{p}$) data. This is greatly beneficial for $n \ll \mathrm{p}$ data, as we observed with the TGP dataset; reducing from 1000 to 93 tests. By reducing the dimensional complexity of the analysis this strategy also reduces the burden of multiple testing, but it does not eradicate the issue and necessary adjustments should be made before concluding if PCs are statistically significant. In contrast to classical PCs, we argue that sparse PCs are interpretable since they are linear combinations consisting of small subsets of genes; however biological context is still required to interpret their values. Built from gene expression data, the sparse PCs in this paper are weighted sums of gene expression values, meaning they can be interpreted as accumulated gene expression across the genes involved. Within the same PC, genes with loading values that have opposite signs are negatively correlated, giving additional insight into structures between genes in the same PC. Since our method returns *composite variables*, as opposed to simply reporting groups of similar genes as with cluster analysis [20], this allows for statistical testing of joint associations. Since PCA methods are *unsupervised* (meaning they are not informed by the outcome of interest, such as DILI concern), it is reasonable to expect to find PCs built exclusively from genes that are not differentially expressed. Any of these types of group structures that are found to be associated with the variable of interest potentially offer a brand new source of transcriptomic material for researchers to explore. PC15 in our analysis of the TGP data is example of this.

We have some cautionary notes regarding the selection of tuning parameters when using sparse PCA to simplify data in search of groups of genes, as it can have major

effects on characteristics of the sparse principal components obtained. By choosing λ=3, for example, we committed to identifying groups of genes of a specific size; a median of 18 genes contributed to building the sparse principal components we obtained. For the purpose of demonstrating immense sparseness, this tuning parameter was a good choice. However, perhaps the true transcriptomic structure underlying the TGP gene expression data involves hundreds of correlated genes expressing similarly and as such, perhaps choosing a less sparse tuning parameter would more accurately capture this truth. This subjective choice of tuning parameter is analogous to restricting the number of clusters, k, in the k-means clustering algorithm. Prediction-based cross-validation methods are available to guide the choice of tuning parameters [6] and are typically the default methods built into packages [17]. Bonner [18] suggested to visually inspect the variance-covariance structure of data to observe blocking of variables (genes), but this strategy has yet to be tested with rigour or under realistic data conditions.

The analysis strategy we proposed is a practical way to explore multivariate patterns in large genomic data and suggest if they are worth pursuing. Sparse methodology has many extensions to a variety of data analysis situations and it has potential to benefit research in toxicogenomics.

Acknowledgements

JB would like to acknowledge funding from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Canadian Institutes of Health Research (CIHR) (grant number 84392). JB is the inaugural holder of the John D. Cameron Endowed Chair in the Genetic Determinants of Chronic Diseases, Department of Clinical Epidemiology and Biostatistics, McMaster University.

References

1. Taboureau O, Hersey A, Audouze K, Gautier L, Jacobsen U, Akhtar R, Atkinson F, Overington J, Brunak S. Toxicogenomics Investigation Under the eTOX Project. Pharmacogenomics & Pharmacoproteomics 2012; S7
2. Uehara T, Ono A, Maruyama T, Kato I, Yamada H, Ohno Y, Urushidani T. The Japanese toxicogenomics project: application of toxicogenomics. Mol. Nutr. Food Res. 2010; 54: 218-227.
3. Chen M, Vijay V, Shi Q, Liu Z, Fang H, Tong W. FDA-approved drug labeling for the study of drug-induced liver injury. Drug Discovery Today 2011; 16: 697-703.
4. Afshari C A, Hamadeh H K, Bushel P R. The evolution of bioinformatics in toxicology: advancing toxicogenomics. Toxicological Sciences 2011; 120: S225-S237.
5. Zou H, Hastie T, Tibshirani R. Sparse Principal Component Analysis. Journal of Computational and Graphical Statistics 2006; 15:265-286.
6. Witten D, Tibshirani R, Hastie T. A penalized matrix decomposition, with application to sparse principal components and canonical correlation analysis. Biostatistics 2009; 10:515-534.
7. Lee D, Lee W, Lee Y, Pawitan Y. Super-sparse principal component analysis for high-throughput genomic data. BMC Bioinformatics 2010; 11:296-305.
8. Tibshirani R. Regression shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological) 1996; 58:267-288.
9. Zou H, Hastie T. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society. Series B (Methodological) 2005; 67:301-320.
10. McMillian M, Nie A, Parker J B, Leone A, Kemmerer M, Bryant S, Herlich J, Yieh L, Bittner A, Liu X, et al. Drug-induced oxidative stress in rat liver from a toxicogenomics perspective. Toxicology and Applied Pharmacology 2005; 207: S171-S178.
11. Hochreiter S, Clevert D-A, Obermayer K. A new summarization method for affymetrix probe level data. Bioinformatics 2006; 22: 943-949.
12. R Core Team (2013). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/
13. Smyth G K. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. Statistical Applications in Genetics and Molecular Biology 2004; 3:Article 3.

14. Smyth G K. Limma: linear models for microarray data. In: Bioinformatics and Computational Biology Solutions using R and Bioconductor 2005, R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, W. Huber (eds.), Springer, New York, pages 397-420.
15. Benjamin Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B (Methodological) 1995; 57: 289-300.
16. Storey J D, Tibshirani R. Statistical significance for genomewide studies. PNAS 2003; 100:9440-9445.
17. Witten D, Tibshirani Rob, Gross S, Narasimhan B. PMA: Penalized Multivariate Analysis. R package version 1.0.9 (2013). http://CRAN.R-project.org/package=PMA
18. Bonner A. Sparse principal component analysis for high-dimensional data: a comparative study. Open Access Dissertations and Theses - McMaster 2012.
19. Friedman J, Hastie T, Tibshirani R. Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software 2010; 33: 1-22.
20. Eisen M B, Spellman P T, Brown P O, Botstein D. Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. U.S.A. 1998; 95: 14863-14868.

Chapter 3

3. EVALUATION OF SPARSE CCA FOR HIGH-DIMENSIONAL DATA

Context

My second project is in the form of a manuscript that will be submitted to a peer-reviewed journal. The work presented in this manuscript includes extensive simulation experiments that compare many sparse CCA methods and a real data analysis to showcase the best performing methods. The work was jointly motivated by my experiences with toxicogenomic data from my first project and my understanding of the current state of the literature for sparse CCA.

The TGP database that I used in my first project (Chapter 2) had more data types than I originally analyzed. In addition to the human DILI concern variable and gene expression data, the TGP database included measurements for conventional toxicology assessments on rat liver samples.[66] Integrating this domain of data to the analysis could give additional insight into the relationships between genes and drug toxicity. However, since it consisted of around 40 variables itself, this would involve finding complex relationships between two sets of variables. Considering the natural relationship between PCA and CCA, I decided to investigate sparse CCA in more detail and planned to apply it to the toxicogenomic database.

Several sparse CCA had been published but there did not seem to be compelling evidence as to which was more accurate under *realistic* data conditions. The simulations accompanying methods development, with rare exception, tended to be smaller in scale or not representative of relationship structures apparent in the toxicogenomic data I was dealing with; groups of genes. Before applying sparse PCA to the toxicogenomic data in Chapter 2, I had previously conducted a simulation study comparing three sparse PCA methods, which gave me confidence in selecting a method that would provide accurate results. That simulation work aided me to design a new simulation to compare sparse CCA approaches.[73]

After reviewing the literature for sparse CCA methods, I came across a method by Wilms and Croux, 2015.[52] Although they conducted simulation experiments to test their sparse CCA method against popular competitors, they included extremely sparse solutions which were not indicative of the data prevalent in many genomic applications, including toxicogenomic data. Nevertheless, the structure of their simulation design, along with my previous experience with sparse PCA simulations, provided a good basis

for me to begin my simulation experiments comparing sparse CCA methods. Details of my work are included in the manuscript.

I have prepared a manuscript including my work and plan to submit it to a peer-reviewed journal. After this page, I include our manuscript. Please note that mathematical notation in this manuscript has been simplified to reflect use of only the first canonical components of sparse CCA. Notation is fully described within the contents of the manuscript.

MANUSCRIPT BEGINS ON THE NEXT PAGE…

Evaluating the performance of sparse canonical correlation analysis methods for high-dimensional data with application to toxicogenomics

Ashley Bonner, BSc, MSc
Department of Health Research Methods, Evidence, and Impact; McMaster University

Jemila Hamid, BSc, MSc, PhD
Children's Hospital of Eastern Ontario, Ottawa, Ontario, Canada
Department of Health Research Methods, Evidence, and Impact; McMaster University

Angelo Canty, BSc, MSc, PhD
Department of Mathematics & Statistics; McMaster University

Joseph Beyene*, BSc, MSc, PhD
Department of Health Research Methods, Evidence, and Impact; McMaster University

*
Corresponding Author:
Joseph Beyene
1280 Main Street W, Hamilton, ON, L8S 4L8, Canada
+1 9055259140 x21333
beyene@mcmaster.ca

EVALUATING SPARSE CCA METHODS FOR HIGH-DIMENSIONAL DATA

Abstract

Background: Many sparse canonical correlation analysis (sparse CCA) methods have been proposed in recent years. Each aims to detect and interpret multivariate relationships between high-dimensional 'omic' data domains, potentially elucidating the biological mechanisms underpinning complex traits and disease. However, how the methods perform relative to one another – which one is best – remains unclear.

Methods: We designed simulation experiments to compare the performance (bias, true positive rate, true negative rate, overall sparsity) of several sparse CCA methods. Simulated data were informed by real pathology and gene expression data from a toxicogenomic database. Sparse CCA methods that demonstrated the best simulation performance were applied to the real data to estimate cross-correlated features between domains.

Results: The sparse CCA methods differed markedly in terms of performance. One method was superior to the rest in a majority of data settings, though not optimal across all the evaluated categories. As such, we applied complimentary sparse CCA methods to the toxicogenomic data to estimate cross-correlation and detect cross-correlated features between data domains.

Conclusion: Sparse CCA methods differ significantly in performance depending on the data structure. Our findings from simulations will provide guidance in practical applications and facilitate optimal analysis and interpretation.

3.1. Introduction

Phenotypes are often influenced by a complex system of biological, environmental, and genetic factors. As a result, studies of phenotypes such as cancer, neurodevelopmental disorders, and drug toxicity are measuring and searching through multiple domains of diverse data and, with ongoing advances in technology, it is common for each to hold thousands of variables on the same subjects or samples. Multivariate statistical methods are now required to efficiently and accurately explore, describe, and infer about the mechanisms underlying these phenotypes.

Canonical correlation analysis (CCA) is a classic multivariate statistical method used to estimate correlation between two sets of variables.[1] It does so by constructing linear combinations from each set that are maximally correlated. However, conventional CCA has poor estimation properties and returns biologically unrealistic, uninterpretable relationships when applied to high-dimensional data, and mathematically breaks down (because of singularity) when the number of variables from any dataset exceeds the number of samples (i.e., $p_1 \; or \; p_2 > n$).[2]

In recent years, several extensions to CCA have been developed to overcome these pitfalls, promising a new way to find multivariate relationships between large data domains, even when $p_1 \; or \; p_2 > n$.[3] In particular, sparse regression techniques[4,5] have been fused into the CCA framework via penalty functions to create sparse CCA methods. Sparse CCA has superior estimation properties and, by exceling at variable selection, estimates interpretable models involving sparse subsets of the large number of variables, offering new and multivariate insights into how biological domains interact.[2,6,7]

These tools have gained popularity in imaging genetics[8–10], having been used to explore biological mechanisms underpinning neurologic disorders such as schizophrenia[11], Alzheimer's disease[9], and several others[12], as well as in cancer research[13] and other conditions such as tuberculosis and malaria.[14] The method has also been used to explore a pharmacogenomics study of gemcitabine therapy.[15] The area of toxicogenomics is particularly in need of multivariate integrative methods. As such, the Japanese Toxicogenomics Project (TGP)[16,17], Innomed PredTox[18], and Drug Matrix[19], three of the largest toxicogenomic databases in the world, have been made publicly available to promote data mining with novel statistical analysis methods in hopes to uncover genetic knowledge regarding drug toxicity.[17]

With a multitude of sparse CCA variants emerging over the past decade and increasing demand for integrative applications, it is important to compare the performance of sparse CCA methods under different data conditions. As well, since

sparse CCA methods currently have few strategies for inference, simulations should be guided by real data and tethered to applications in order to instill confidence in the results.

In this paper, we compare the performance of several sparse CCA methods for extracting groups of cross-correlated variables from high-dimensional data. Performance criteria include bias, true positive rate, true negative rate, and overall sparsity of the canonical vectors. We use real data from the publicly available Japanese Toxicogenomic Project (TGP) database to guide our simulations. We then apply the best performing methods to extract rich drug toxicity information from the database. Collections of biomarkers are extracted, potentially leading to new screening tools for future drugs. Differences between the sparse CCA methods are highlighted to provide readers directions regarding which is best to use, under what scenarios.

3.2. Motivating data and sparse CCA description

3.2.1. The Japanese Toxicogenomic Project (TGP) data

We used data from the Japanese Toxicogenomics Project (TGP) [16] that were summarized and provided by the 2013 Conference on the Critical Assessment of Massive Data Analysis (CAMDA; http://dokuwiki.bioinf.jku.at/doku.php). The toxicogenomic data include pathology (hematology, biochemistry) and gene expression measurements taken from liver or kidney samples at different time points after exposure to different doses of over 170 drugs. The TGP database was made publicly available for investigators to apply advanced analysis tactics and discover relationships between genomic data, conventional toxicity parameters, and clinical endpoints (e.g., drug-induced damage to the liver or kidney).[17] For a full description of the TGP dataset, along with the data processing steps we used, we refer the reader to the Appendix.

Our specific data setup involves 40 continuous pathology variables ($\mathbb{X}_1$) and 1000 continuous gene expression variables ($\mathbb{X}_2$) measured on $n = 226$ rat liver samples, of which 96 had been exposed to a drug having increased concern for drug-induced liver injury (DILI; “Most DILI concern”) and the remaining 130 had been exposed to a drug having less to no concern for causing DILI (“Less or no DILI concern”).

Figures 1a and 1b show images of the cross-correlation within and between data domains using all samples. To aid in identifying correlation structure, we reordered variables based on cluster membership, obtained from using a complete linkage hierarchical clustering algorithm.[20] These visuals suggest that strong, multivariate correlations exist between pathology and gene expression variables. This exploration

motivates us to estimate multivariate correlation between data domains using sparse CCA.

Figure 1: *a) Correlation matrix between and across pathology and gene expression variables for our data (left). b) Cross-correlation matrix between pathology and gene expression variables. The grey sidebars are aligned with pathology variables and the green sidebars are aligned with gene expression variables. Darker colors represent stronger correlation. Variables have been ordered based on complete linkage hierarchical clustering.*

3.2.2. Sparse CCA

Sparse CCA is a flexible multivariate tool that can estimate complex multivariate relationships such as those that appear to be present in the TGP data. In this paper, we tested and utilized several sparse CCA methods to estimate relationships between pathology and gene expression data domains. In this section, we briefly describe the general mathematical infrastructure of sparse CCA including inputs, outputs, tuning parameter selection, and statistical inference.

Given $\mathbb{X}_1$ is an $n \times p_1$ matrix of data (e.g., pathology variables) and $\mathbb{X}_2$ an $n \times p_2$ matrix of data (e.g., gene expression), the sample version of sparse CCA seeks to find a $p_1 \times 1$ vector $\boldsymbol{w}_1 = (w_{11}, w_{12}, \dots, w_{1p_1})'$ and a $p_2 \times 1$ vector $\boldsymbol{w}_2 = (w_{11}, w_{12}, \dots, w_{1p_2})'$, which create linear combination variables $\mathbb{X}_1\boldsymbol{w}_1$ and $\mathbb{X}_2\boldsymbol{w}_2$ that are maximally correlated. The precise objective is to maximize correlation $\rho_{12} = Corr(\mathbb{X}_1\boldsymbol{w}_1, \mathbb{X}_2\boldsymbol{w}_2)$ over all possible choices of $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$, subject to constraints that introduce sparsity. The individual weights, w_{1j_1}, $j_1 = 1, \dots, p_1$ and w_{2j_2}, $j_2 = 1, \dots, p_2$, often referred to as

loadings, reflect the contribution of each variable to the correlation, with a larger magnitude reflecting a larger contribution.

The general objective function for the maximum correlation can be written as

$$\underset{\boldsymbol{w}_1,\boldsymbol{w}_2}{\text{maximize}} \left\{ \frac{\boldsymbol{w}_1 \widehat{\boldsymbol{\Sigma}}_{12} \boldsymbol{w}_2}{\sqrt{\boldsymbol{w}_1 \widehat{\boldsymbol{\Sigma}}_{11} \boldsymbol{w}_1} \sqrt{\boldsymbol{w}_2 \widehat{\boldsymbol{\Sigma}}_{22} \boldsymbol{w}_2}} \right\},$$

subject to the constraints

$$\|\boldsymbol{w}_1\|_2^2 = 1, \|\boldsymbol{w}_2\|_2^2 = 1$$
and
$$P_1(\boldsymbol{w}_1) \leq t_1, P_2(\boldsymbol{w}_2) \leq t_2.$$

In the objective function, $\widehat{\boldsymbol{\Sigma}}_{11}$ and $\widehat{\boldsymbol{\Sigma}}_{22}$ represent the sample covariance matrices for $\mathbb{X}_1$ and $\mathbb{X}_2$, respectively, and $\widehat{\boldsymbol{\Sigma}}_{12}$ is the sample cross-covariance matrix between $\mathbb{X}_1$ and $\mathbb{X}_2$. The first pair of constraints simply equates the sum of squared loading values to 1, because solutions are invariant to scaling; the operator $\|.\|_m$ denotes the L_m-norm. The second pair of constraints involves the *penalty functions,* P_1 and P_2 with *tuning parameters,* t_1 and t_2, which together are the defining features of sparse CCA. There are a number of penalty functions available [5] including classic choices such as the 'ridge' [21], least absolute shrinkage and selection operator (LASSO) [4], and elastic net [22], all of which aim to enable estimation under the $p_1, p_2 < n$ scenario and introduce sparsity to the estimated canonical (loading) vectors, preventing extraneous variables from contributing to the canonical variates. The level of sparsity is influenced by the users' choice of tuning parameters, for which there are many approaches, including a variety of grid-search and cross-validation methods.

Solving the above objective function returns the *first pair of estimated canonical vectors* $\widehat{\boldsymbol{w}}_1$ and $\widehat{\boldsymbol{w}}_2$, which are used to calculate the *first pair of estimated canonical variates* $\mathbb{X}_1\widehat{\boldsymbol{w}}_1$ and $\mathbb{X}_2\widehat{\boldsymbol{w}}_2$ (the linear combinations), with *estimated canonical correlation* $\hat{\rho}_{12}$. Subsequent pairs of canonical vectors that correspond to having the second, third, and so on, maximum correlation can be obtained by using deflated matrices. [7]. Several methods can provide the full set of canonical components (correlations, vectors), including solving eigenvector equations [23] or a singular value decomposition [6], as well as alternating regression procedures [24–27] and other means.[28]

The choice of penalty function, approach to select tuning parameters, and method to obtain solutions all contribute to defining a sparse CCA method. In this paper we adapt the work from Wilms and Croux [29] and compare the performance of several

sparse CCA methods. We use traditional CCA [1] (which we denote by **trcca**); ridge CCA by Vinod et al., 1976 [21] with tuning parameters selected via cross-validation maximizing test-sample canonical correlation (**ridge.cv**); sparse CCA by Parkhomenko et al., 2009 [6] with tuning parameters selected via cross-validation maximizing test-sample canonical correlation (**parkh.cv**); sparse CCA by Witten et al., 2009 [7] with tuning parameters selected via a permutation method described by the authors (**witte.au**) or via cross-validation maximizing test-sample canonical correlation (**witte.cv**); and sparse CCA by Wilms & Croux, 2015 [29] with tuning parameter minimizing Bayes' Information Criterion (**wilms.au**) or via cross-validation maximizing test-sample canonical correlation (**wilms.cv**). Details for each method can be found in the original papers. As initiated by Wilms & Croux, the consistent cross-validation approach (suffix **.cv**) allows fair comparison between CCA methods, but for this paper, we are primarily interested in finding the optimal combination of tuning parameter selection method and CCA method for accurately estimating multivariate correlation.

Analysis is performed in the R statistical package and all codes are freely available as supplementary materials. They include modified versions of codes produced by Wilms and Croux, 2015 as well as additional functions to aid in the simulations and to facilitate graphical presentations.

3.3. Simulations

In this section, we describe the simulation design and the methods for performance evaluation, followed by results and conclusions regarding which methods are most appropriate for applications involving high-dimensional data, similar to that of the TGP data analyzed and presented in this paper.

3.3.1. Simulation design

Our general simulation strategy consists of five steps:

1. Define true data structure by specifying values for the parameters n, p_1, p_2, Σ.
2. Calculate the true canonical vectors and true canonical correlations using the true covariance matrix Σ.
3. Generate 1000 pairs of data $[\mathbb{X}_1^r, \mathbb{X}_2^r]$, for $r = 1, \ldots, 1000$.
4. Apply sparse CCA to each of the 1000 pairs of data to calculate estimated canonical vectors and canonical correlations, using the estimated covariance matrices.

5. Evaluate the performance of the different sparse CCA methods by comparing estimated and true outputs.

In step 1, to define simulated data structure, we first explored the TGP data and set the parameter values so as our simulated data mimics the structure of the TGP data. We used scenarios with $p_1 = 40$ and $p_2 = 1000$ variables in simulations, and varied sample size $n = 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 300, 500, 1000$ to cover a range of high-dimensionality. For the covariance structure, we used the estimated covariance structure from the TGP data (presented in Figures 1a and 1b), as well as covariance estimates from selected dose-time subgroups to allow variations in the covariance structure of the simulated data. Instead of using the exact estimated covariance matrices for Σ in our simulations, we used simplified, sparse covariance structures to represent the underlying structure.

The structures presented in Figures 1a and 1b reveal blocks of correlated variables in both data domains $\mathbb{X}_1$ and $\mathbb{X}_2$, and varying degrees of correlation between domains. As such, in our simulations, we assumed a block-diagonal covariance structure for each data domain, and assumed variables within those correlated groups were cross-correlated between domains. The specific formulation of Σ we used is

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

where Σ_{11} is the $p_1 \times p_1 = 40 \times 40$ covariance matrix for variables from $\mathbb{X}_1$, Σ_{22} is the $p_2 \times p_2 = 1000 \times 1000$ covariance matrix for variables from $\mathbb{X}_2$, and $\Sigma_{12} = \Sigma_{21}^T$ is the $p_1 \times p_2 = 40 \times 1000$ cross-covariance matrix between $\mathbb{X}_1$ and $\mathbb{X}_2$. We defined

$$\Sigma_{11} = \begin{bmatrix} \Sigma_{11,1} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Sigma_{11,2} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Sigma_{11,M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \Sigma_{11,e} \end{bmatrix}, \quad \Sigma_{22} = \begin{bmatrix} \Sigma_{22,1} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Sigma_{22,2} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Sigma_{22,M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \Sigma_{22,e} \end{bmatrix},$$

where we set M to be the number of groups of variables that are cross-correlated between $\mathbb{X}_1$ and $\mathbb{X}_2$. Our simulation design allows group sizes $p_{1,m}$ $(m = 1, \dots, M)$ and $p_{2,m}$ $(m = 1, \dots, M)$ to vary and $p_{1,e} = p_1 - \sum_{m=1}^{M} p_{1,m}$ and $p_{2,e} = p_2 - \sum_{m=1}^{M} p_{2,m}$ denote the number of uncorrelated 'noise' variables for the corresponding data domains.

These group sizes are the dimensions for the corresponding covariance blocks which are defined simply as

$$\Sigma_{11,m} = \sigma_{11,m}^2 \begin{bmatrix} 1 & c_{11,m} & \dots & c_{11,m} \\ c_{11,m} & 1 & \dots & c_{11,m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{11,m} & c_{11,m} & \dots & 1 \end{bmatrix}, \quad \Sigma_{22,m} = \sigma_{22,m}^2 \begin{bmatrix} 1 & c_{22,m} & \dots & c_{22,m} \\ c_{22,m} & 1 & \dots & c_{22,m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{22,m} & c_{22,m} & \dots & 1 \end{bmatrix},$$

where $\sigma_{11,m}^2$ is the common variance and $c_{11,m}$ is the common correlation shared by variables in the m^{th} group in $\mathbb{X}_1$ ($\sigma_{22,m}^2$ and $c_{11,m}$ for $\mathbb{X}_2$). The ungrouped 'noise' variables have $\Sigma_{11,e} = \sigma_{11,e}^2 \boldsymbol{I}$, $\Sigma_{22,e} = \sigma_{22,e}^2 \boldsymbol{I}$. We define the cross-covariance matrix between $\mathbb{X}_1$ and $\mathbb{X}_2$ as

$$\Sigma_{12} = \Sigma_{21}^T = \begin{bmatrix} \Sigma_{12,1} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Sigma_{12,2} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Sigma_{12,M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

with

$$\Sigma_{12,m} = \sigma_{11,m}\sigma_{22,m}c_{12,m}\mathbf{1}_{p_{1,m},p_{2,m}},$$

where $\mathbf{1}_{a,b}$ is a matrix of 1's with dimensions $a \times b$. This means we have assumed that for each pair of correlated groups, all pairwise correlations between groups are equal.

Table 1 lists the parameters used to define Σ as well as specific choice of values used in the simulation, which are used to generate various scenarios. We included one low-dimensional scenario (Scenario s = 1) and four high-dimensional scenarios (Scenario s = 2, 3, 4, 5). The low-dimensional scenario was included to show performance on a smaller scale, but is of lesser importance. For each high-dimensional scenario, we fixed $p_1 = 40$ and $p_2 = 1000$ and chose different combinations of group size and correlation to reflect the TGP data. The designed correlation matrices for these high-dimensional scenarios are presented in Figure 2.

*** Table 1 APPROXIMATELY HERE ***

Table 1: Simulation scenarios used.

Scenario (s)	M	$p_{1,1}, p_{1,e}$	$\sigma^2_{11,1}, \sigma^2_{11,e}$	$c_{11,1}, c_{11,M}$	$p_{2,1}, p_{2,e}$	$\sigma^2_{22,1}, \sigma^2_{22,e}$	$c_{22,1}, c_{22,M}$	$c_{12,1}, c_{12,M}$
1 (Low-dimensional)	1	5, 5	10, 5	0.5, 0	10, 10	10, 5	0.5, 0	0.5, 0
2 (Primary)	1	10, 30	10, 5	0.5, 0	100, 900	10, 5	0.5, 0	0.5, 0
3 (Smaller group size)	1	5, 35	10, 5	0.5, 0	50, 950	10, 5	0.5, 0	0.5, 0
4 (Less cross-corr)	1	10, 30	10, 5	0.5, 0	100, 900	10, 5	0.5, 0	0.4, 0
5 (Both)	1	5, 35	10, 5	0.5, 0	50, 950	10, 5	0.5, 0	0.4, 0

*** Figure 2 APPROXIMATELY HERE ***

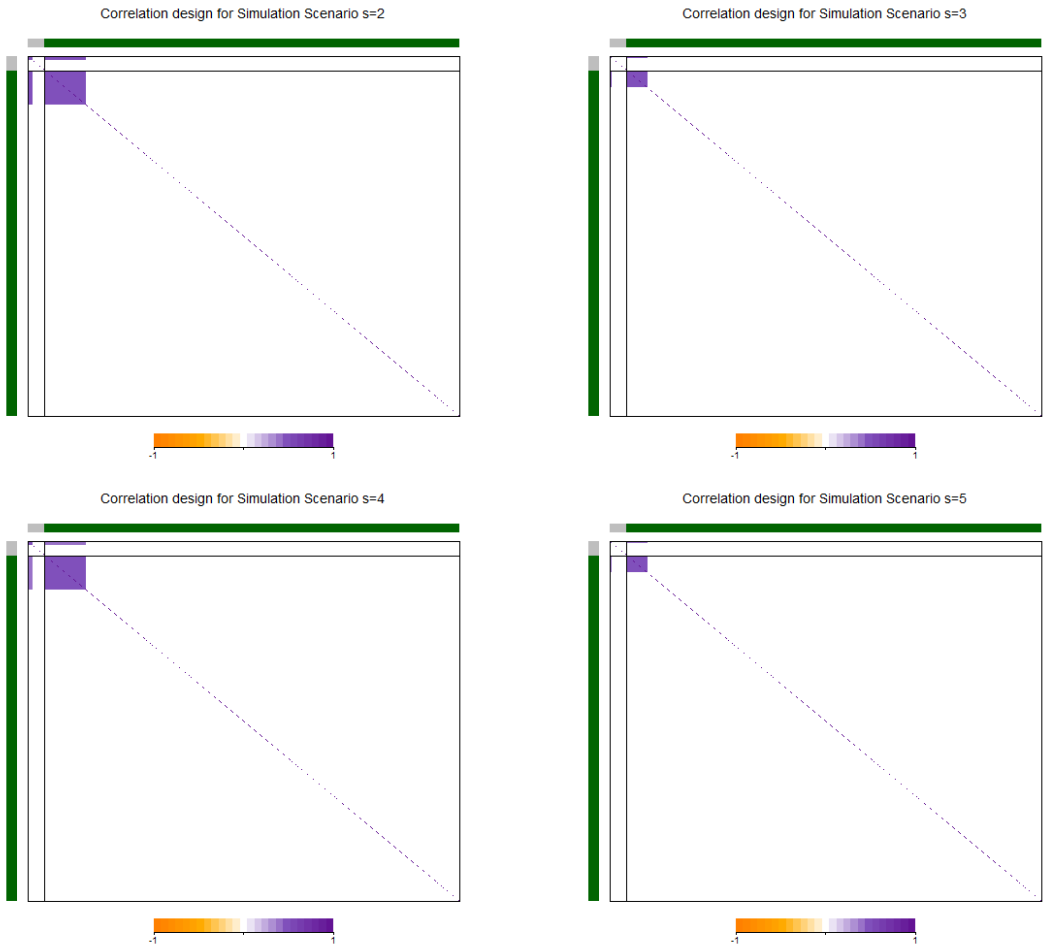


Figure 2: Correlation matrices designed for our simulation study.

Each scenario had $M = 1$ group and the same distribution of group variances, which seemed reasonable given most covariance images showed two groups and the core objective function for sparse CCA involves correlation, not covariance. For each scenario, we tried all 13 sample sizes from our design.

In step 2, to obtain true canonical components, we compute the singular value decomposition of $K = \Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1/2} = \boldsymbol{UDV}^T$, where $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_M]$ and $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_M]$ are the left and right singular vectors of K and $\boldsymbol{D} = diag(\boldsymbol{d}) = diag([d_1, d_2, \dots, d_M]^T)$ are the singular values of K. These relate to the quantities of interest from CCA. The true canonical vectors are $\boldsymbol{W}_1 = \Sigma_{11}^{-1/2}\boldsymbol{U} = [\boldsymbol{w}_{1,1}, \boldsymbol{w}_{1,2}, \dots, \boldsymbol{w}_{1,M}]$ and $\boldsymbol{W}_2 = \Sigma_{22}^{-1/2}\boldsymbol{V} = [\boldsymbol{w}_{2,1}, \boldsymbol{w}_{2,2}, \dots, \boldsymbol{w}_{2,M}]$, and the true canonical correlations are $\boldsymbol{\rho}_{12} = \boldsymbol{d} = [\rho_{12,1}, \rho_{12,2}, \dots, \rho_{12,M}]$. This notation accommodates subsequent canonical components, but we simply investigated the first set: $\boldsymbol{w}_1 = \boldsymbol{w}_{1,1}$, $\boldsymbol{w}_2 = \boldsymbol{w}_{2,1}$, and $\rho_{12} = \rho_{12,1}$. After obtaining $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ we scaled each canonical vector $\boldsymbol{w}_{1,m}$ by its magnitude $\|\boldsymbol{w}_{1,m}\|$ to achieve the CCA condition $\|\boldsymbol{w}_{1,m}\|_2^2 = 1$ for all $m = 1, \dots, M$ (same for $\boldsymbol{w}_{2,m}$).

In step 3, we generated the n samples from a multivariate normal distribution with mean $\boldsymbol{\mu} = \boldsymbol{0}$ and covariance Σ. For step 4, we applied each of the CCA methods (**trcca**, **ridge.cv**, **parkh.cv**, **witte.au**, **witte.cv**, **wilms.au**, **wilms.cv**) to each of the R sets of data using functions we adapted from code made available by Wilms and Croux, 2015.[29] The tuning parameter ranges we supplied to the tuning parameter selection processes are presented in Table 2. We chose ranges that spanned the possible tuning parameter space (as many levels of sparsity as possible), to allow the tuning parameter selection process to objectively choose which of the methods resulted in the best fit to the data.

*** Table 2 APPROXIMATELY HERE ***

Table 2: Tuning parameter ranges used. For some methods, we used a two-stage grid search to speed up the selection process.

Method	Tuning parameter	Stage 1 range	Stage 2 resolution
trcca	NA	NA	NA
ridge.cv	ridge parameter	0.001, 0.25, 0.50, 0.75, 1	None
parkh.cv	soft-threshold	0, 0.02, 0.04, …, 2	None
witte.au	lasso proportion of OLS	0, 0.05, 0.10, …, 1	0.01
witte.cv	lasso proportion of OLS	0, 0.05, 0.10, …, 1	0.01
wilms.au	lasso proportion of OLS	0, 0.05, 0.10, …, 1	0.01
wilms.cv	lasso proportion of OLS	0, 0.05, 0.10, …, 1	0.01

Ideally, for a grid of tuning parameters, this would mean extremely high resolution. To minimize computation time, where practical, we integrated a two-stage tuning

parameter selection process. For example, for Witten et al., 2009, we covered the full theoretical range of tuning parameters with 0.05 granularity, using (0, 0.05, 0.10,…, 0.95, 1) in the first stage, followed by a second search with resolution of 0.01 around whichever tuning parameter was selected from the first stage. Notably, we used less granular range for **ridge.cv,** because it had much longer computation times than the other methods and, since it does not result in sparsity, which is of secondary importance in this paper. Other parameters for the methods, such as convergence thresholds and the maximum number of iterations until convergence were set to code defaults.

For each CCA method we extracted the estimated canonical correlations and estimated canonical vectors, along with additional summaries to assist with performance evaluation.

In step 5, we summarized the results across simulation iterations and compared them to the designed truth. We assessed the performance of the methods using several criteria. For canonical correlations we tracked bias (BIAS) and for canonical vectors we tracked the true positive rate (TPR), being the proportion of the $p_{1,m}$ (and $p_{2,m}$) truly cross-correlated variables that were correctly estimated to have non-zero contribution in $\boldsymbol{w}_{1,m}$ (and $\boldsymbol{w}_{2,m}$); true negative rate (TNR), being the proportion of the $p_1 - p_{1,m}$ (and $p_2 - p_{2,m}$) truly not cross-correlated variables that were correctly estimated to have zero contribution in $\boldsymbol{w}_{1,m}$ (and $\boldsymbol{w}_{2,m}$); and total number of non-zeros (NNZ) for an indicator of overall sparsity.

3.3.2. Simulation Results

All simulation results for first canonical components are summarized numerically and visually in Figures 3 and 4a through 5c. Each figure contains a grid of cells, where each cell contains a numerically summarized performance measure, in each case the average, across the 1000 simulation iterations corresponding to a simulation scenario (row of the grid) and CCA method (column of the grid). We include Scenario 1 but focus interpretation on high-dimensional scenarios because application to high-dimensional data is the goal in this paper.

Figure 3 reports the mean bias found while estimating canonical correlation for the first pair of canonical variates. All methods seem to overestimate canonical correlation (i.e., positive bias) in scenarios with lower sample sizes. As sample size approaches 1000, most methods appear to become unbiased, with **parkh.cv** and **witte.cv** realizing this asset sooner than **wilms.au** and **wilms.cv**. The **witte.au** method begins to underestimate canonical correlation at the larger sample sizes. Judging

performance entirely based on bias of the canonical correlation, it appears reasonable to suggest that the methods perform similarly, with a slight preference to **parkh.cv**.

*** Figure 3 APPROXIMATELY HERE ***

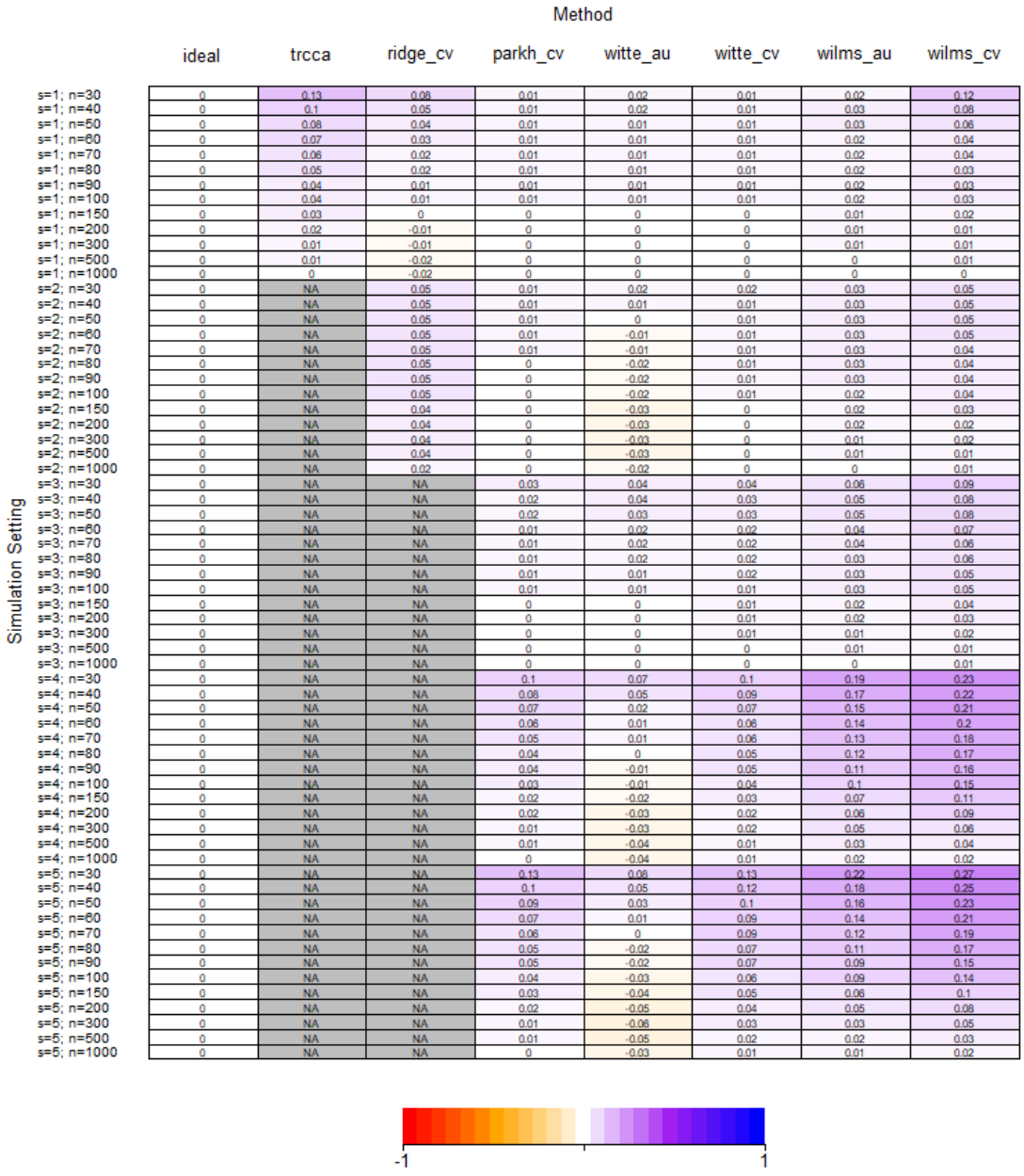


Figure 3: A presentation of simulation results pertaining to bias of canonical correlation values from the first set of canonical variates (in the cells) across simulation scenarios

(rows) and CCA methods (columns). Column 1 contains the ideal value of bias, which is 0. All other cells contain a mean bias across R=1000 simulation runs for the scenario and CCA method corresponding to the cells' location. The color scale is used to assist reading the table, with white being the ideal case where bias is 0, and cells closer to red corresponding to more (positive or negative) bias.

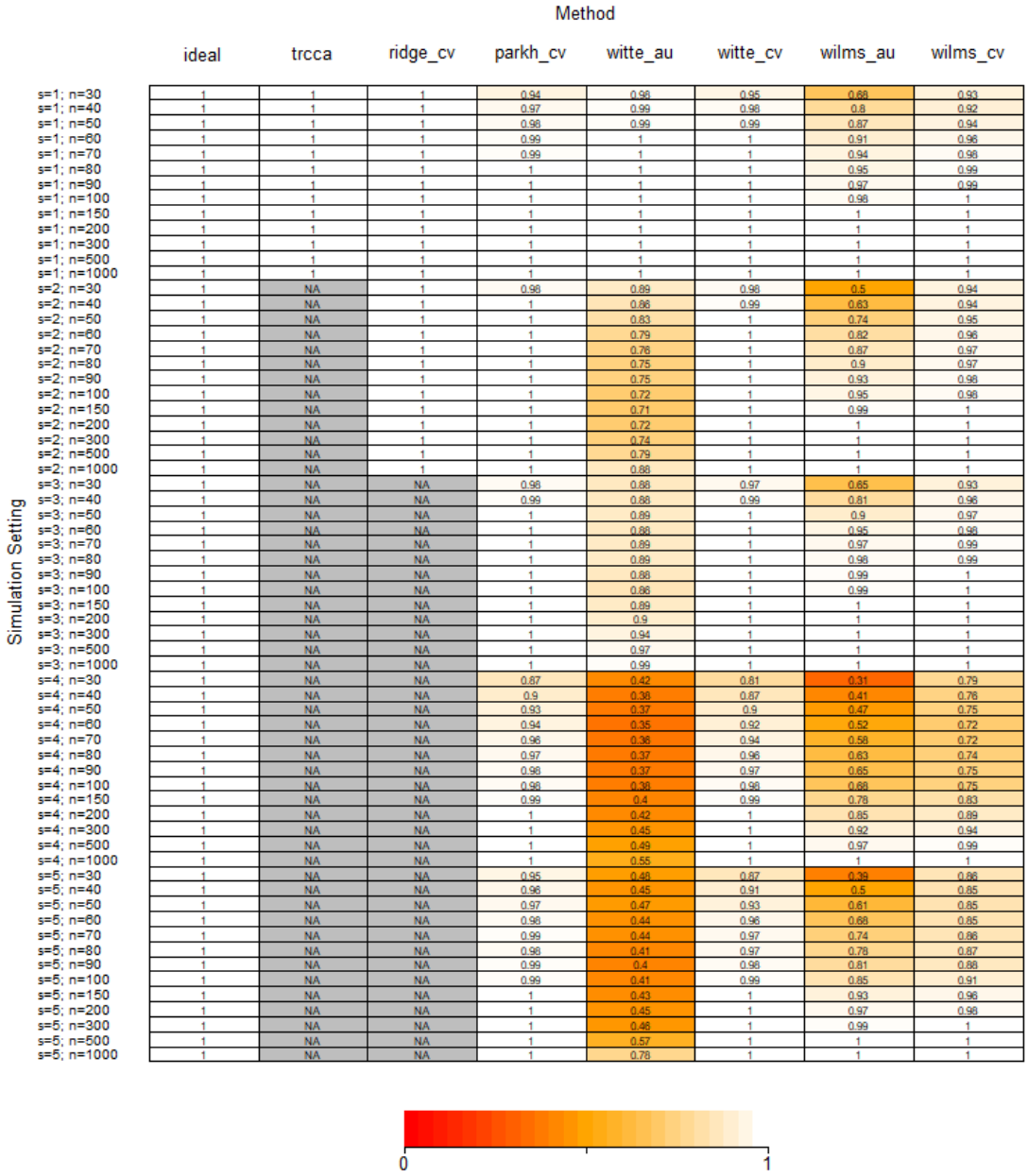
Next we consider summaries of accuracy for the first pair of canonical vectors, where accuracy is measured by TPR and TNR. Figures 4a and 4b reports the TPR and TNR, respectively, of $\boldsymbol{w}_1$; the first canonical vector for the smaller set of data $\mathbb{X}_1$ (mimicking pathology data). To compliment these results, we report overall sparsity (NNZ) in Figure 4c. The ideal TPR and TNR values are 1, but the ideal NNZ value depends on the simulation scenarios we designed.

*** Figure 4a APPROXIMATELY HERE ***

*** Figure 4b APPROXIMATELY HERE ***

*** Figure 4c APPROXIMATELY HERE ***

Overall, we would recommend the **parkh.cv** method for applying to data of similar structure to our simulation scenarios. Compared to the other methods, it was relatively unbiased in estimating canonical correlation and quickly obtained perfect TPR as sample size increased, accepting only a small proportion of variables that were not truly cross-correlated. It also has a practical benefit in being the fastest method to execute. However, at sample sizes $n \geq 200$, we recommend **wilms.au**, especially for lower cross-correlations. At very large sample sizes of $n \geq 1000$, **wilms.cv** could be used as well.



Simulation Setting	ideal	trcca	ridge_cv	parkh_cv	witte_au	witte_cv	wilms_au	wilms_cv
s=1; n=30	1	1	1	0.94	0.98	0.95	0.68	0.93
s=1; n=40	1	1	1	0.97	0.99	0.98	0.8	0.92
s=1; n=50	1	1	1	0.98	0.99	0.99	0.87	0.94
s=1; n=60	1	1	1	0.99	1	1	0.91	0.96
s=1; n=70	1	1	1	0.99	1	1	0.94	0.98
s=1; n=80	1	1	1	1	1	1	0.95	0.99
s=1; n=90	1	1	1	1	1	1	0.97	0.99
s=1; n=100	1	1	1	1	1	1	0.98	1
s=1; n=150	1	1	1	1	1	1	1	1
s=1; n=200	1	1	1	1	1	1	1	1
s=1; n=300	1	1	1	1	1	1	1	1
s=1; n=500	1	1	1	1	1	1	1	1
s=1; n=1000	1	1	1	1	1	1	1	1
s=2; n=30	1	NA	1	0.98	0.89	0.98	0.5	0.94
s=2; n=40	1	NA	1	1	0.86	0.99	0.63	0.94
s=2; n=50	1	NA	1	1	0.83	1	0.74	0.95
s=2; n=60	1	NA	1	1	0.79	1	0.82	0.96
s=2; n=70	1	NA	1	1	0.76	1	0.87	0.97
s=2; n=80	1	NA	1	1	0.75	1	0.9	0.97
s=2; n=90	1	NA	1	1	0.75	1	0.93	0.98
s=2; n=100	1	NA	1	1	0.72	1	0.95	0.98
s=2; n=150	1	NA	1	1	0.71	1	0.99	1
s=2; n=200	1	NA	1	1	0.72	1	1	1
s=2; n=300	1	NA	1	1	0.74	1	1	1
s=2; n=500	1	NA	1	1	0.79	1	1	1
s=2; n=1000	1	NA	1	1	0.86	1	1	1
s=3; n=30	1	NA	NA	0.98	0.88	0.97	0.65	0.93
s=3; n=40	1	NA	NA	0.99	0.88	0.99	0.81	0.96
s=3; n=50	1	NA	NA	1	0.89	1	0.9	0.97
s=3; n=60	1	NA	NA	1	0.88	1	0.95	0.98
s=3; n=70	1	NA	NA	1	0.89	1	0.97	0.99
s=3; n=80	1	NA	NA	1	0.89	1	0.98	0.99
s=3; n=90	1	NA	NA	1	0.88	1	0.99	1
s=3; n=100	1	NA	NA	1	0.86	1	0.99	1
s=3; n=150	1	NA	NA	1	0.89	1	1	1
s=3; n=200	1	NA	NA	1	0.9	1	1	1
s=3; n=300	1	NA	NA	1	0.94	1	1	1
s=3; n=500	1	NA	NA	1	0.97	1	1	1
s=3; n=1000	1	NA	NA	1	0.99	1	1	1
s=4; n=30	1	NA	NA	0.87	0.42	0.81	0.31	0.79
s=4; n=40	1	NA	NA	0.9	0.38	0.87	0.41	0.76
s=4; n=50	1	NA	NA	0.93	0.37	0.9	0.47	0.75
s=4; n=60	1	NA	NA	0.94	0.35	0.92	0.52	0.72
s=4; n=70	1	NA	NA	0.96	0.36	0.94	0.58	0.72
s=4; n=80	1	NA	NA	0.97	0.37	0.96	0.63	0.74
s=4; n=90	1	NA	NA	0.98	0.37	0.97	0.65	0.75
s=4; n=100	1	NA	NA	0.98	0.38	0.98	0.68	0.75
s=4; n=150	1	NA	NA	0.99	0.4	0.99	0.78	0.83
s=4; n=200	1	NA	NA	1	0.42	1	0.85	0.89
s=4; n=300	1	NA	NA	1	0.45	1	0.92	0.94
s=4; n=500	1	NA	NA	1	0.49	1	0.97	0.99
s=4; n=1000	1	NA	NA	1	0.55	1	1	1
s=5; n=30	1	NA	NA	0.95	0.46	0.87	0.39	0.86
s=5; n=40	1	NA	NA	0.96	0.45	0.91	0.5	0.85
s=5; n=50	1	NA	NA	0.97	0.47	0.93	0.61	0.85
s=5; n=60	1	NA	NA	0.98	0.44	0.96	0.68	0.85
s=5; n=70	1	NA	NA	0.99	0.44	0.97	0.74	0.86
s=5; n=80	1	NA	NA	0.98	0.41	0.97	0.78	0.87
s=5; n=90	1	NA	NA	0.99	0.4	0.96	0.81	0.88
s=5; n=100	1	NA	NA	0.99	0.41	0.99	0.85	0.91
s=5; n=150	1	NA	NA	1	0.43	1	0.93	0.96
s=5; n=200	1	NA	NA	1	0.45	1	0.97	0.98
s=5; n=300	1	NA	NA	1	0.46	1	0.99	1
s=5; n=500	1	NA	NA	1	0.57	1	1	1
s=5; n=1000	1	NA	NA	1	0.78	1	1	1

Figure 4a: A presentation of simulation results: the true positive rate (TPR) of the first canonical vector for $\mathbb{X}_1$ (in the cells), across simulation scenarios (rows) and CCA methods (columns). Column 1 contains the ideal value of TPR, which is 1. All other cells contain a mean TPR across R=1000 simulation runs for the scenario and CCA method corresponding to the cells' location. The color scale is used to assist reading the table, with white being the ideal case where TPR is 1, and cells closer to red corresponding to reduced TPR.

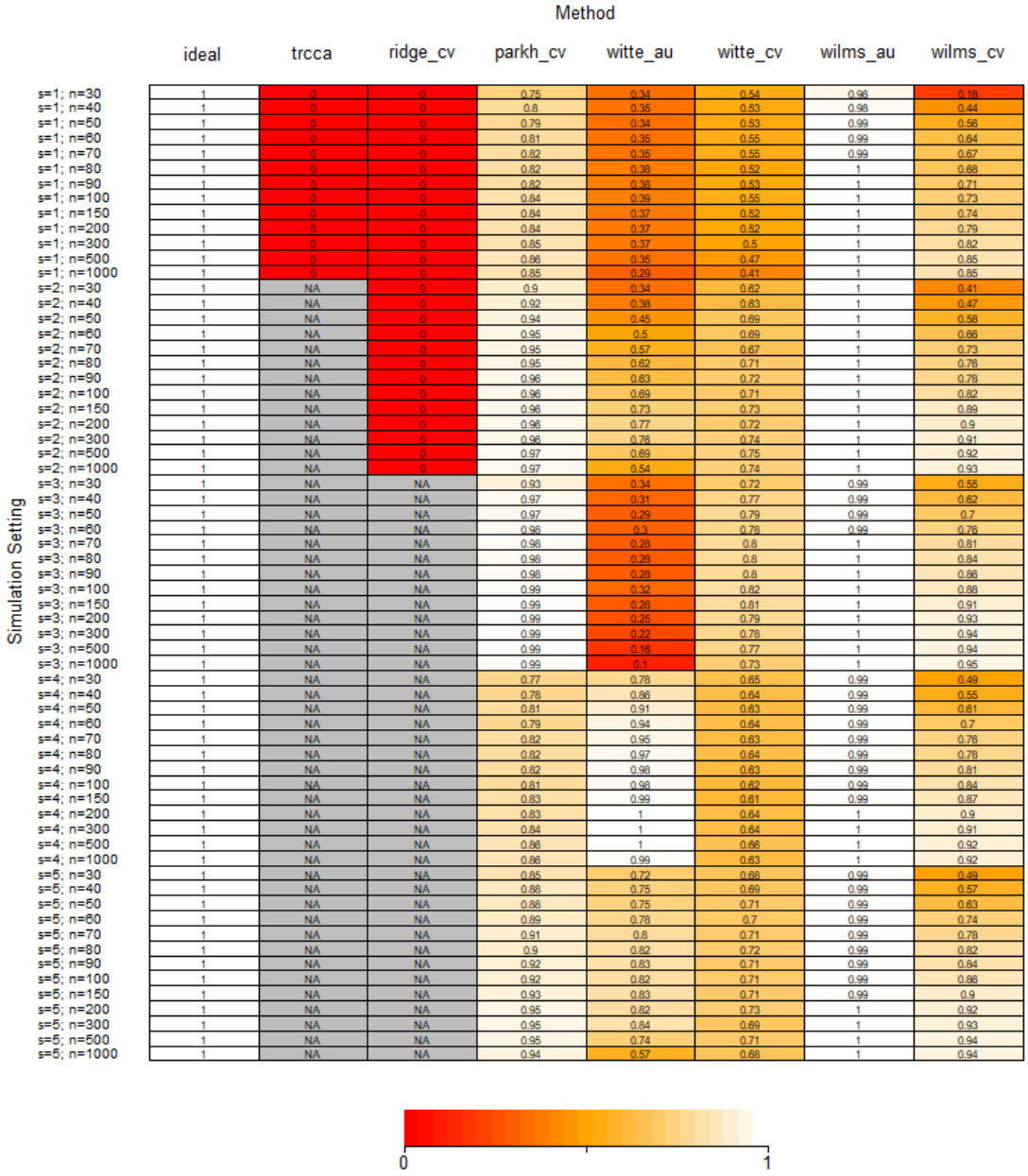


Figure 4b: A presentation of simulation results: the true negative rate (TNR) of the first canonical vector for $\mathbb{X}_1$ (in the cells), across simulation scenarios (rows) and CCA methods (columns). Column 1 contains the ideal value of TNR, which is 1. All other cells contain a mean TNR across R=1000 simulation runs for the scenario and CCA method corresponding to the cells' location. The color scale is used to assist reading the table, with white being the ideal case where TNR is 1, and cells closer to red corresponding to reduced TNR.

Simulation Setting	Method							
	ideal	trcca	ridge_cv	parkh_cv	witte_au	witte_cv	wilms_au	wilms_cv
s=1; n=30	5	10	10	5.9	8.2	7.1	3.6	8.8
s=1; n=40	5	10	10	5.9	8.2	7.3	4.1	7.4
s=1; n=50	5	10	10	6	8.3	7.3	4.4	6.9
s=1; n=60	5	10	10	5.9	8.2	7.2	4.6	6.6
s=1; n=70	5	10	10	5.9	8.2	7.2	4.7	6.5
s=1; n=80	5	10	10	5.9	8.1	7.4	4.8	6.5
s=1; n=90	5	10	10	5.9	8.1	7.3	4.9	6.4
s=1; n=100	5	10	10	5.8	8.1	7.3	4.9	6.3
s=1; n=150	5	10	10	5.8	8.1	7.4	5	6.3
s=1; n=200	5	10	10	5.8	8.2	7.4	5	6.1
s=1; n=300	5	10	10	5.7	8.2	7.5	5	5.9
s=1; n=500	5	10	10	5.7	8.2	7.7	5	5.8
s=1; n=1000	5	10	10	5.7	8.5	7.9	5	5.7
s=2; n=30	10	NA	40	12.7	28.8	21	5.1	27.1
s=2; n=40	10	NA	40	12.3	27.2	21.2	6.3	25.4
s=2; n=50	10	NA	40	11.9	24.7	19.4	7.4	22.1
s=2; n=60	10	NA	40	11.5	22.9	19.2	8.2	19.8
s=2; n=70	10	NA	40	11.6	20.6	19.9	8.7	17.9
s=2; n=80	10	NA	40	11.5	19	18.8	9	16.9
s=2; n=90	10	NA	40	11.2	18.4	18.4	9.3	16.5
s=2; n=100	10	NA	40	11.1	16.6	18.6	9.5	15.2
s=2; n=150	10	NA	40	11.1	15.2	18.2	9.9	13.4
s=2; n=200	10	NA	40	11.1	14.1	18.4	10	13
s=2; n=300	10	NA	40	11.1	14.6	17.8	10	12.7
s=2; n=500	10	NA	40	11	17.2	17.5	10	12.5
s=2; n=1000	10	NA	40	10.9	22.6	17.7	10	12
s=3; n=30	5	NA	NA	7.4	27.7	14.8	3.5	20.4
s=3; n=40	5	NA	NA	6	28.4	12.9	4.3	18
s=3; n=50	5	NA	NA	6	29.2	12.5	4.7	15.3
s=3; n=60	5	NA	NA	5.8	28.9	12.5	4.9	13.2
s=3; n=70	5	NA	NA	5.7	29.6	12.1	5	11.8
s=3; n=80	5	NA	NA	5.7	29.7	11.9	5	10.4
s=3; n=90	5	NA	NA	5.5	29.5	12.1	5	9.8
s=3; n=100	5	NA	NA	5.5	28.1	11.1	5	9.1
s=3; n=150	5	NA	NA	5.4	29.6	11.6	5	8
s=3; n=200	5	NA	NA	5.4	30.7	12.3	5	7.5
s=3; n=300	5	NA	NA	5.4	31.9	12.6	5	7.3
s=3; n=500	5	NA	NA	5.4	34.3	13.1	5	7.2
s=3; n=1000	5	NA	NA	5.3	36.5	14.4	5	6.8
s=4; n=30	10	NA	NA	15.6	10.8	18.6	3.3	23.1
s=4; n=40	10	NA	NA	15.5	7.9	19.5	4.3	21.1
s=4; n=50	10	NA	NA	14.9	6.3	20.2	5.1	19.3
s=4; n=60	10	NA	NA	15.8	5.2	20	5.5	16.2
s=4; n=70	10	NA	NA	14.9	5.2	20.5	6.1	14.3
s=4; n=80	10	NA	NA	15	4.5	20.6	6.6	14.1
s=4; n=90	10	NA	NA	15.1	4.3	20.9	6.8	13.2
s=4; n=100	10	NA	NA	15.5	4.4	21.2	7	12.3
s=4; n=150	10	NA	NA	15.2	4.1	21.5	8	12.1
s=4; n=200	10	NA	NA	15	4.3	20.9	8.6	11.7
s=4; n=300	10	NA	NA	14.9	4.6	20.8	9.2	12.1
s=4; n=500	10	NA	NA	14.2	5	20.2	9.8	12.4
s=4; n=1000	10	NA	NA	14.3	5.7	21	10	12.4
s=5; n=30	5	NA	NA	9.9	12.2	15.5	2.2	22.3
s=5; n=40	5	NA	NA	8.9	10.9	15.4	2.9	19.4
s=5; n=50	5	NA	NA	9	11	14.7	3.4	17
s=5; n=60	5	NA	NA	8.7	10.1	15.4	3.9	13.3
s=5; n=70	5	NA	NA	8.2	9.4	15.1	4.1	11.9
s=5; n=80	5	NA	NA	8.3	8.3	14.7	4.4	10.8
s=5; n=90	5	NA	NA	7.6	7.8	15.1	4.5	10
s=5; n=100	5	NA	NA	7.7	8.2	15.1	4.7	9.6
s=5; n=150	5	NA	NA	7.6	8.2	15.2	4.9	8.3
s=5; n=200	5	NA	NA	6.8	8.4	14.4	5	7.8
s=5; n=300	5	NA	NA	6.8	7.9	15.9	5	7.6
s=5; n=500	5	NA	NA	6.7	12.1	15.3	5	7.2
s=5; n=1000	5	NA	NA	7.1	19.1	16.3	5	6.9

Figure 4c: A presentation of simulation results: the number of non-zero values (NNZ), or overall sparsity, in the first canonical vector for $\mathbb{X}_1$ (in the cells), across simulation scenarios (rows) and CCA methods (columns). Column 1 contains the ideal value of NNZ, which is the designed group size corresponding to the simulation scenario; see Table 1. All other cells contain a mean NNZ across R=1000 simulation runs for the scenario and CCA method corresponding to the cells' location. No color has been used here.

Now we examine the accuracy results for $\boldsymbol{w}_2$; the first canonical vector for $\mathbb{X}_2$, the larger set of data (mimicking gene expression). Figure 5a, 5b, and 5c present the

TPR, TNR, and NNZ, respectively. For the most part, relative performance of the methods is the same as it was when comparing estimation accuracy of $\boldsymbol{w}_1$. However, the larger number of $p_2 = 1000$ variables emphasized some challenges.

*** Figure 5a APPROXIMATELY HERE ***

*** Figure 5b APPROXIMATELY HERE ***

*** Figure 5c APPROXIMATELY HERE ***

Overall, for the higher-dimensional data $\mathbb{X}_2$ (mimicking gene expression), considering all sample sizes, **parkh.cv** would be the most trustworthy. It was able to capture almost all important variables while imposing a moderate level of sparsity, eliminating a minimum of 71% of the 1000 variables that were truly not cross-correlated. At larger sample sizes and higher correlations ($s = 2{,}3;\ n \geq 200$), both **wilms.au** and **wilms.cv** were able to pick out between 38% and 55% of the important variables without including any unimportant variables, which could supply complimentary information to the somewhat under-sparse **parkh.cv** method.

Finally, since estimation for $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ is handled simultaneously with these methods, we need to select methods that work well with estimating both. For this reason, we ultimately choose **parkh.cv** to guide our analysis of the TGP data, using the strictly sparse **wilms.au** as a complimentary method to strengthen our conclusions regarding which variables are cross-correlated. The **wilms.cv** method would be suitable to include if we had $n \geq 1000$ samples.

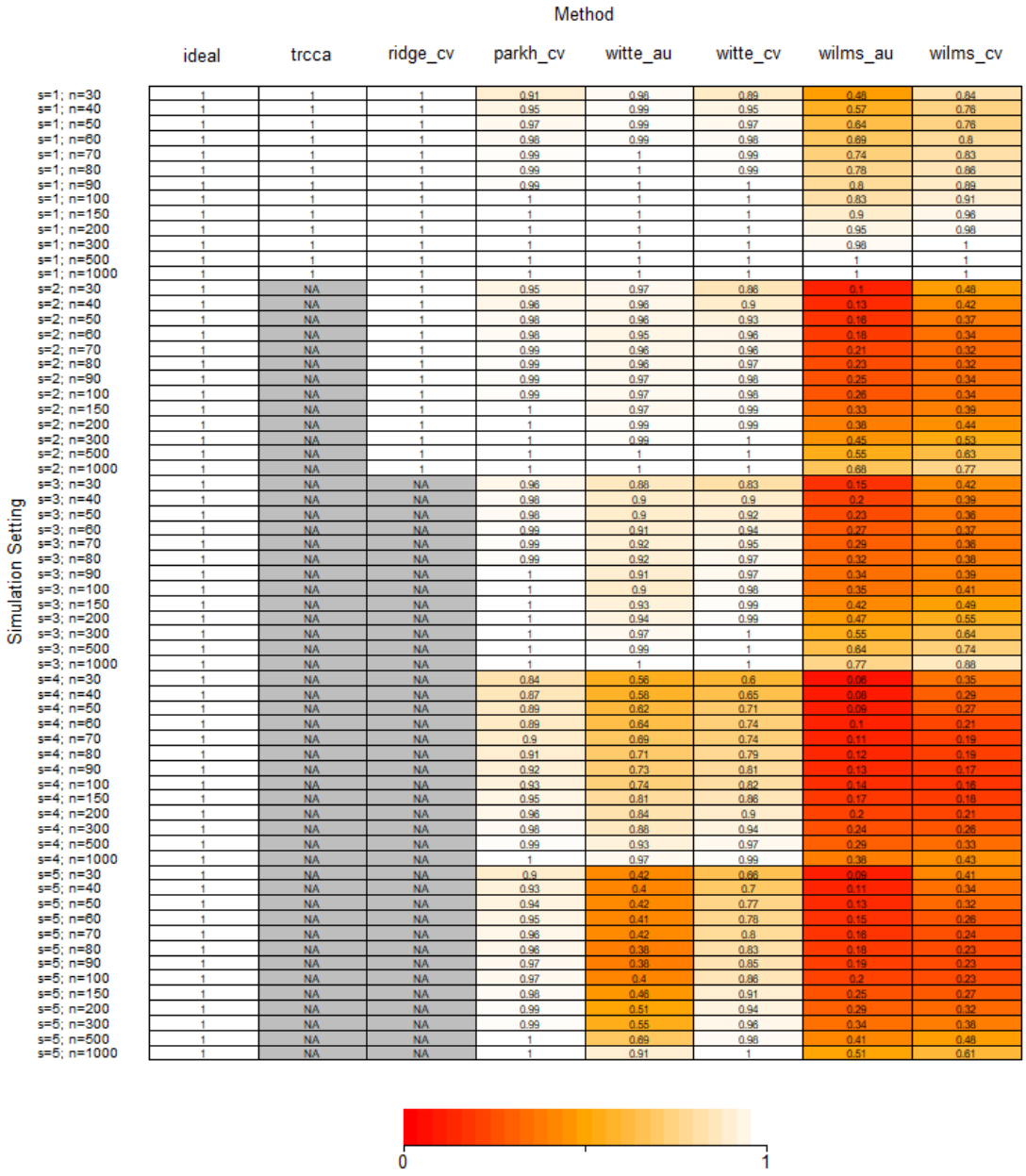


Figure 5a: A presentation of simulation results: the true positive rate (TPR) of the first canonical vector for $\mathbb{X}_2$ (in the cells), across simulation scenarios (rows) and CCA methods (columns). Column 1 contains the ideal value of TPR, which is 1. All other cells contain a mean TPR across R=1000 simulation runs for the scenario and CCA method corresponding to the cells' location. The color scale is used to assist reading the table, with white being the ideal case where TPR is 1, and cells closer to red corresponding to reduced TPR.

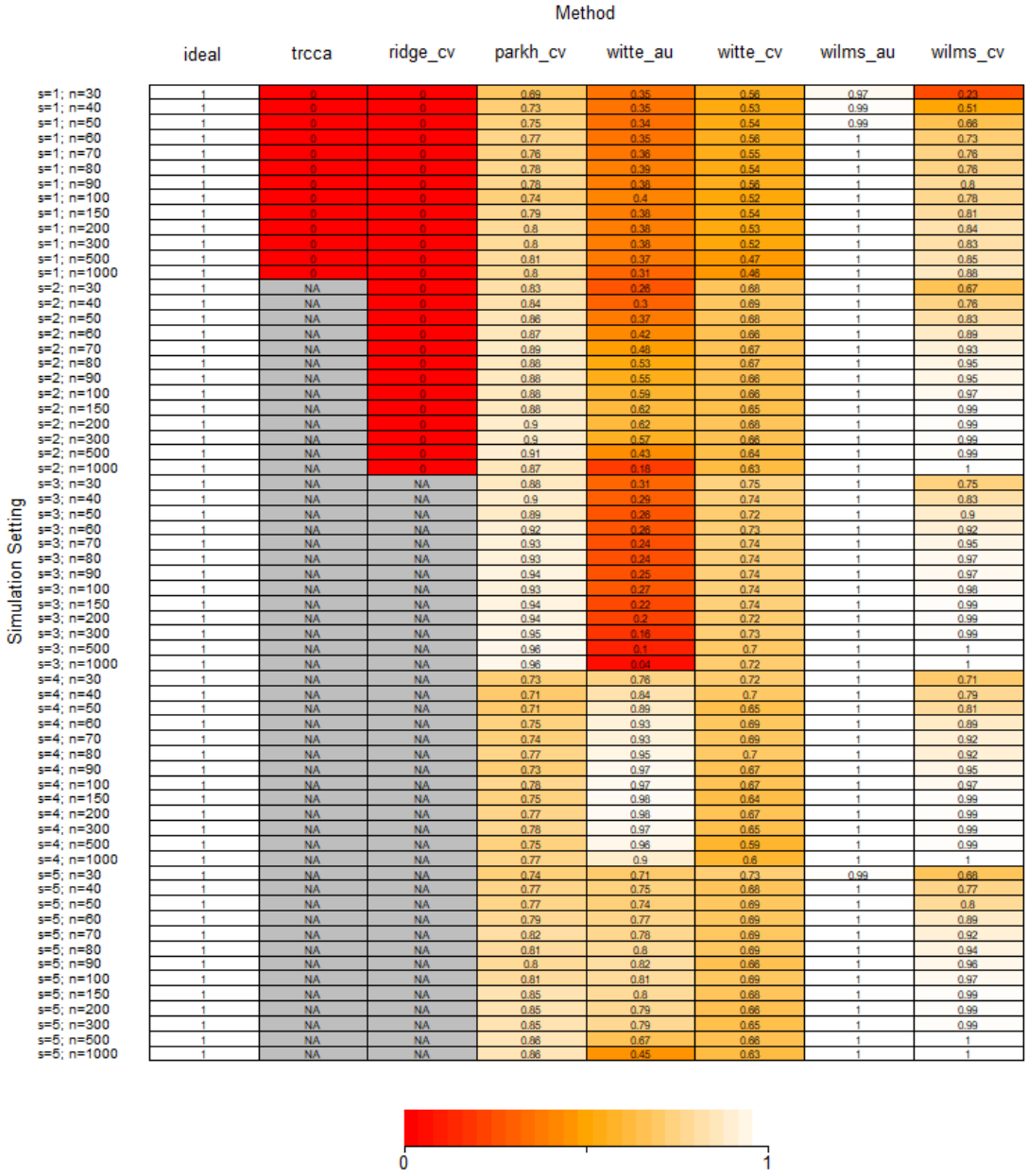


Figure 5b: A presentation of simulation results: the true negative rate (TNR) of the first canonical vector for $\mathbb{X}_2$ (in the cells), across simulation scenarios (rows) and CCA methods (columns). Column 1 contains the ideal value of TNR, which is 1. All other cells contain a mean TNR across R=1000 simulation runs for the scenario and CCA method corresponding to the cells' location. The color scale is used to assist reading the table, with white being the ideal case where TNR is 1, and cells closer to red corresponding to reduced TNR.

Simulation Setting	Method							
	ideal	trcca	ridge_cv	parkh_cv	witte_au	witte_cv	wilms_au	wilms_cv
s=1; n=30	10	20	20	12.2	16.3	13.2	5	16.1
s=1; n=40	10	20	20	12.2	16.4	14.2	5.9	12.5
s=1; n=50	10	20	20	12.1	16.5	14.3	6.4	10.9
s=1; n=60	10	20	20	12.2	16.4	14.2	7	10.7
s=1; n=70	10	20	20	12.2	16.4	14.4	7.4	10.7
s=1; n=80	10	20	20	12	16.1	14.6	7.8	11
s=1; n=90	10	20	20	12.1	16.1	14.4	8.1	10.9
s=1; n=100	10	20	20	12.5	16	14.7	8.3	11.3
s=1; n=150	10	20	20	12.1	16.2	14.6	9	11.5
s=1; n=200	10	20	20	12	16.2	14.7	9.5	11.4
s=1; n=300	10	20	20	12	16.2	14.8	9.8	11.6
s=1; n=500	10	20	20	11.9	16.3	15.3	10	11.5
s=1; n=1000	10	20	20	12	16.9	15.4	10	11.2
s=2; n=30	100	NA	1000	247.8	759.8	375.6	10.2	347.6
s=2; n=40	100	NA	1000	241.4	722.3	367.8	13	256
s=2; n=50	100	NA	1000	224.7	659.2	384.6	15.8	192
s=2; n=60	100	NA	1000	214.5	614.5	397.8	18.5	131.4
s=2; n=70	100	NA	1000	199.6	559.9	394	20.8	95.6
s=2; n=80	100	NA	1000	207.5	521.1	397.3	22.7	79.8
s=2; n=90	100	NA	1000	209.7	505.3	399.3	24.8	77.7
s=2; n=100	100	NA	1000	208.4	461.2	401.6	26.5	61.1
s=2; n=150	100	NA	1000	204.2	439.8	410.2	33.2	50.4
s=2; n=200	100	NA	1000	187.8	438.1	386.8	38.2	53.1
s=2; n=300	100	NA	1000	193.4	482.9	402.1	45.4	60.4
s=2; n=500	100	NA	1000	179.8	610.6	423.4	55	68.2
s=2; n=1000	100	NA	1000	214.1	837.3	437.4	68	81
s=3; n=30	50	NA	NA	163.1	702.4	282.3	9	257.4
s=3; n=40	50	NA	NA	148.3	721.4	287.3	10.6	183.8
s=3; n=50	50	NA	NA	154.4	748.1	308.9	12.1	115.6
s=3; n=60	50	NA	NA	124.7	743.9	303.2	13.7	96.9
s=3; n=70	50	NA	NA	118.9	765.5	294.1	14.7	67.8
s=3; n=80	50	NA	NA	113.2	768.5	293.7	15.9	52.2
s=3; n=90	50	NA	NA	107.1	762.2	296.9	16.9	47.2
s=3; n=100	50	NA	NA	113.4	735.1	291.9	17.7	39.8
s=3; n=150	50	NA	NA	105.8	783.7	292.6	21.1	33.8
s=3; n=200	50	NA	NA	103.2	808.3	317.7	23.6	33.5
s=3; n=300	50	NA	NA	93.5	845.5	309.2	27.3	38
s=3; n=500	50	NA	NA	91.6	908.5	337.5	32.1	41.8
s=3; n=1000	50	NA	NA	90.6	966	318	38.6	47.4
s=4; n=30	100	NA	NA	330.1	276.5	315.9	9.6	293.3
s=4; n=40	100	NA	NA	343.9	202.7	338	10.4	216.6
s=4; n=50	100	NA	NA	353.3	160.9	387.6	11.4	193.8
s=4; n=60	100	NA	NA	317.3	130.3	357.3	12.3	120.2
s=4; n=70	100	NA	NA	327.4	130.2	357.3	13.2	94.4
s=4; n=80	100	NA	NA	295.8	112	346.7	14	86.7
s=4; n=90	100	NA	NA	334.2	104.1	378.8	14.5	57.8
s=4; n=100	100	NA	NA	291.7	104.5	379.6	15	47.2
s=4; n=150	100	NA	NA	319.7	96.2	414.5	17.6	30.4
s=4; n=200	100	NA	NA	302.5	102.1	389.6	19.9	28.7
s=4; n=300	100	NA	NA	295.1	110.7	409.4	23.9	31.6
s=4; n=500	100	NA	NA	322	128	468.7	29.4	37.9
s=4; n=1000	100	NA	NA	305	187.5	462.8	38.2	46.9
s=5; n=30	50	NA	NA	291	294.7	291.9	9.5	327.9
s=5; n=40	50	NA	NA	261.4	262.2	338.5	9.6	232.4
s=5; n=50	50	NA	NA	265	267.4	333.1	10	206.1
s=5; n=60	50	NA	NA	244.3	243.4	334.5	10.1	117
s=5; n=70	50	NA	NA	215.1	232.3	334.4	10.5	87.8
s=5; n=80	50	NA	NA	233.2	205.1	338.4	10.8	64.5
s=5; n=90	50	NA	NA	237.2	191.1	365.7	11	48.5
s=5; n=100	50	NA	NA	226.3	202.9	335.3	11.2	40
s=5; n=150	50	NA	NA	194.6	215.5	354.1	12.8	24.3
s=5; n=200	50	NA	NA	187.9	227	369.2	14.5	24.4
s=5; n=300	50	NA	NA	190.6	224.5	382	16.9	24.9
s=5; n=500	50	NA	NA	180.9	347	367.2	20.3	26.6
s=5; n=1000	50	NA	NA	182	570.4	403.9	25.7	34.7

Figure 5c: A presentation of simulation results: the number of non-zero values (NNZ), or overall sparsity, in the first canonical vector for $\mathbb{X}_2$ (in the cells), across simulation scenarios (rows) and CCA methods (columns). Column 1 contains the ideal value of NNZ, which is the designed group size corresponding to the simulation scenario; see Table 1. All other cells contain a mean NNZ across R=1000 simulation runs for the scenario and CCA method corresponding to the cells' location. No color has been used here.

3.4. Real data analysis

In this section, we describe our analysis for the real TGP data and present results. We conduct two analysis strategies involving sparse CCA to tackle different research

questions regarding the relationship between toxicity and gene expression. The first one involves applying sparse CCA to the full data, and the second involves applying sparse CCA to data split by the FDA drug label DILI concern; once for "Most" and once for "Less or no" DILI concern. For each analysis strategy, guided by findings from our simulation experiments, we use **parkh.cv** as the primary sparse CCA method, using **wilms.au** as secondary method to highlight the most likely variables to be truly cross-correlated.

Analysis strategy 1

Our first sparse CCA strategy aims to estimate the overall association between conventional toxicity variables and gene expression, regardless of type of drug. We run sparse CCA analyses on the full dataset involving all $n = 226$ samples, and observe strength of canonical correlation as well as which variables are cross-correlated.

From the 40 pathology variables and 1000 gene expression variables, the **parkh.cv** method retained 27 pathology variables and 27 genes, with estimated canonical correlation 0.821. The **wilms.au** method, which we learned from our simulation experiments to be over sparse in a number of data scenarios, estimated the maximal correlation to exist between just 1 pathology variable, "platelet count", and 1 gene, "A2m: alpha-2-macroblobulin", with estimated canonical correlation 0.768. These two variables had the highest estimated loading values in the canonical vectors from **parkh.cv**. Figure 6 depicts the sparsely estimated cross-correlation between the pathology variables and genes. Table A1 in the Appendix shows exact loading values for the first pair of estimated canonical vectors from applying the **parkh.cv** sparse CCA method on the TGP data.

*** Figure 6 APPROXIMATELY HERE ***

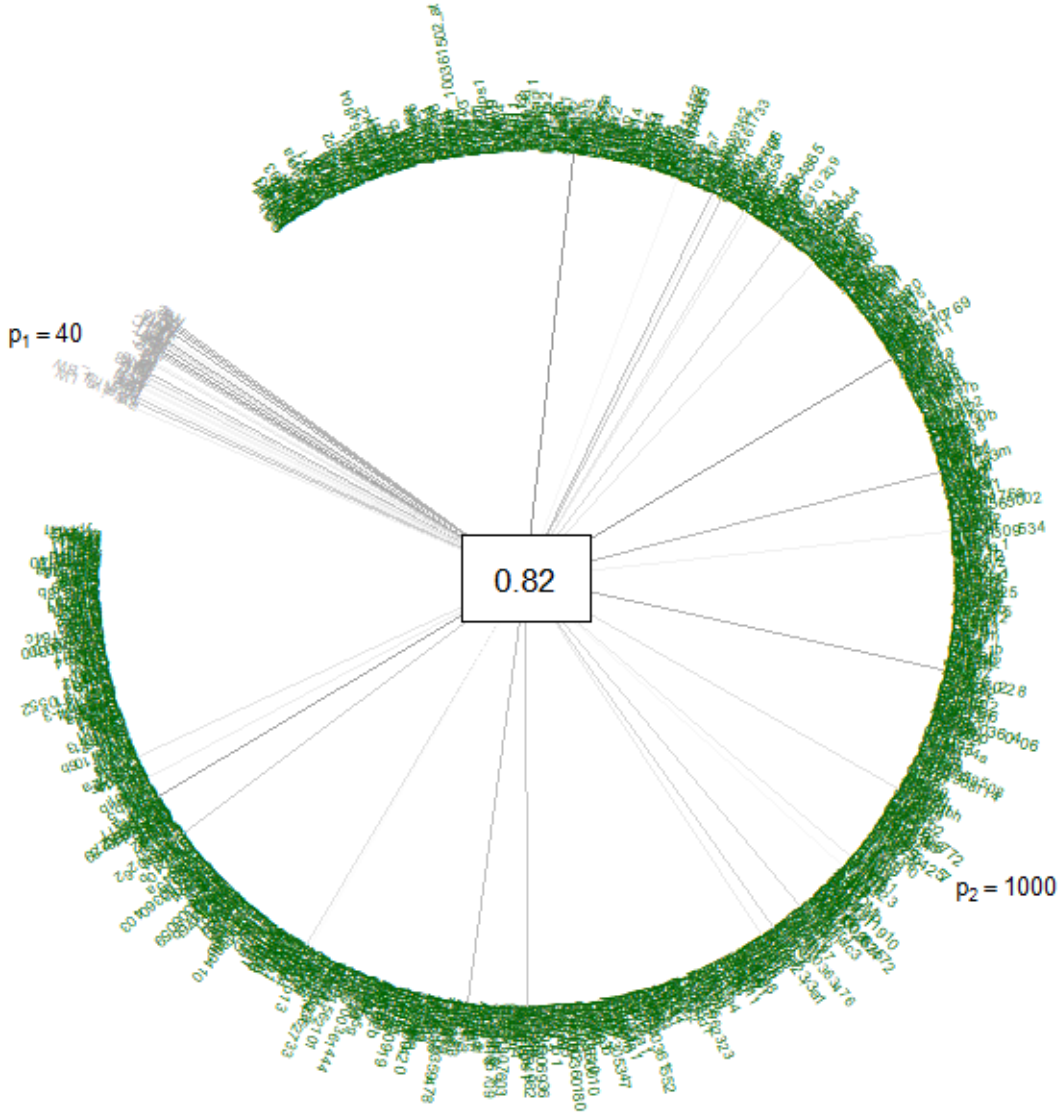


Figure 6: A visualization of the first pair of canonical vectors estimated by the ***parkh.cv*** *sparse CCA method. Variable names are listed around the circumference of the circular plot; grey variable names are from the pathology data and green variable names are genes. The estimated canonical correlation is presented in the middle. Lines have been drawn from the center of the plot to those variables which were estimated to have non-zero contribution in the canonical variates. The loading values are represented by the transparency of the lines; darkest when* $|w_{.j}| = 1$ *and invisible when* $|w_{.j}| = 0$.

Analysis strategy

Our second sparse CCA strategy aims to estimate toxicity-gene associations found for drugs of "Most DILI concern" and for drugs of "Less or no DILI concern", and find commonalities and differences between the results. We accomplish this by running two sparse CCA analyses; one for samples that had received a "Most DILI concern" drug and one for samples that had received a "Less or no DILI concern" drug. We then observe which conventional toxicity variables ($\mathbb{X}_1$) and gene expression variables ($\mathbb{X}_2$) are estimated to be cross-correlated (loading non-zero) for both levels of DILI concern, as well as those variables that are estimated to be cross-correlated in only one level of DILI concern. The most interesting variables will be those that are cross-correlated for "Most DILI concern" samples but not for "Less or no DILI concern" samples, since they will potentially strongly discriminate drugs that will unexpectedly harm humans from those that will perform safely.

The sparse CCA methods returned different results across levels of DILI concern. The **parkh.cv** method estimated a canonical correlation of 0.790 between 18 toxicity variables and all 1000 genes for samples subjected to drugs of most DILI concern and estimated a canonical correlation of 0.858 between all 40 toxicity variables and 41 genes for samples subjected to drugs of less or no DILI concern. The **wilms.au** method estimated a canonical correlation of 0.641 for just 1 pathology variable, "aspartate aminotransferase", and 1 gene, "Cyp17a1: cytochrome P450, family 17, subfamily a, polypeptide 1", for samples subjected to drugs of most DILI concern and estimated a canonical correlation of 0.799 between just 1 toxicity variable, "platelet count", and 1 gene, "A2m: alpha-2-macroblobulin", for samples subjected to drugs of less or no DILI concern. Figure 7 depicts the estimated cross-correlation between the pathology variables and genes for both subsets. Tables A2 and A3 in the Appendix show the estimated loadings from the **parkh.cv** method for most DILI concern and less to no DILI concern, respectively.

*** Figure 7 APPROXIMATELY HERE ***

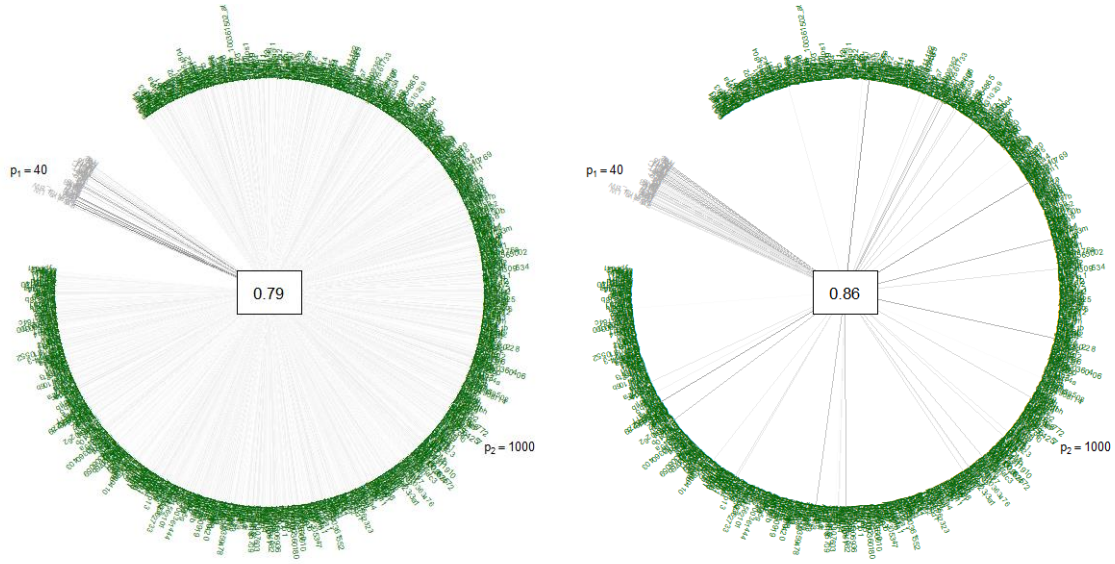


*Figure 7: A visualization of the first pair of canonical vectors estimated by the **parkh.cv** sparse CCA method for most DILI concern samples (left) and less or no DILI concern samples (right). Variable names are listed around the circumference of the circular plots; grey variable names are from the pathology data and green variable names are genes. The estimated canonical correlation is presented in the middle. Lines have been drawn from the center of the plot to those variables which were estimated to have non-zero contribution in the canonical variates. The loading values are represented by the transparency of the lines; darkest when* $|w_{.j}| = 1$ *and invisible when* $|w_{.j}| = 0$.

3.5. Discussion:

In this paper, we conducted extensive simulations to evaluate and compare performances of various sparse CCA methods, with a focus on high dimensional data. The results from our simulations demonstrate that both the methodology used to solve the sparse CCA optimization problem and the criteria used to select tuning parameters can greatly affect the performance of a sparse CCA method. The performance of the methods we compared also differed with respect to strength of correlation and sample size, while group size was a less influential factor. Our findings indicate that the **parkh.cv** method performs well relative to **witte** and **wilms** methods, and the aggressively sparse **wilms.au** could be used to flag variables that are highly likely to be cross-correlated.

Our study has several strengths. First, we modeled the data in our simulation experiments based on real data from a toxicogenomic study, and hence making the scenarios relevant in practical applications. This, therefore, improves upon proof-of-concept simulations that are typically conducted alongside newly-developed methods.

For example, the newer methods by Wilms & Croux, 2015 used synthetic designs involving only a handful of cross-correlated variables that had no group structure. Using real data to inform our simulations allowed us to choose methods that performed best for the application at hand.

Second, by using and improving upon code by Wilms & Croux, we compared the relative performance of a variety of methods, including a standard approach to tuning parameter selection; cross-validation to maximize test-sample canonical correlation. This allowed a fair comparison of sparse CCA methods, but also highlighted the influence of tuning parameter selection approach. Third, where appropriate, we implemented and provided a two-stage grid search for tuning parameter selection, which can cut down run times by a large factor. Finally, we demonstrated use of multivariate methods in the field of toxicogenomics; a field in need of advanced techniques to maximize the return on investment for long-standing research projects. [17,18]

Several improvements could be made to our simulations. First, the sparse CCA methods we considered in this paper are not exhaustive. We included two of the original methods being that they are easily accessible through readily available code.[6,7] This is because we wanted to bring light to what is most likely to be used in practical applications. We included the newer approaches by Wilms & Croux, 2015, because the authors showed promising results in their simulations and provided a nice set of codes to build upon. Other methods and approaches to selecting tuning parameters could have been added. For instance, both Lin et al., 2013 [11,30] and Wang et al., 2014 [31] developed methods involving a group-LASSO penalty [32,33] to better select groups of correlated features from each domain together. This improvement could be beneficial to data similar to the toxicogenomic data we used in this paper, where genes are highly correlated. Second, our simulations are restricted to assessing the accuracy of the first canonical variates. Although most simulation work to date has adopted this restriction, in application it is valuable to look beyond the first canonical variates because there can obviously be second, third, and so forth, cross-correlated groups of variables. Computation power is a limiting factor, as well as the fact that orthogonality between subsequent canonical variates is usually forfeited with the onset of sparsity constraints. Finally, we could expand the range of simulation scenarios to incorporate a high-dimensional $\mathbb{X}_1$ alongside $\mathbb{X}_2$.

Dedicated simulation work involving sparse CCA has been conducted and presented by other researchers. Chalise et al., 2012, [34] compared a variety of penalty functions infused to the sparse CCA formulation by Parkhomenko et al., 2009. From the results in their study, they suggested the use of the smoothly clipped absolute deviation

(SCAD) penalty [35] with an additional Bayesian information criterion (BIC) filter. [36]. Considering we found the **parkh.cv** method to perform best in our simulations, we recommend exploring the adaptation suggested by Chalise et al., 2012. Grellman et al., 2015 [37] compared the performance of sparse CCA by Witten et al., 2009 [7] with other multivariate feature selection methods, including partial least squares (PLS) variants, for high-dimensional, multi-collinear data, with the motivating context of prediction in neuroimaging genetic studies.

Finally, we would like to highlight that sparse CCA methodology has been shown to vary widely in performance, and hence we suggest that simulation experiments, when computationally feasible, should preface every instance of application. The framework and codes designed in this paper may assist future researchers in executing this somewhat challenging task.

References:

1. Hotelling H. Relations between two sets of variates. *Biometrika*. 1936;28(3-4):321-377. doi:10.1093/biomet/28.3-4.321.

2. Waaijenborg S, Verselewel de Witt Hamer PC, Zwinderman AH. Quantifying the association between gene expressions and DNA-markers by penalized canonical correlation analysis. *Stat Appl Genet Mol Biol*. 2008;7(1):Article3. doi:10.2202/1544-6115.1329.

3. Wanichthanarak K, Fahrmann J, Grapov D. Genomic, Proteomic, and Metabolomic Data Integration Strategies. *Biomark Insights*. 2015:1. doi:10.4137/BMI.S29511.

4. Tibshirani R. Regression shrinkage and selection via the lasso. *J R Stat Soc Ser B*. 1996:267-288.

5. Tibshirani R. Regression shrinkage and selection via the lasso: A retrospective. *J R Stat Soc Ser B Stat Methodol*. 2011;73(3):273-282.

6. Parkhomenko E, Tritchler D, Beyene J. Sparse canonical correlation analysis with application to genomic data integration. *Stat Appl Genet Mol Biol*. 2009;8:Article 1. doi:10.2202/1544-6115.1406.

7. Witten DM, Tibshirani R, Hastie T. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*. 2009;10(3):515-534. doi:10.1093/biostatistics/kxp008.

8. Liu Y, Hong Z, Tan G, et al. NMR and LC/MS-based global metabolomics to identify serum biomarkers differentiating hepatocellular carcinoma from liver cirrhosis. *Int J Cancer*. 2013. doi:10.1002/ijc.28706.

9. Chi EC, Allen GI, Zhou H, Kohannim O, Lange K, Thompson PM. Imaging genetics via sparse canonical correlation analysis. *Proc IEEE Int Symp Biomed Imaging*. 2013;2013:740-743. doi:10.1109/ISBI.2013.6556581.

10. Shen X, Sun Q. A novel semi-supervised canonical correlation analysis and extensions for multi-view dimensionality reduction. *J Vis Commun Image Represent*. 2014;25(8):1894-1904. doi:10.1016/j.jvcir.2014.09.004.

11. Lin D, Calhoun VD, Wang Y-P. Correspondence between fMRI and SNP data by group sparse canonical correlation analysis. *Med Image Anal*. 2014;18(6):891-902. doi:10.1016/j.media.2013.10.010.

12. Avants BB, Libon DJ, Rascovsky K, et al. Sparse canonical correlation analysis relates network-level atrophy to multivariate cognitive measures in a

neurodegenerative population. *Neuroimage*. 2014;84:698-711. doi:10.1016/j.neuroimage.2013.09.048.

13. Lee G, Singanamalli A, Wang H, et al. Supervised Multi-View Canonical Correlation Analysis (sMVCCA): Integrating Histologic and Proteomic Features for Predicting Recurrent Prostate Cancer. *Med Imaging, IEEE Trans*. 2015;34(1):284-297. doi:10.1109/TMI.2014.2355175.

14. Rousu J, Agranoff DD, Sodeinde O, Shawe-Taylor J, Fernandez-Reyes D. Biomarker discovery by sparse canonical correlation analysis of complex clinical phenotypes of tuberculosis and malaria. *PLoS Comput Biol*. 2013;9(4):e1003018. doi:10.1371/journal.pcbi.1003018.

15. Chalise P, Batzler A, Abo R, Wang L, Fridley BL. Simultaneous analysis of multiple data types in pharmacogenomic studies using weighted sparse canonical correlation analysis. *OMICS*. 2012;16(7-8):363-373. doi:10.1089/omi.2011.0126.

16. Uehara T, Ono A, Maruyama T, et al. The Japanese toxicogenomics project: application of toxicogenomics. *Mol Nutr Food Res*. 2010;54(2):218-227. doi:10.1002/mnfr.200900169.

17. Igarashi Y, Nakatsu N, Yamashita T, et al. Open TG-GATEs: a large-scale toxicogenomics database. *Nucleic Acids Res*. 2015;43(Database issue):D921-7. doi:10.1093/nar/gku955.

18. Suter L, Schroeder S, Meyer K, et al. EU framework 6 project: predictive toxicology (PredTox)--overview and outcome. *Toxicol Appl Pharmacol*. 2011;252(2):73-84. doi:10.1016/j.taap.2010.10.008.

19. Ganter B, Snyder RD, Halbert DN, Lee MD. Toxicogenomics in drug discovery and development: mechanistic analysis of compound/class-dependent effects using the DrugMatrix database. *Pharmacogenomics*. 2006;7(7):1025-1044. doi:10.2217/14622416.7.7.1025.

20. Hartigan JA. Clustering Algorithms. *Inf Retr Data Struct Algorithms*. 1975;2:419-442. doi:10.2307/2529577.

21. Vinod HD. Canonical ridge and econometrics of joint production. *J Econom*. 1976;4(2):147-166.

22. Zou H, Hastie T. Regularization and variable selection via the elastic net. *J R Stat Soc Ser B-Statistical Methodol*. 2005;67:301-320. doi:10.1111/j.1467-9868.2005.00503.x.

23. Johnson RA, Wichern DW. *Applied Multivariate Statistical Analysis*. 6th ed.; 2007.

doi:10.1198/tech.2005.s319.

24. Brillinger DR. *Time Series: Data Analysis and Theory*.; 1981. doi:10.1016/0304-4149(79)90039-5.

25. Izenman AJ. Reduced-rank regression for the multivariate linear model. *J Multivar Anal*. 1975;5(2):248-264. doi:10.1016/0047-259X(75)90042-1.

26. Wold HOA. *Nonlinear Estimation by Iterative Least Square Procedures*. Wiley; 1968.

27. Branco JA, Croux C, Filzmoser P, Oliveira MR. Robust canonical correlations: A comparative study. *Comput Stat*. 2005;20(2):203-229. doi:10.1007/BF02789700.

28. Lee W, Lee D, Lee Y, Pawitan Y. Sparse Canonical Covariance Analysis for High-throughput Data. *Stat Appl Genet Mol Biol*. 2011;10(1):1-24.

29. Wilms I, Croux C. Sparse canonical correlation analysis from a predictive point of view. *Biom J*. 2015;57(5):834-851. doi:10.1002/bimj.201400226.

30. Lin DD, Zhang JG, Li JY, Calhoun VD, Deng HW, Wang YP. Group sparse canonical correlation analysis for genomic data integration. *BMC Bioinformatics*. 2013;14:16. doi:10.1186/1471-2105-14-245.

31. Wang H, Singanamalli A, Ginsburg S, Madabhushi A. Selecting features with group-sparse nonnegative supervised canonical correlation analysis: multimodal prostate cancer prognosis. *Med Image Comput Comput Assist Interv*. 2014;17(Pt 3):385-392. http://www.ncbi.nlm.nih.gov/pubmed/25320823. Accessed December 15, 2014.

32. Meier L, Van De Geer S, Bühlmann P. The group lasso for logistic regression. *J R Stat Soc Ser B (Statistical Methodol*. 2008;70(1):53-71.

33. Simon N, Friedman J, Hastie T, Tibshirani R. A sparse-group lasso. *J Comput Graph Stat*. 2013;22(2):231-245.

34. Chalise P, Fridley BL. Comparison of penalty functions for sparse canonical correlation analysis. *Comput Stat Data Anal*. 2012;56(2):245-254.

35. Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. *J Am Stat Assoc*. 2001. doi:10.1198/016214501753382273.

36. Zhou J, He X. Dimension reduction based on constrained canonical correlation and variable filtering. *Ann Stat*. 2008;36(4):1649-1668. doi:10.1214/07-AOS529.

37. Grellmann C, Bitzer S, Neumann J, et al. Comparison of variants of canonical

correlation analysis and partial least squares for combined analysis of MRI and genetic data. *Neuroimage*. 2015;107:289-310. doi:10.1016/j.neuroimage.2014.12.025.

Appendix:

Data description:

In this section, we provide a brief description of the toxicogenomics database motivating our methods, along with the subset of data we analyzed in this paper.

The Japanese Toxicogenomic Project (TGP) database

The TGP was a joint collaboration between the National Institute of Biomedical Innovation (NIBIO), the National Institute of Health Sciences (NIHS), and 18 pharmaceutical companies starting in 2002.[1–3] The TGP initiative produced a massive toxicogenomics database containing quantitative hematology, biochemistry, and gene expression measurements, alongside histopathology assessments from pathology images. Measurements were taken from liver or kidney samples at different time points after exposing rat in vivo, rat in vitro, and human in vitro experimental units to different doses of over 170 drugs. The database has been made publicly available (http://toxico.nibiohn.go.jp/english/) via the Open Toxicogenomics Project-Genomics Assisted Toxicity Evaluation Systems (TG-GATEs). For full details regarding the TGP, we refer the reader to.[2,3]

Subset of TGP data analyzed in this paper

We used data summarized and provided by the 2013 Conference on the Critical Assessment of Massive Data Analysis (CAMDA). The TGP data from CAMDA can be obtained here: http://dokuwiki.bioinf.jku.at/doku.php/contest_dataset. The CAMDA competition also linked a classification of human drug-induced liver injury (DILI) concern for drugs that had been assessed by the Food and Drug Administration (FDA).[4] These drugs had been on the market for over 10 years, and were attributed a class of "most", "less", or "no" DILI concern, based on FDA-approved drug labels. This classification offers critical insight into the toxicity of drugs in the human population. In the following paragraphs, we describe our specific choices regarding the samples and variables from the TGP database.

Samples: We considered only liver samples from rat in vivo repeated dose experiments. These samples had both conventional toxicity assessment data (hematology, biochemistry, liver weight) measurements and gene expression data; we excluded samples that were missing data from either one of these domains. Furthermore, appreciating that higher doses result in more robust measurements of gene expression,[5] we considered only those samples that received a middle or high dose

of drug. We also considered only those samples that had measurements taken at 15 day or 29 day time points, anticipating that longer exposure times would generate data more concordant with situations where humans would experience toxicity (i.e., repeated doses, prolonged exposure). Finally, we only considered samples that had received a drug with a FDA human DILI concern classification.

Variables: There were three data domains – conventional toxicology assessment variables, gene expression variables, and a single human drug-induced liver injury (DILI) concern variable.

Conventional toxicology assessment (pathology) variables: We considered all hematology assessment parameters (16 variables), all biochemical parameters (21 variables), and all liver weight measurements (3 variables) in tandem as one data domain containing 40 conventional toxicology assessment variables. All variables were continuous by nature and we centered and scaled each.

Gene expression variables: Gene expression data had been measured with Affymetrix GeneChip® RAE 230A 2.0 Array technology. We used the replicate-collapsed gene expression data that had undergone batch-correction and Factor Analysis for Robust Microarray Summarization (FARMS) [REF: Hochreiter2006], as performed by the CAMDA organizers. A total of 12088 gene expression variables were available, annotated with unique gene names. We kept only those gene expression variables that had an informative/non-informative call value less than 0.5, indicating the variable is likely to offer more signal than noise.[6] This unsupervised filtering technique is extended from the FARMS method and has been shown to be a successful dimension reduction tool.[6] Finally, we excluded variables with low variability across samples, keeping only the top 1000 variables based on their inter-quartile range.

Human DILI concern variable: We used a binary version of the three-class human DILI concern variable. The variable originally consisted of class labels: “most…”, “less…”, or “no human DILI concern”. However, we combined the class labels “less…” and “no human DILI concern” because for each analysis we planned to perform, there were seldom enough samples that received a drug labeled “no human DILI concern” to reliably incorporate and measure the impact of this distinction.

Final data considered for analysis

In summary, the final data we used for analysis contained $n = 226$ rat liver samples, each having been subjected, in vivo, to repeated high or moderate doses of a drug for either 15 days or 29 days. Each sample had measurements from $p_1 = 40$ conventional

toxicology assessment variables and $p_2 = 1000$ gene expression variables, as well as a human DILI concern classification for the drug it was subjected to.

Data visualization

Figures A1 shows an image (heatmap) of the pathology data. Variables are displayed across the columns, with the grey sidebar indicating pathology variables. Samples are displayed across the rows; the red sidebar spans samples that received a drug classified as "Most DILI concern" and the blue sidebar spans samples that received a drug classified as "Less or no DILI concern".

*** Figure A1 APPROXIMATELY HERE ***

Figure A1: An image of pathology data for samples from our primary dataset. The data has been scaled. The range of the data was [-6.07, 14.31] but the color scale range was set to [-14.31, 14.31] to ensure data coloring was centered at 0 (white).

Figure A2 shows an image of the gene expression data. The image is organized similarly, but with the green sidebar indicating the variables are gene expression and the color scale now covering the gene expression data range.

*** Figure A2 APPROXIMATELY HERE ***

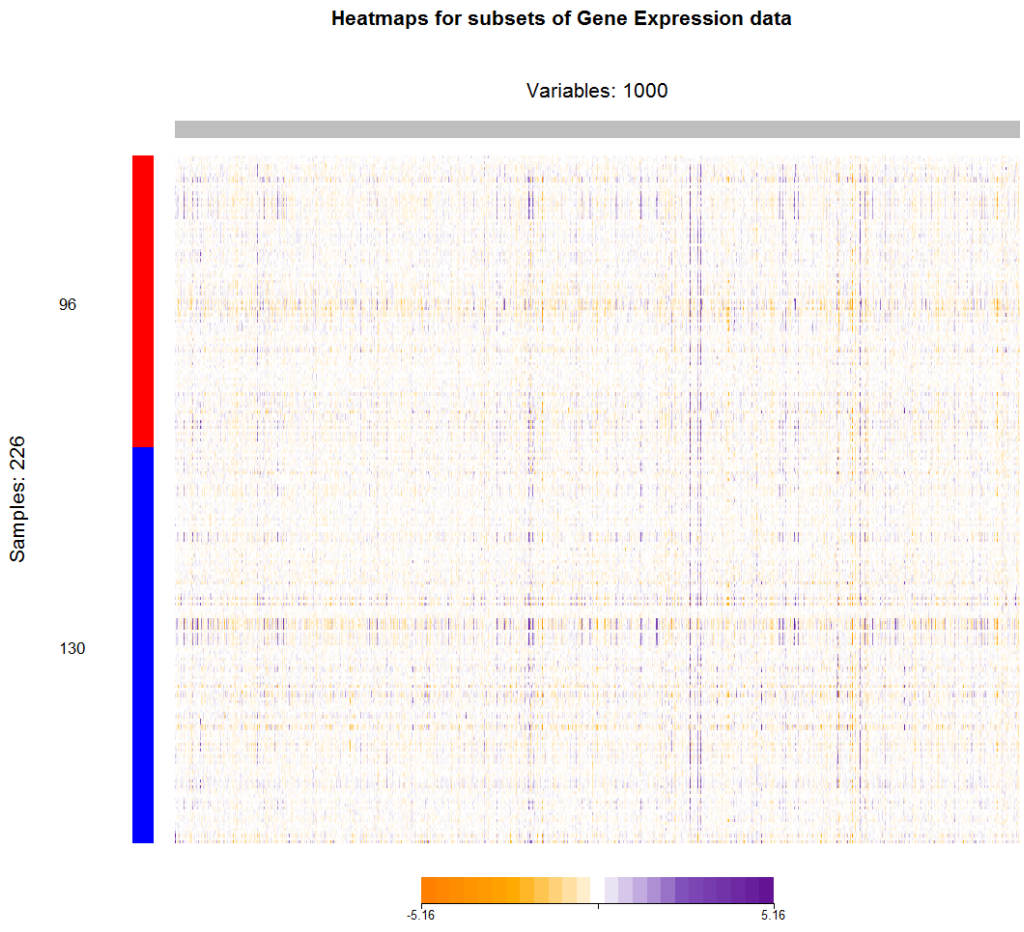


Figure A2: An image of the gene expression data for samples from our primary set of samples. The range of the data was [-5.16, 4.9] but the color scale range was set to [-5.16, 5.16] to ensure data coloring was centered at 0 (white).

Full Interpretation from Simulation Results:

Interpretation of estimation accuracy for canonical vector from $\mathbb{X}_1$ (Figures 4a, 4b, 4c):

As expected, with a TPR of 1 and TNR of 0, both non-sparse methods, **trcca** and **ridge.cv**, capture all truly cross-correlated variables but fail to exclude any variables that were truly not cross-correlated. The **parkh.cv** method was very successful, with a perfect or near-perfect TPR for all sample sizes under both the primary scenario ($s = 2$) and the smaller group scenario ($s = 3$). When strength of correlation was lowered ($s = 4,5$), it captured a minimum average of 87% (for $n = 30$) truly cross-correlated variables from $\mathbb{X}_1$ and perfect TPR with $n \geq 200$. The **parkh.cv** method also succeeded at eliminating variables that were unimportant to the cross-correlation in high-dimensional scenarios, with the smallest average TNR of 0.77 experienced when correlation was lowered ($s = 4$) and at $n = 30$; for higher correlation ($s = 2,3$) as sample size increased, **parkh.cv** reached almost perfect TPR and TNR. The **witte.cv** method performed similarly to the **parkh.cv** method, but had slightly less TPR at lower sample sizes and, most importantly, was not as sparse as **parkh.cv**; **witte.cv** had between 0.12 and 0.26 less TNR than **parkh.cv** depending on the simulation scenario, with larger deficits at higher sample sizes (it did not get better as sample size increased, whereas **parkh.cv** did).

The **wilms.cv** method also performed similarly to **parkh.cv** but with less TPR and, most importantly, a differing degree of TNR depending on the scenario. At lower sample sizes, **wilms.cv** had less TNR than **parkh.cv**, and lower than **witte.cv**, too. However, at larger samples sizes it approached the performance of **parkh.cv**, and even surpassed it in the presence of lower correlation ($s = 4$) when $n \geq 100$. The **wilms.au** method had either near-perfect or perfect 1 TNR across all scenarios, but was overly sparse at lower sample sizes, particularly in lower correlation scenarios ($s = 4,5$), with TPR as low as 0.31. However, at larger sample sizes, **wilms.au** obtains near-perfect to perfect 1 TPR; $n \geq 200$ for higher correlation scenarios ($s = 2,3$), $n > 500$ for lower correlation scenarios ($s = 4,5$). Given its' consistent specificity, **wilms.au** is likely the best performing method for large sample sizes.

The **witte.au** method generally underperformed compared to the other methods. At higher cross-correlations ($s = 2,3$) it lost some truly cross-correlated variables with TPR ranging from 0.71 to 0.99 at the highest sample size, while failing to exclude a large number of variables that were designed not to be truly cross-correlated; evident from TNR values as low as 0.1 depending on sample size. For lower cross-correlation ($s = 4,5$), similar to **wilms.au,** it became aggressively over-sparse. However,

contrary to what was experienced by **wilms.au**, **witte.au** did not enjoy the benefits of perfect TNR and did not improve as sample size increased.

Interpretation of estimation accuracy for canonical vector from $\mathbb{X}_2$ (Figures 5a, 5b, 5c):

The **parkh.cv** method again performs well, though with the larger number of variables experienced slightly reduced TPR and TNR. The **witte.cv** method was competitive with **parkh.cv**, but the larger number of variables increased its deficit in TPR and especially TNR (it was less sparse) relative to **parkh.cv**. The **wilms.cv** method was aggressively over-sparse, failing to capture as many as 84% truly cross-correlated variables from $\mathbb{X}_2$ (TPR of 0.16) across the simulation settings. Despite the over-sparse solutions, it only achieved perfect 1 TNR at the highest ($n \geq 500$) of sample sizes. Interestingly, the TPR was not monotone increasing with sample size and the **wilms.cv** method performed worst at moderate sample sizes (e.g., $n = 100$). The **wilms.au** method proved even more aggressive at reducing model complexity in the presence of higher-dimensional data. For example, in the primary simulation setting ($s = 2$), the **wilms.au** began by letting an average of only 10.2 variables in the canonical vector (TPR=0.1) with $n = 30$ and increased to an average of 68 variables (TPR = 0.68) with $n = 1000$. However, with rare exception, the method only ever included variables that were designed to be important (TNR of 1). The **witte.au** method performed similarly poorly as it did for estimating $\boldsymbol{w}_1$.

Tables from Real Data Analysis:

*Table A1: Variable names and loading values for the top canonical vector pair from the **parkh.cv** sparse CCA method used in Analysis 1.*

	Pathology Variable	loading	Gene Name	Gene Description	loading
1	platelet count*	0.368	A2m*	alpha-2-macroglobulin*	0.4083
2	lymphocyte	-0.3229	Lcn2	lipocalin 2	0.3931
3	hemoglobin	-0.3147	Lbp	lipopolysaccharide binding protein	0.3598
4	reticulocyte	0.3133	LOC360228	WDNM1 homolog	0.3006
5	hematocrit value	-0.3059	S100a9	S100 calcium binding protein A9	0.2792
6	neutrophil	0.2979	Alox15	arachidonate 15-lipoxygenase	0.2458
7	red blood cell count	-0.2625	Stac3	SH3 and cysteine rich domain 3	-0.2351
8	total protein	-0.2475	Serpina7	serpin peptidase	0.2106

				inhibitor, clade A (alpha-1 antiproteinase, antitrypsin), member 7	
9	fibrinogen	0.2434	Igfbp2	insulin-like growth factor binding protein 2	0.2004
10	terminal body weight	-0.2019	Cxcl1	chemokine (C-X-C motif) ligand 1 (melanoma growth stimulating activity, alpha)	0.1862
11	gamma-glutamyltranspeptidase	0.1929	Spink3	serine peptidase inhibitor, Kazal type 3	0.1599
12	white blood cell count	0.1802	Dhrs7	dehydrogenase/reductase (SDR family) member 7	-0.1456
13	albumin	-0.1513	Ddc	dopa decarboxylase (aromatic L-amino acid decarboxylase)	-0.1436
14	mean corpuscular hemoglobin concentration	-0.1486	Hamp	hepcidin antimicrobial peptide	-0.1103
15	glucose	-0.1063	Cyp3a9	cytochrome P450, family 3, subfamily a, polypeptide 9	-0.1068
16	calcium	-0.0944	Akr7a3	aldo-keto reductase family 7, member A3 (aflatoxin aldehyde reductase)	-0.1001
17	relative liver weight	0.0572	Car3	carbonic anhydrase 3	-0.098
18	total cholesterol	0.0546	Ces1f	carboxylesterase 1F	-0.0874
19	monocyte	0.0542	Mettl7b	methyltransferase like 7B	-0.0833
20	blood urea nitrogen	0.0422	Nupr1	nuclear protein, transcriptional regulator, 1	0.0778
21	prothrombin time	0.0372	Igfbp1	insulin-like growth factor binding protein 1	0.0686
22	lactate dehydrogenase	0.0355	Gpt	glutamic-pyruvate transaminase (alanine aminotransferase)	-0.0672
23	aspartate aminotransferase	0.035	C5	complement component 5	0.0531
24	direct bilirubin	0.0248	Oat	ornithine aminotransferase	-0.0333
25	albumin globulin ratio	-0.0156	Inmt	indolethylamine N-methyltransferase	-0.0323
26	mean corpuscular	-0.0076	Stat3	signal transducer and	0.0264

	hemoglobin			activator of transcription 3 (acute-phase response factor)	
27	phospholipid	0.0044	Insig2	insulin induced gene 2	-0.0061

** Variables involved in the canonical vectors estimated by **wilms.au***

*Table A2: Variable names and loading values for the top canonical vector pair from the **parkh.cv** sparse CCA method used in Analysis 2, for rat liver samples receiving drugs of most DILI concern. The gene list is truncated to match the length of the pathology variable list; in reality, all genes were estimated to have non-zero loadings.*

	Pathology Variable	loading	Gene Name	Gene Description	loading
1	terminal body weight	-0.4727	Pla2g12a	phospholipase A2, group XIIA	0.0819
2	lactate dehydrogenase	0.407	Ctsl1	cathepsin L1	0.0768
3	aspartate aminotransferase*	0.3812	Rbm3	RNA binding motif (RNP1, RRM) protein 3	0.0741
4	prothrombin time	0.3799	Pgcp	plasma glutamate carboxypeptidase	-0.072
5	gamma-glutamyltranspeptidase	0.2559	Asl	argininosuccinate lyase	0.0719
6	calcium	-0.2493	Pter	phosphotriesterase related	0.0717
7	triglyceride	-0.2292	Adipor2	adiponectin receptor 2	-0.0705
8	blood urea nitrogen	0.1874	Afm	afamin	-0.0705
9	liver weight	-0.1424	Enpp2	ectonucleotide pyrophosphatase/ phosphodiesterase 2	-0.0698
10	chlorine	0.1336	Pygl	phosphorylase, glycogen, liver	-0.0689
11	total protein	-0.1295	Trim5	tripartite motif-containing 5	-0.0681
12	lymphocyte	-0.1211	Gucy1b2	guanylate cyclase 1, soluble, beta 2	-0.0679
13	neutrophil	0.1182	Cyp17a1*	cytochrome P450, family 17, subfamily a, polypeptide 1*	0.0677
14	mean corpuscular hemoglobin	-0.0944	Gstm2	glutathione S-transferase mu 2	-0.0674
15	activated partial thromboplastin time	0.0838	Slc4a4	solute carrier family 4, sodium bicarbonate cotransporter, member 4	0.0674
16	mean corpuscular	-0.06	Rps5	ribosomal protein S5	0.0672

	volume				
17	eosinophil	-0.053	Abcb1a	ATP-binding cassette, sub-family B (MDR/TAP), member 1A	0.067
18	total cholesterol	0.0131	LOC100360406	rCG34104-like	0.0661

** Variables involved in the canonical vectors estimated by **wilms.au***

*Table A3: Variable names and loading values for the top canonical vector pair from the **parkh.cv** sparse CCA method used in Analysis 2, for rat liver samples receiving drugs of less or no DILI concern.*

	Pathology Variable	loading	Gene Name	Gene Description	loading
1	platelet count*	0.304	A2m*	alpha-2-macroglobulin*	0.3657
2	hemoglobin	-0.2908	Lcn2	lipocalin 2	0.3567
3	hematocrit value	-0.2898	Lbp	lipopolysaccharide binding protein	0.3266
4	lymphocyte	-0.2768	LOC360228	WDNM1 homolog	0.2929
5	red blood cell count	-0.2766	S100a9	S100 calcium binding protein A9	0.2606
6	reticulocyte	0.2765	Igfbp2	insulin-like growth factor binding protein 2	0.2521
7	neutrophil	0.2549	Cxcl1	chemokine (C-X-C motif) ligand 1 (melanoma growth stimulating activity, alpha)	0.2203
8	total protein	-0.2329	Serpina7	serpin peptidase inhibitor, clade A (alpha-1 antiproteinase, antitrypsin), member 7	0.2172
9	fibrinogen	0.226	Alox15	arachidonate 15-lipoxygenase	0.2166
10	gamma-glutamyltranspeptidase	0.1947	Stac3	SH3 and cysteine rich domain 3	-0.2038
11	terminal body weight	-0.1925	Spink3	serine peptidase inhibitor, Kazal type 3	0.1539
12	white blood cell count	0.1856	Ddc	dopa decarboxylase (aromatic L-amino acid decarboxylase)	-0.1478
13	albumin	-0.1694	Ces1f	carboxylesterase 1F	-0.1477
14	glucose	-0.1602	Dhrs7	dehydrogenase/reductase (SDR family) member 7	-0.1458
15	mean corpuscular	-0.154	Cyp3a9	cytochrome P450, family	-0.1389

	hemoglobin concentration			3, subfamily a, polypeptide 9	
16	calcium	-0.1294	Car3	carbonic anhydrase 3	-0.1149
17	monocyte	0.1098	Nupr1	nuclear protein, transcriptional regulator, 1	0.1088
18	blood urea nitrogen	0.1066	Hamp	hepcidin antimicrobial peptide	-0.1022
19	total cholesterol	0.0983	Igfbp1	insulin-like growth factor binding protein 1	0.1018
20	direct bilirubin	0.0931	C5	complement component 5	0.1003
21	alkaline phosphatase	-0.0911	Stat3	signal transducer and activator of transcription 3 (acute-phase response factor)	0.0995
22	lactate dehydrogenase	0.0868	Inmt	indolethylamine N-methyltransferase	-0.0925
23	phospholipid	0.083	Akr7a3	aldo-keto reductase family 7, member A3 (aflatoxin aldehyde reductase)	-0.0885
24	relative liver weight	0.0827	Oat	ornithine aminotransferase	-0.0877
25	chlorine	0.0803	Insig2	insulin induced gene 2	-0.0831
26	aspartate aminotransferase	0.0801	Mettl7b	methyltransferase like 7B	-0.0762
27	albumin globulin ratio	-0.0779	Hao2	hydroxyacid oxidase 2 (long chain)	-0.0683
28	prothrombin time	0.0771	Hmox1	heme oxygenase (decycling) 1	0.0675
29	mean corpuscular hemoglobin	-0.0767	Gpt	glutamic-pyruvate transaminase (alanine aminotransferase)	-0.0533
30	potassium	0.0766	Me1	malic enzyme 1, NADP(+)-dependent, cytosolic	-0.0441
31	triglyceride	0.0737	Bdh1	3-hydroxybutyrate dehydrogenase, type 1	-0.0431
32	mean corpuscular volume	0.0692	Cyp3a23/3a1	cytochrome P450, family 3, subfamily a, polypeptide 23/polypeptide 1	-0.0394
33	liver weight	-0.0676	Fgl1	fibrinogen-like 1	0.0335

34	inorganic phosphorus	0.0586	Tnfrsf9	tumor necrosis factor receptor superfamily, member 9	0.0312
35	sodium	-0.0583	Cd63	Cd63 molecule	0.0261
36	total bilirubin	0.0404	RGD1307603	similar to hypothetical protein MGC37914	-0.0112
37	alanine aminotransferase	0.0359	Mgll	monoglyceride lipase	-0.0075
38	activated partial thromboplastin time	0.0338	Igfals	insulin-like growth factor binding protein, acid labile subunit	-0.0052
39	creatinine	-0.0338	Ces1d	carboxylesterase 1D	-0.0049
40	eosinophil	-0.0087	Btg2	BTG family, member 2	0.004
41			Rbm3	RNA binding motif (RNP1, RRM) protein 3	0.0014

** Variables involved in the canonical vectors estimated by **wilms.au***

Appendix References:

1. Kondo C, Minowa Y, Uehara T, et al. Identification of genomic biomarkers for concurrent diagnosis of drug-induced renal tubular injury using a large-scale toxicogenomics database. *Toxicology*. 2009;265(1-2):15-26. doi:10.1016/j.tox.2009.09.003.

2. Uehara T, Ono A, Maruyama T, et al. The Japanese toxicogenomics project: application of toxicogenomics. *Mol Nutr Food Res*. 2010;54(2):218-227. doi:10.1002/mnfr.200900169.

3. Igarashi Y, Nakatsu N, Yamashita T, et al. Open TG-GATEs: a large-scale toxicogenomics database. *Nucleic Acids Res*. 2015;43(Database issue):D921-7. doi:10.1093/nar/gku955.

4. Chen M, Vijay V, Shi Q, Liu Z, Fang H, Tong W. FDA-approved drug labeling for the study of drug-induced liver injury. *Drug Discov Today*. 2011;16(15-16):697-703. doi:10.1016/j.drudis.2011.05.007.

5. McMillian M, Nie A, Parker JB, et al. Drug induced oxidative stress in rat liver from a toxicogenomics perspective. *Toxicol Appl Pharmacol*. 2005;207(2 Suppl):171-178.

6. Talloen W, Clevert D-A, Hochreiter S, et al. I/NI-calls for the exclusion of non-informative genes: a highly effective filtering tool for microarray data. *Bioinformatics*. 2007;23(21):2897-2902. doi:10.1093/bioinformatics/btm478.

Chapter 4

4. EVALUATION OF BOOTSTRAPING FOR SPARSE CCA

Context

My third project is in the form of a manuscript that will be submitted to a peer-reviewed journal. The work presented in this manuscript includes a rigorous simulation study that evaluates the performance of the non-parametric bootstrap approach at generating inferential measures for sparse CCA. The work was motivated by my understanding of the current state of the literature for sparse CCA and multivariate methods in general.

While reading methods and application papers, I noticed that reporting results from sparse CCA tended to consist of estimated canonical correlations and loading vectors, but no standard errors or confidence intervals. As well, there are limited tests of hypotheses available to judge the statistical significance of results from sparse CCA, as well as other multivariate methods.[24,74] As I outline in my manuscript, only a few papers have included such measures of variation and tests.[74–77] I decided to explore what inferential measures were available and potential strategies to bring more objective measures to the otherwise exploratory sparse CCA tools.

I found a couple instances where authors used bootstrapping to create standard errors and confidence intervals for the canonical correlations and loading vectors.[75,77] However, they did not investigate the reliability of the bootstrapped measures, which is suspect in the presence of $n < p$ data. I set forth to design and conduct simulation experiments to test the performance of the bootstrap approach to generate such measures for sparse CCA estimates.

I have prepared a manuscript including my work and plan to submit to a peer-reviewed journal. Starting on the next page, I include an MS Word formatted version of our manuscript. Please note that mathematical notation in this manuscript has been simplified to reflect use of only the first canonical components of sparse CCA. Notation is fully described within the contents of the manuscript.

MANUSCRIPT BEGINS ON THE NEXT PAGE…

Bootstrapping provides useful inferential measures for sparse canonical correlation analysis under certain high-dimensional data conditions: a simulation study

Ashley Bonner, BSc, MSc
Department of Health Research Methods, Evidence, and Impact; McMaster University

Jemila Hamid, BSc, MSc, PhD
Children's Hospital of Eastern Ontario, Ottawa, Ontario, Canada

Angelo Canty, BSc, MSc, PhD
Department of Mathematics & Statistics; McMaster University

Joseph Beyene*, BSc, MSc, PhD
Department of Health Research Methods, Evidence, and Impact; McMaster University

*
Corresponding Author:
Joseph Beyene
1280 Main Street W, Hamilton, ON, L8S 4L8, Canada
+1 9055259140 x21333
beyene@mcmaster.ca

Abstract:

Background: Sparse canonical correlation analysis (sparse CCA) excels at exploring complex relationships between multiple data domains, especially in situations involving high-dimensional data. However, certain limitations and methodological gaps do exist, where researchers face challenges in providing inferential measures related to strength of correlation between domains and in determining which variables are truly cross-correlated.

Methods: We considered the non-parametric bootstrap method and performed extensive simulations to investigate the performance of inferential measures for sparse CCA. Coverage probabilities of the bootstrapped confidence intervals for the canonical correlation coefficients were assessed for a range of high-dimensional data scenarios, varying sample sizes, number of variables, and strengths of correlations. Bootstrapped probabilities of each variable's inclusion were calculated to aid in identifying cross-correlated variables.

Results: The performance of bootstrapped measures ranged, based primarily on strength of canonical correlation and sample size relative to number of variables. At moderate to high canonical correlation, and as sample size approaches the number of variables, the bootstrapped confidence intervals of the coefficients approached nominal coverage. However, coverage was severely lacking for small sample sizes and moderate to weak correlation. Variables determined to have the high probability of being cross-correlated, as estimated by the bootstrap, had higher true-positive rate compared to those conventionally estimated to be cross-correlated with the sparse CCA method.

Conclusion: Bootstrapping allows inferential statements regarding sparse CCA to be made, but reliability of bootstrapped measures can be suspect when derived using high-dimensional data. Investigators using sparse CCA on data domains should consider adding bootstrapped inferential measures to strengthen their conclusions.

4.1. Introduction

Detecting and characterizing complex relationships among massive sets of data is becoming a common goal for researchers. Recent papers in neuroimaging and cancer research tackled this challenging task of 'cross-correlation' using variants of the classic multivariate analysis tool canonical correlation analysis (CCA).[1–4]

Given n observations on p_1 variables represented by $\mathbb{X}_1$ and p_2 variables represented by $\mathbb{X}_2$, CCA finds a linear combination of variables from $\mathbb{X}_1$ and a linear combination of variables from $\mathbb{X}_2$ that are maximally correlated.[5] The resulting linear combinations $\mathbb{X}_1\boldsymbol{w}_1$ and $\mathbb{X}_2\boldsymbol{w}_2$ are called *canonical variables* and the correlation between the two linear combinations is referred to as *canonical correlation, denoted by* ρ_{12}. The coefficient vectors $\boldsymbol{w}_1 = (w_{11}, w_{12}, \dots, w_{1p_1})'$ and $\boldsymbol{w}_2 = (w_{21}, w_{22}, \dots, w_{2p_2})'$ can be used to determine which variables contribute to the multivariate cross-correlation between $\mathbb{X}_1$ and $\mathbb{X}_2$, providing insight to more complex associations existing between data domains.

Conventional CCA, however, is not well-suited for studying complex relationships between data domains that are high-dimensional in size (i.e., the 'small n, large p' scenario), such as those easily captured by advanced technologies today.[6] When $n < \min(p_1, p_2)$ or when high collinearity exists between variables in either domain, solutions to CCA do not exist due to matrices involved in the estimation becoming ill-conditioned.[7,8] Even when solutions to CCA exist, it is almost always the case that all coefficients within the vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are estimated to be non-zero. With large p_1 and p_2, this leads to the implausible and impractical interpretation that all variables from $\mathbb{X}_1$ are correlated with all variables from $\mathbb{X}_2$.

Over the past decade, several groups of researchers have added special regularization constraints to the CCA objective function [7,9–15], both enabling its application when $n < \min(p_1, p_2)$ and estimating *sparse* coefficient vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$, which have a more plausible and practical interpretation; that only a subset of variables from $\mathbb{X}_1$ are correlated with a subset of variables from $\mathbb{X}_2$. This class of so-called *sparse* CCA methods allows us to easily explore multivariate connections between massive data domains and have the potential to uncover and characterize newfound relationships. In the case of genetics, for example, these methods can elucidate cross-omic mechanisms behind complex diseases and disorders; the challenging goal of many health research disciplines.

One challenge with any CCA method, and multivariate statistical methods in general, is drawing statistical inference.[16] Without inferential approaches for CCA, investigators are limited when drawing conclusions regarding the strength of correlation

between $\mathbb{X}_1$ and $\mathbb{X}_2$ or in determining which variables are involved in the relationship. Theoretical sampling distributions for estimators and test statistics arising from CCA are either questionable in reality or all-together non-existent.[17] This opens the door for resampling strategies to become the main form of statistical inference in CCA.

Some resampling approaches have been applied or tested for the conventional CCA. For instance, permutation tests have been used to obtain empirical p-values for canonical correlation estimates[18], but merely help to conclude whether or not the canonical correlation is likely to be greater than zero. In 1996, Fan & Wang showed, on a small scale ($p_1 = 3$, $p_2 = 3$), that the non-parametric bootstrap performed better than the jackknife at obtaining reasonable standard errors for canonical vector loadings.[17] In 2014, Sakar demonstrated that an ensemble CCA approach based on resampling and estimation aggregation can return canonical variables with increased canonical correlation in unseen test data, as compared to canonical variables estimated from a single instance of training data alone.[19] Their resampling strategies included the bootstrap, jackknife, and data partitioning, with the bootstrap having notably better performance in a number of examples.

A few papers have used the bootstrap for regularized or sparse CCA to accompany their results with standard errors or confidence intervals.[18,20,21] As well, one of the original sparse CCA papers [6] used it briefly to report the stability of canonical variates and appearance of certain variables within their experiments. Szefer et al., 2017 bootstrapped sparse CCA in an attempt to improve variable selection via selecting variables that were most frequently estimated across bootstrap samples.[22] However, despite dealing with high-dimensional (i.e., $n < p$) data, these papers did not investigate the reliability of the bootstrap strategy for sparse CCA. If resampling is to play a role in improving inference for sparse CCA, its performance must be evaluated using extensive simulations.

In this paper, we examine the performance of the bootstrap approach in facilitating inference for sparse CCA, with a focus on high-dimensional scenarios. In particular, we examine the reliability of bootstrap confidence intervals for canonical correlation coefficients as well as bootstrap probability estimates for variables to be included in the cross-correlation. In Section 2, we describe the methodology of conventional CCA, sparse CCA, and the general bootstrap, followed by our specific definitions for the bootstrap procedure for sparse CCA. In Section 3, we describe the simulation strategy we used to evaluate the bootstrap methods and present the simulation results. In Section 4, we discuss key findings from our simulation and potential future directions regarding the bootstrap method for sparse CCA.

4.2. Methods

4.2.1. Canonical Correlation Analysis (CCA)

Consider observing n samples on two sets of data; let $\mathbb{X}_1$ be a $n \times p_1$ matrix of data and $\mathbb{X}_2$ be a $n \times p_2$ matrix of data. Suppose we are interested in finding a linear combination $\mathbb{X}_1\boldsymbol{w}_1$, and a linear combination $\mathbb{X}_2\boldsymbol{w}_2$, that are highly correlated. CCA seeks to find a pair of coefficient vectors, $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$, that return linear combinations with the highest correlation, $\rho_{12} = Corr(\mathbb{X}_1\boldsymbol{w}_1, \mathbb{X}_2\boldsymbol{w}_2)$.[16] Since the choice of $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ is invariant to scaling, constraints $\boldsymbol{w}_1 \in \{\|\boldsymbol{w}_1\|_2^2 = 1\}$ and $\boldsymbol{w}_2 \in \{\|\boldsymbol{w}_2\|_2^2 = 1\}$ are included when formulating the objective of CCA:

$$\underset{\boldsymbol{w}_1, \boldsymbol{w}_2}{\text{maximize}} \left\{ \frac{\boldsymbol{w}_1 \widehat{\boldsymbol{\Sigma}}_{12} \boldsymbol{w}_2}{\sqrt{\boldsymbol{w}_1 \widehat{\boldsymbol{\Sigma}}_{11} \boldsymbol{w}_1} \sqrt{\boldsymbol{w}_2 \widehat{\boldsymbol{\Sigma}}_{22} \boldsymbol{w}_2}} \right\}$$
$$subject\ to\ \|\boldsymbol{w}_1\|_2^2 = 1, \|\boldsymbol{w}_2\|_2^2 = 1,$$

where $\widehat{\boldsymbol{\Sigma}}_{11} = \widehat{Var}(\mathbb{X}_1)$, $\widehat{\boldsymbol{\Sigma}}_{22} = \widehat{Var}(\mathbb{X}_2)$, and $\widehat{\boldsymbol{\Sigma}}_{12} = \widehat{Cov}(\mathbb{X}_1, \mathbb{X}_2)$ are sample quantities and the "hat" symbol ($\widehat{\ }$) in combination with a function represents an estimator and in combination with a parameter represents an estimate. The operator $\|.\|_m$ denotes the L-m norm. Subsequent pairs of linear combinations can be determined but for the entirety of this paper we consider only the first and most correlated pair.

The quantities of interest for CCA are the estimated *coefficient vectors* $\widehat{\boldsymbol{w}}_1$ and $\widehat{\boldsymbol{w}}_2$ that solve the above maximization problem, the resulting pairs of *canonical variables* $\mathbb{X}_1\widehat{\boldsymbol{w}}_1$ and $\mathbb{X}_2\widehat{\boldsymbol{w}}_2$, and their estimated *canonical correlation* $\hat{\rho}_{12}$. These are estimated quantities of the true coefficient vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$, true canonical variables $\mathbb{X}_1\boldsymbol{w}_1$ and $\mathbb{X}_2\boldsymbol{w}_2$, and true canonical correlation ρ_{12}.

Solutions to CCA exist given certain conditions.[16] Define the following sample variance-covariance matrix that holds all sample variances and covariances between variables in both $\mathbb{X}_1$ and $\mathbb{X}_2$:

$$\widehat{\boldsymbol{\Sigma}} = \begin{bmatrix} \widehat{\boldsymbol{\Sigma}}_{11} & \widehat{\boldsymbol{\Sigma}}_{12} \\ \widehat{\boldsymbol{\Sigma}}_{21} & \widehat{\boldsymbol{\Sigma}}_{22} \end{bmatrix}.$$

Then, $\widehat{\boldsymbol{\Sigma}}$ must be of full rank to obtain solutions to CCA. In addition, the maximum number of pairs of coefficient vectors will be less than $\min(p_1, p_2)$. Solutions are based upon the eigen-structure of elements from $\widehat{\boldsymbol{\Sigma}}$. Specifically, $\widehat{\boldsymbol{w}}_1$ and $\widehat{\boldsymbol{w}}_2$ are the first eigen-vectors of $\widehat{\boldsymbol{\Sigma}}_{11}^{-1}\widehat{\boldsymbol{\Sigma}}_{12}\widehat{\boldsymbol{\Sigma}}_{22}^{-1}\widehat{\boldsymbol{\Sigma}}_{21}$ and $\widehat{\boldsymbol{\Sigma}}_{22}^{-1}\widehat{\boldsymbol{\Sigma}}_{21}\widehat{\boldsymbol{\Sigma}}_{11}^{-1}\widehat{\boldsymbol{\Sigma}}_{12}$, respectively, and $\hat{\rho}_{12}$ is the square root of

the first corresponding eigen-value. Common alternate and related routes to solutions are through the singular value decomposition (SVD) [7] and the non-iterative partial least squares (NIPALS) algorithm [15].

4.2.2. Sparse Canonical Correlation Analysis (Sparse CCA)

Sparse CCA follows the above set-up with some adaptations. To obtain *sparse* coefficient vectors $\widehat{\boldsymbol{w}}_1$ and $\widehat{\boldsymbol{w}}_2$, additional constraints are applied to $\widehat{\boldsymbol{w}}_1$ and $\widehat{\boldsymbol{w}}_2$. This results in a sparse version of the objective function for CCA:

$$\underset{\boldsymbol{w}_1,\boldsymbol{w}_2}{\text{maximize}}\left\{\frac{\boldsymbol{w}_1\widehat{\boldsymbol{\Sigma}}_{12}\boldsymbol{w}_2}{\sqrt{\boldsymbol{w}_1\widehat{\boldsymbol{\Sigma}}_{11}\boldsymbol{w}_1}\sqrt{\boldsymbol{w}_2\widehat{\boldsymbol{\Sigma}}_{22}\boldsymbol{w}_2}}\right\}$$
$$subject\ to\ \|\boldsymbol{w}_1\|_2^2 = 1, \|\boldsymbol{w}_2\|_2^2 = 1$$
$$and\ P_1(\boldsymbol{w}_1) \le t_1, P_2(\boldsymbol{w}_2) \le t_2.$$

The *penalty functions* P_1 and P_2 can take on many forms, while being constrained by tuning parameters t_1 and t_2, which are specified by the user. Several versions of sparse CCA have emerged by choosing different penalty functions, or solving the optimization problem in different ways.[4,6,7,9,10,13,23–29] For example, Witten et al., 2009 considered the least absolute shrinkage and selection operator (LASSO) $P(\boldsymbol{w}) = \|\boldsymbol{w}\|_1$ and the fused LASSO $P(\boldsymbol{w}) = \sum_j |w_{j+1} - w_j|$.[10] Sparse CCA solutions are often obtained by transforming the objective function into a convex optimization problem and solving via alternating algorithms [10].

Tuning parameter selection is critical during the implementation of a sparse CCA method.[13] Authors of methodological papers have suggested a wide range of approaches. The most common are cross-validation strategies whereby sparse CCA is applied numerous times with different tuning parameters, and they often accompany code for executing the method itself.[7,13,26,30]

4.2.3. Bootstrap re-sampling for sparse CCA

The bootstrap technique is a means to estimate the sampling distribution of a random variable by sampling from a probability distribution or resampling from observed data.[31,32] In this paper, we utilize the non-parametric bootstrap method, which makes no assumptions regarding the underlying distribution from which data is generated.

We denote the data setup for sparse CCA as $[\mathbb{X}_1, \mathbb{X}_2]$, representing the pair of original datasets with dimensions $n \times p_1$ and $n \times p_2$, respectively. From this original pair of data, B bootstrapped pairs of datasets $[\mathbb{X}_1, \mathbb{X}_2]^{*1}, [\mathbb{X}_1, \mathbb{X}_2]^{*2}, \ldots, [\mathbb{X}_1, \mathbb{X}_2]^{*B}$ can be

obtained by taking random samples of size n with replacement from the n observations contained in the original pair of data. For each of these B bootstrapped pairs of samples, sparse CCA is applied, thereby returning B estimated canonical correlation values, denoted by $\hat{\rho}_{12}^{*1}, \hat{\rho}_{12}^{*2}, \dots, \hat{\rho}_{12}^{*B}$, and B sets of estimated coefficient vectors, denoted by $[\widehat{\boldsymbol{w}}_1, \widehat{\boldsymbol{w}}_2]^{*1}, [\widehat{\boldsymbol{w}}_1, \widehat{\boldsymbol{w}}_2]^{*2}, \dots, [\widehat{\boldsymbol{w}}_1, \widehat{\boldsymbol{w}}_2]^{*B}$. Summary measures can subsequently be used to infer about their distribution, notably their variability, and give us a level of confidence in the results.

Specifically, we focus on two primary measures of inference; one for canonical correlation and one for the canonical vector loadings. For the canonical correlations, we construct bootstrap confidence intervals using the empirical distribution[33], by taking 2.5th and 97.5th percentiles of the ordered B bootstrapped correlations to construct 95% confidence intervals. For the canonical vector loadings, for each variable's loading, we calculated the frequency at which the bootstrapped sparse CCA resulted in a non-zero estimate. We then calculated the proportion, where the denominator is the number of runs (B), which estimates the probability that a particular variable is involved in the cross-correlation. This is very important in practice because a single run of sparse CCA, as typically done, may not include all truly associated variables.

4.3. Simulations

In this section, we test the performance of the bootstrap approach for conducting inference in sparse CCA, where extensive simulations with a range of data scenarios are considered. We first describe the design of our simulation experiments and then report the results. Fundamentally, this section is devoted to testing the ability of the bootstrapping method to produce reliable inferential measures for sparse CCA. The R statistical software version 3.4.3 was used for performing all components of our simulation and codes will be made available upon request.[34]

4.3.1. Simulation Design

A depiction of our simulation design is presented in Figure 1. Our simulation strategy consists of the following seven steps, which are described in detail in this section:

1. Define the true data structure by selecting various values of the parameters n, p_1, p_2, Σ.
2. Calculate the true canonical vectors and canonical correlations by using the true Σ.
3. Simulate $R = 1000$ pairs of data, denoted $[\mathbb{X}_1^r, \mathbb{X}_2^r]$, for $r = 1, \dots, R$.

4. Bootstrap $B = 1000$ pairs of data, denoted $[\mathbb{X}_1^r, \mathbb{X}_2^r]^{*b}$, for $r = 1, \ldots, R$, $b = 1, \ldots, B$.
5. Apply sparse CCA to all $R * B$ pairs of data.
6. Calculate bootstrap confidence intervals and probability estimates of variable inclusion.
7. Summarize performance of these bootstrapped measures.

*** Figure 1 APPROXIMATELY HERE ***

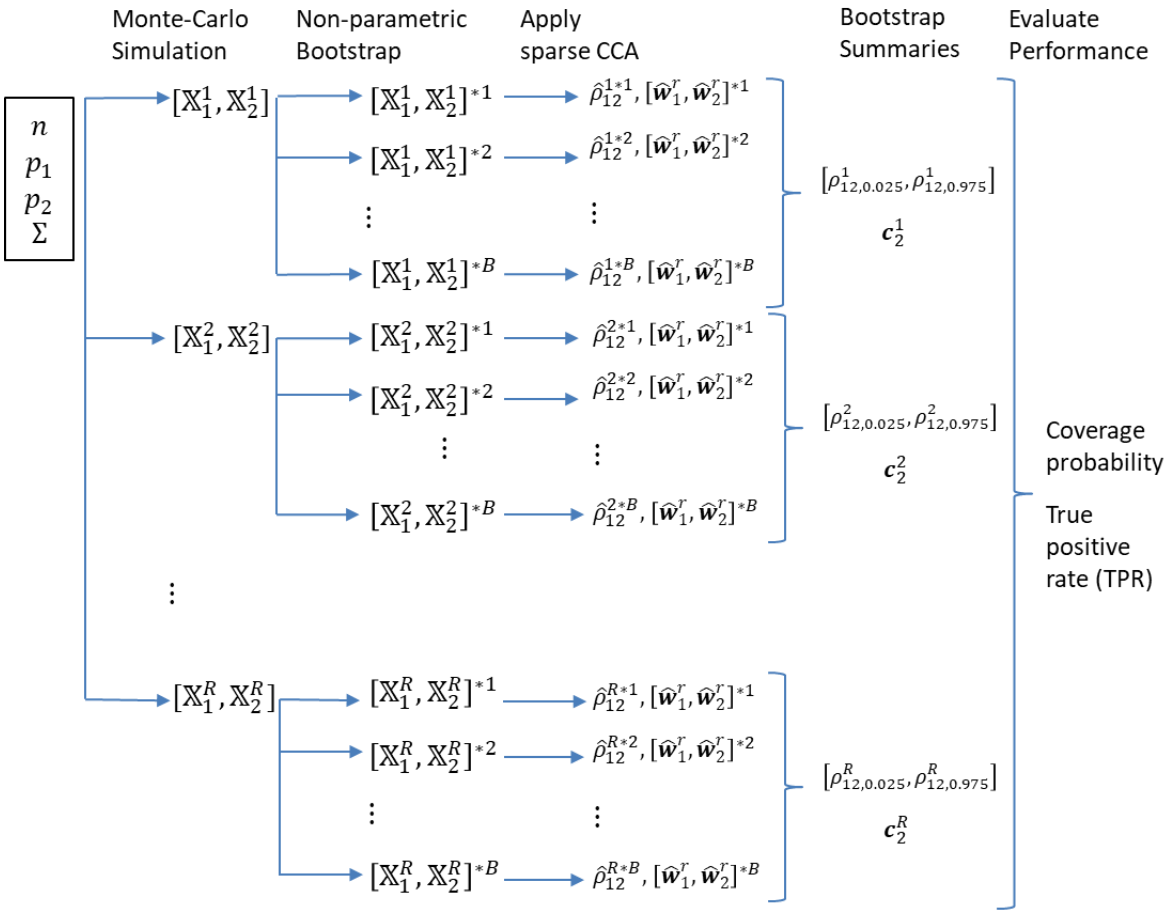


Figure 1: *A depiction of our simulation strategy for each simulation setting.*

Step 1 (define true parameters / data structure)

We used block-diagonal covariance matrices to specify simulated data structures. A more generalized definition for our simulation structure can be found in a previous

paper. [REF: Bonner2018Paper2] We denoted the covariance between all variables across both data domains as

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix},$$

where Σ_{11} is the covariance matrix for the p_1 variables from $\mathbb{X}_1$, Σ_{22} is the covariance matrix for the p_2 variables from $\mathbb{X}_2$, and $\Sigma_{12} = \Sigma_{21}^T$ is the cross-covariance matrix between the p_1 variables from $\mathbb{X}_1$ and p_2 variables from $\mathbb{X}_2$. Without loss of generality, we kept $p_1 = 10$ for the entirety of the simulations but allow p_2 to fluctuate from small to very large, assuming $p_{2,1} = 10$ of the variables from $\mathbb{X}_2$ to be cross-correlated with the p_1 variables from $\mathbb{X}_1$, leaving the remaining $p_{2,2} = p_2 - p_{2,1}$ variables as uncorrelated noise. The covariance matrices for $\mathbb{X}_1$ and $\mathbb{X}_2$ are defined as

$$\Sigma_{11} = \sigma_{11}^2 \begin{bmatrix} 1 & c_{11} & \dots & c_{11} \\ c_{11} & 1 & \dots & c_{11} \\ \vdots & \vdots & \ddots & \vdots \\ c_{11} & c_{11} & \dots & 1 \end{bmatrix}_{p_1 \times p_1}, \qquad \Sigma_{22} = \begin{bmatrix} \Sigma_{22,1} & \mathbf{0} \\ \mathbf{0} & \Sigma_{22,2} \end{bmatrix}_{p_2 \times p_2},$$

$$\Sigma_{22,1} = \sigma_{22,1}^2 \begin{bmatrix} 1 & c_{22,1} & \dots & c_{22,1} \\ c_{22,1} & 1 & \dots & c_{22,1} \\ \vdots & \vdots & \ddots & \vdots \\ c_{22,1} & c_{22,1} & \dots & 1 \end{bmatrix}_{p_{2,1} \times p_{2,1}}, \qquad \Sigma_{22,2} = \sigma_{22,2}^2 I_{p_{2,2} \times p_{2,2}},$$

where σ_{11}^2 and c_{11} are the common variance and correlation for variables within $\mathbb{X}_1$, $\sigma_{22,1}^2$ and $c_{22,1}$ are the common variance and correlation for the first $p_{2,1}$ variables within $\mathbb{X}_2$, and $\sigma_{22,2}^2$ and $c_{22,2} = 0$ are the common variance and correlation for the next $p_{2,2}$ variables within $\mathbb{X}_2$. We define the cross-covariance matrix as

$$\Sigma_{12} = \Sigma_{21}^T = \begin{bmatrix} \sigma_{11}^2 \sigma_{22,1}^2 c_{12} \mathbf{1}_{p_1, p_{2,1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where c_{12} is the common correlation shared between variables from $\mathbb{X}_1$ and the first $p_{2,1}$ variables from $\mathbb{X}_2$, and $\mathbf{1}_{a,b}$ is notation for a matrix of 1's with dimensions $a \times b$.

Table 1 lists our simulation scenarios in terms of the essential parameters described above. We tested 15 scenarios based on a combination of three possible cross-correlation values ($c_{12} = 0.8$, 0.4, or 0.2) and five possible variable sizes for $\mathbb{X}_2$

($p_2 = 50$, 100, 200, 500, or 1000). For each of these 15 scenarios, we ran simulations with a range of five possible sample sizes ($n = 50, 100, 200, 500$, or 1000), leading to a total of 75 data settings. Figure 2 displays a colored plot of Σ for simulation scenarios 1 and 15, demonstrating the scaling of dimension for $\mathbb{X}_2$ compared to the number of truly cross-correlated variables.

*** Table 1 APPROXIMATELY HERE ***

Table 1: A list of simulation scenarios we designed to test the performance of our methods. Each scenario was tested with sample sizes $n = 50, 100, 200, 500, 1000$.

Scenario	p_1	$\sigma^2_{11,1}$	$c_{11,1}$	p_2	$p_{2,1}, p_{2,2}$	$\sigma^2_{22,1}, \sigma^2_{22,2}$	$c_{22,1}, c_{22,2}$	c_{12}
1	10	10	0.8	50	10, 40	10, 6	0.8, 0	0.8
2	10	10	0.8	100	10, 90	10, 6	0.8, 0	0.8
3	10	10	0.8	200	10, 190	10, 6	0.8, 0	0.8
4	10	10	0.8	500	10, 490	10, 6	0.8, 0	0.8
5	10	10	0.8	1000	10, 990	10, 6	0.8, 0	0.8
6	10	10	0.5	50	10, 40	10, 6	0.5, 0	0.4
7	10	10	0.5	100	10, 90	10, 6	0.5, 0	0.4
8	10	10	0.5	200	10, 190	10, 6	0.5, 0	0.4
9	10	10	0.5	500	10, 490	10, 6	0.5, 0	0.4
10	10	10	0.5	1000	10, 990	10, 6	0.5, 0	0.4
11	10	10	0.5	50	10, 40	10, 6	0.5, 0	0.2
12	10	10	0.5	100	10, 90	10, 6	0.5, 0	0.2
13	10	10	0.5	200	10, 190	10, 6	0.5, 0	0.2
14	10	10	0.5	500	10, 490	10, 6	0.5, 0	0.2
15	10	10	0.5	1000	10, 990	10, 6	0.5, 0	0.2

*** Figure 2 APPROXIMATELY HERE ***

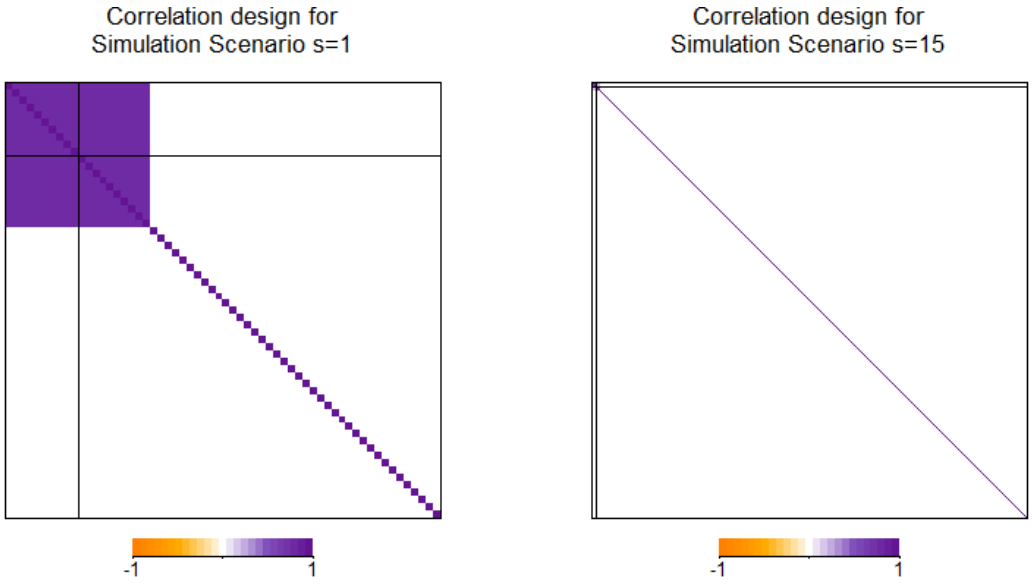


Figure 2: A colored plot of the Σ matrices behind the most extreme simulation scenarios; simulation 1 (left) with $p_2 = 50, c_{12} = 0.8$ and simulation 15 (right) with $p_2 = 1000, c_{12} = 0.2$. The thin black lines within the matrix help to distinguish variables from $\mathbb{X}_1$ and $\mathbb{X}_2$.

Step 2 (calculate true CCA components)

We performed the singular value decomposition of the matrix $K = \Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1/2} = UDV^T$ to obtain true CCA components, for all the 75 data settings. This involves the matrix Σ being positive semi-definite, as such we tested and ensured that our simulation scenarios resulted in a positive semi-definite Σ. The true canonical correlations were then calculated as $\rho_{12} = diag(D)$, the true canonical vectors are calculated as $W_1 = \Sigma_{11}^{-1/2}U$ and $W_2 = \Sigma_{22}^{-1/2}V$.

Step 3 (simulate $R = 1000$ pairs of data)

We used a multivariate normal distribution with mean vector $\boldsymbol{\mu} = \mathbf{0}$ and covariance matrix Σ to generate all R pairs of simulated data, using unique seeds to randomize a reproducible generation process. Figure 3 illustrates the comparability between the true Σ for simulation scenario 1 compared to estimates $\hat{\Sigma}$ using generated data at different sample sizes.

*** Figure 3 APPROXIMATELY HERE ***

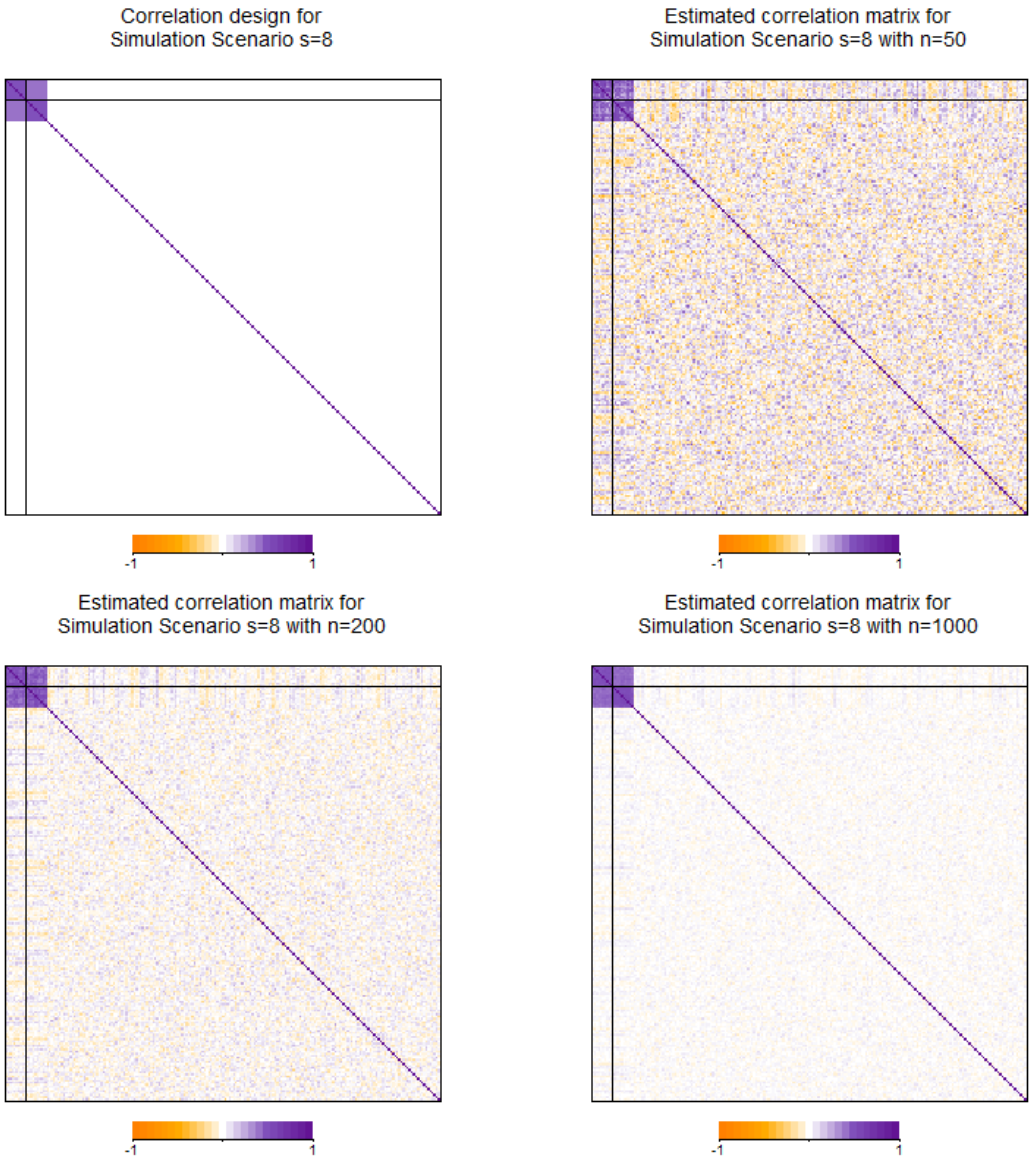


Figure 3: Colored images of Σ from our simulated design (top-left), and corresponding estimates $\hat{\Sigma}$ from $n = 50$ (top-right), 200 (bottom-left), and 1000 samples (bottom-right) generated from a multivariate normal distribution.

Step 4 (bootstrap $B = 1000$ pairs of data, for each $r = 1, \ldots, 1000$)

We used the non-parametric bootstrap approach described in Section 2.3 to generate all B pairs of bootstrapped data for each of the R pairs of simulated data. For the r^{th} pair of data, the b^{th} bootstrapped pair of data is denoted by $[\mathbb{X}_1^r, \mathbb{X}_2^r]^{*b}$. Again, unique seeds were set and used to make the generation process reproducible.

Step 5 (apply sparse CCA to all pairs of data)

We applied the sparse CCA approach developed by Parkhomenko et al., 2009 to each pair of data $[\mathbb{X}_1^r, \mathbb{X}_2^r]^{*b}$, $r = 1, \ldots, 1000$, $b = 1, \ldots, 1000$.[7] We chose this sparse CCA method over others [10,13] because it out-performed them when tested via rigorous simulation experiments, [REF: Bonner2018Paper2] and showed minimal, if any, bias and good variable selection properties including true-positive rate, true-negative rate, and overall sparsity.

Two tuning parameters need to be chosen while using the sparse CCA method proposed by Parkhomenko et al., 2009; λ_1 controls sparsity in $\boldsymbol{w}_1$ and λ_2 controls sparsity in $\boldsymbol{w}_2$. The possible range of tuning parameters for this method is from 0 (no sparsity) to 2 (maximum theoretical sparsity). We designed our simulation scenarios such that all p_1 variables from $\mathbb{X}_1$ would be involved in the cross-correlation. Therefore, we set $\lambda_1 = 0$. We considered but chose against using one of the many tuning parameter selection strategies we implemented in a previous work (see Bonner et al., 2018 [REF: Bonner2018Paper2] for a brief summary). Selecting tuning parameters optimally for sparse CCA is still an open research problem and if a tuning parameter selection strategy performs poorly, it may confound our conclusions regarding the performance of the bootstrap. As well, many tuning parameter selection strategies involve heavy, iterative computation, which would drastically inflate overall execution time for simulations when combined with the bootstrapping.

Instead, we chose λ_2 such that the resulting number of non-zero loadings in $\widehat{\boldsymbol{w}}_2$ would equal $p_{2,1}$, the true number of variables involved in the cross-correlation. This was achieved by an iterative search approach starting at $\lambda_2 = 0$ (no sparsity), increasing λ_2 and re-running if the number of non-zero loadings in the estimated $\widehat{\boldsymbol{w}}_2$ was more than $p_{2,1}$, decreasing λ_2 and re-running if the number was less, or accepting λ_2 and associated sparse CCA results if the estimated number of non-zeros in $\widehat{\boldsymbol{w}}_2$ was equal to $p_{2,1}$. As a safeguard to excessive computation, we always halved the 'step' of λ_2 between iterations and set the maximum number of computations for this iterative step to 20, accepting the results afterward.

We estimated canonical correlations $\hat{\rho}_{12}^{r*1}, \hat{\rho}_{12}^{r*2}, \ldots, \hat{\rho}_{12}^{r*B}$ and estimated coefficient vectors $[\widehat{\boldsymbol{w}}_1^r, \widehat{\boldsymbol{w}}_2^r]^{*1}, [\widehat{\boldsymbol{w}}_1^r, \widehat{\boldsymbol{w}}_2^r]^{*2}, \ldots, [\widehat{\boldsymbol{w}}_1^r, \widehat{\boldsymbol{w}}_2^r]^{*B}$, for $r = 1, \ldots, R$.

Step 6 (calculate bootstrapped measures of variability)

We calculated 95% basic bootstrap percentile confidence intervals for the canonical correlations $\left[\hat{\rho}^1_{12,0.025}, \hat{\rho}^1_{12,0.975}\right], \left[\hat{\rho}^2_{12,0.025}, \hat{\rho}^2_{12,0.975}\right], \ldots, \left[\hat{\rho}^R_{12,0.025}, \hat{\rho}^R_{12,0.975}\right]$ and calculated the proportion at which a variable's loading within its' canonical vector was non-zero, storing them in vectors denoted $\boldsymbol{c}_2^1, \boldsymbol{c}_2^2, \ldots, \boldsymbol{c}_2^R$ (we only applied sparsity to $\boldsymbol{w}_2$).

Step 7 (summarize performance of bootstrapped measures of variability)

We examined the coverage probability of the bootstrapped 95% confidence intervals for canonical correlations; which is the proportion of times they covered the true canonical correlation value. We calculated the true-positive rate (TPR) of the variables corresponding to the top 10 values within each $\boldsymbol{c}_2^r$ (i.e., the 10 variables most probable to be cross-correlated). We compared the mean TPR, across R simulation iterations, of those 'top-10' variables with the TPR of variables corresponding to non-zero loadings in $\boldsymbol{w}_2$ estimated at the simulation level. This revealed if the top 10 obtained using the bootstrap resampling could out-perform the sets estimated by single runs of sparse CCA.

4.3.2. Simulation Results

Quality assurance:

The sparse CCA algorithm successfully ran for all $R * B = 1{,}000{,}000$ pairs of data, for each of the 75 data settings. The iterative step to select tuning parameters converged before the maximum number of iterations in 99.986% of all runs. For the very small percentage of runs, for which the simulations did not converge, the number of variables from $\mathbb{X}_2$ estimated to be cross-correlated with $\mathbb{X}_1$ had a range of 2 to 19. Nevertheless, 90.9% of results estimated between 9 and 11 non-zero loadings (i.e., within one value of the true number of variables). All sparse CCA results were retained for summarizing and interpretation.

Canonical correlation:

Figure 4 shows estimated coverage of the bootstrap confidence intervals across all simulation scenarios. When the canonical correlation is large (left plot; $c_{12} = 0.8$, $\rho_{12} = 0.976$) the sample size and number of variables have little to no effect on the coverage. All simulation scenarios tend towards achieving nominal coverage, but have a small positive bias in coverage probability, especially for the larger sample sizes. The coverage probabilities range from 0.943 to 0.980 for the large correlation scenarios.

When the canonical correlation is designed to be moderate (middle plot; $c_{12} = 0.4$, $\rho_{12} = 0.727$), the effect of sample size and number of variables becomes

evident. At the larger sample sizes ($n = 200, 500, 1000$), coverage tends toward nominal regardless of the number of variables in $\mathbb{X}_2$; again, a small positive bias in coverage for the largest sample sizes ($n = 500,1000$), with coverage ranging from 0.950 to 0.978. At lower samples sizes ($n = 50,100$), however, coverage probability is lower than nominal and we start to see the effect of p_2. For $n = 100$ scenarios, coverage probability ranges from 0.878 at $p_2 = 50$ to 0.818 at $p_2 = 1000$. For $n = 50$, coverage probability starts at 0.665 at $p_2 = 50$ but quickly decreases towards 0 as p_2 increases; coverage is 0.001 at $p_2 = 1000$.

For the smallest canonical correlation scenarios (right plot; $c_{12} = 0.2$, $\rho_{12} = 0.364$), only bootstrap intervals built from the largest sample size scenarios ($n = 1000$) achieve nominal coverage; again, regardless of p_2. When $n = 500$, coverage ranges from 0.918 at $p_2 = 50$ to 0.846 at $p_2 = 1000$. When $n = 200$, coverage sharply declines from 0.464 at $p_2 = 50$ to almost or exactly 0 coverage. Bootstrap intervals made from smaller sample sizes ($n = 50,100$) have 0 coverage probability regardless of p_2.

*** Figure 4 APPROXIMATELY HERE ***

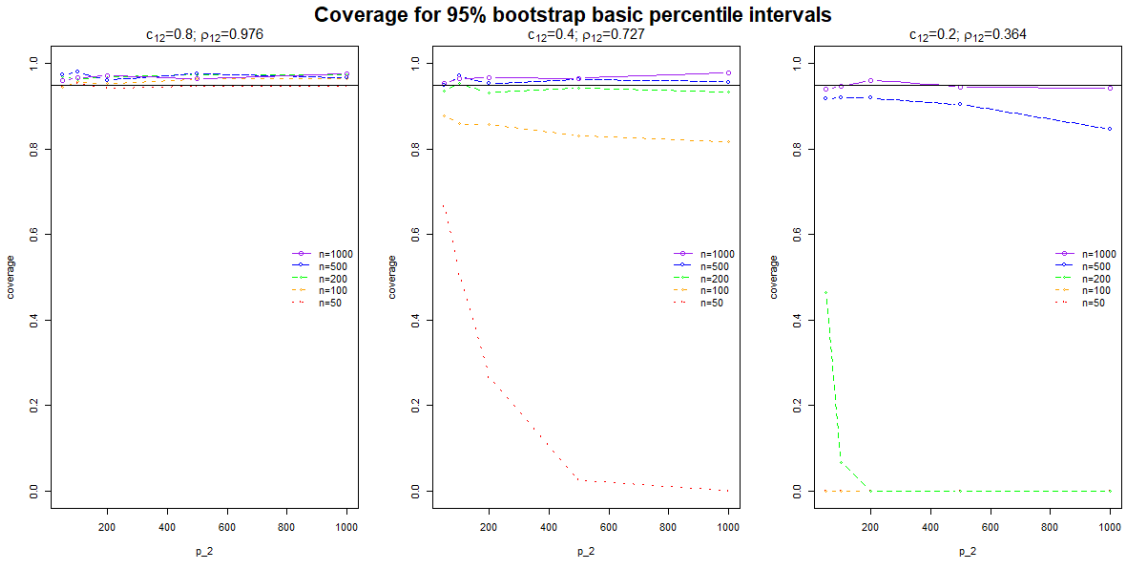


Figure 4: Plots for the coverage of 95% bootstrap confidence intervals for the canonical correlation coefficient from sparse CCA versus canonical correlation, sample size, and number of variables.

In Figure 5 we show 10% of the R bootstrap confidence intervals (selected at random) for each simulation scenario, excluding the moderate levels of p_2 $(100, 200, 500)$. These plots show that the lack of coverage is due to an overestimation of canonical correlation in some cases; a result that is emphasized when true canonical correlation and sample size becomes smaller.

*** Figure 5 APPROXIMATELY HERE ***

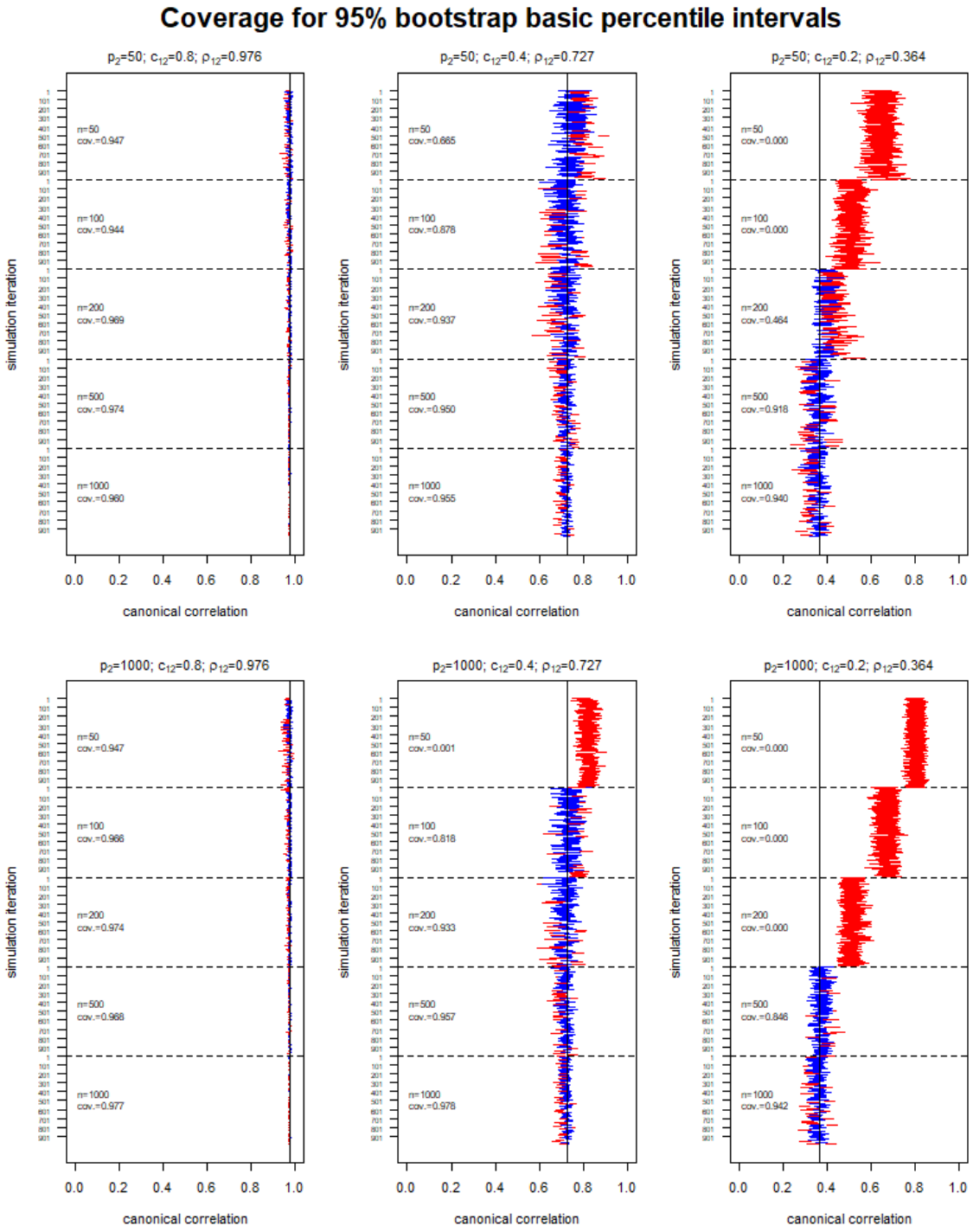


Figure 5: Plots of 95% bootstrap confidence intervals from simulation scenarios. A random selection of 10% (100) confidence intervals are presented for each scenario displayed. The back vertical lines represent the true canonical correlation coefficient for

the corresponding scenarios. Bootstrap intervals that overlap the true canonical correlation are highlighted blue, and those that do not are highlighted red.

Canonical vectors (variable selection):

Figure 6 displays our comparison of mean TPR from the simulation-level estimates of $\boldsymbol{w}_2$ and the 'top-10' list of variables derived from the bootstrapped estimates of $\boldsymbol{w}_2$. For higher canonical correlations, the sparse CCA method had perfect TPR. At lower canonical correlations, we see a clear discrepancy in TPR in favor of the bootstrapped 'top-10' list of variables that were deemed most likely to be involved in the cross-correlation. This discrepancy in TPR was larger for lower samples sizes, but did not seem related to number of variables in $\mathbb{X}_2$. However, the number of variables certainly affected sparse CCA performance in general, shown by decreasing TPR for both strategies for identifying cross-correlated variables.

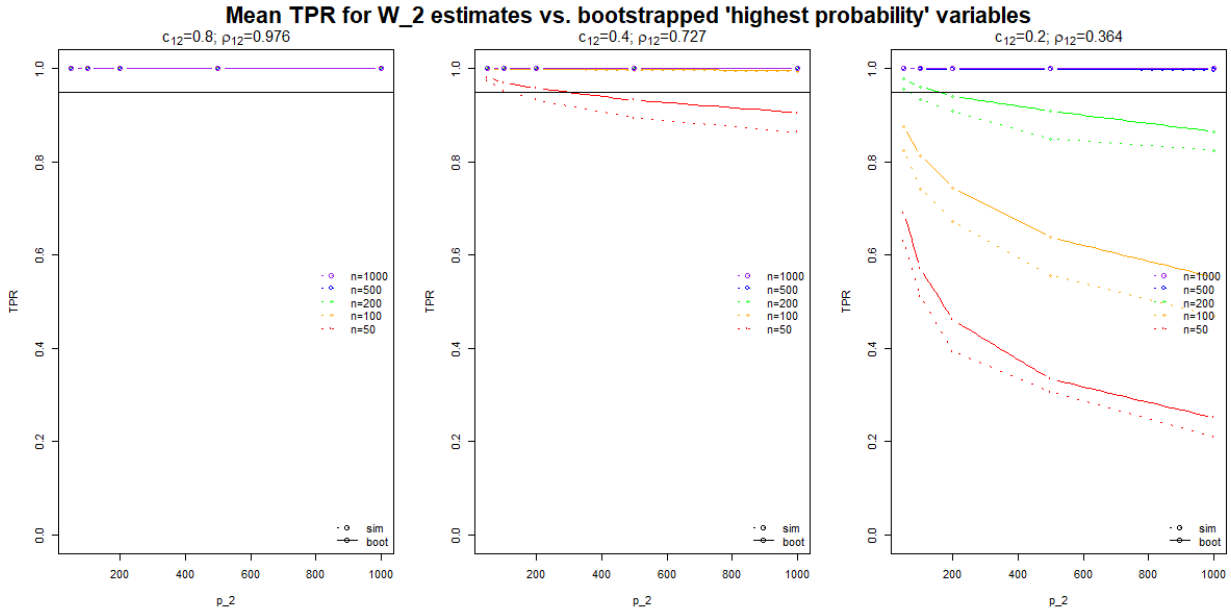


Figure 6: Plots of the TPR for simulation-level estimates of $\boldsymbol{w}_2$ vs. 'top-10' variables derived from the bootstrap-level estimates of $\boldsymbol{w}_2$.

4.4. Discussion

In this paper, we considered sparse CCA and the non-parametric bootstrap resampling approach, and conducted extensive simulations to investigate whether or not the non-parametric bootstrap resampling approach can offer reliable measures of inference for sparse CCA. Our findings show that the bootstrap confidence intervals for

the canonical correlation coefficients attain nominal coverage for higher values of canonical correlations and when sample size approaches the number of variables. However, results also showed that the confidence intervals lack coverage for smaller sample sizes, when correlation between data is weak. The bootstrap approach allowed us to estimate the probability at which a variable is involved in the cross-correlation. For lower canonical correlation values, the variables with highest bootstrapped probabilities had superior true positive rate (TPR) than those estimated from single-runs of sparse CCA. This finding should encourage further research into the benefits of bootstrapping with statistical methods that focus on variable selection. Similar attempts were made at improving variable selection via the bootstrap with sparse regression and CCA methods, with application in neurodevelopment.[22,35] These authors reported empirical improvements in the variable selection process, further encouraging the pursuit of research into such approaches.

Several improvements could be made to our simulation experiments. First, we only applied sparsity constraints to one data domain. Though this is suitable for some data applications, certain data scenarios will call for sparsity in both. We made this decision to mitigate a great deal of computation time; a common hindrance when applying the bootstrap in conjunction with another iterative procedure, such as tuning parameter selection. Although we used parallel computing, there are more advanced options to speed up computation times worth trying in the future. Using GPUs rather than CPUs, for example, can drastically improve computational power of sparse CCA and other iterative.[36] Second, although our approach to selecting tuning parameters allowed us to focus on the performance of the bootstrap, there would be no guarantee that such an approach would be accurate in practice. Fixing the number of variables is a suitable approach if the investigator has good rationale behind their choice. [9] However, in the absence of such information, a more objective tuning parameter selection method, such as maximizing some function of canonical correlation, would be more suitable. Third, there are many sparse CCA methods that we did not test in our simulations. Though we supported our choice of method based on comparative simulation work [REF: Bonner2018paper2], performance of the bootstrap measures is likely to differ based on which method is used. Extending our work to test different tuning parameter selection and sparse CCA methods could further illuminate the performance of the bootstrap approach.

We restricted our investigation of inferential measures to the canonical correlation coefficient and loading values. Another less popular, though important, output of CCA methods is the correlation between each variable contributing to the

canonical variates and the canonical variates themselves.[37] These are sometimes referred to as structural coefficients; loadings are sometimes referred to as functional coefficients. Building confidence intervals for structural coefficients could further reveal which variables truly drive to the relationships between data domains.

Sparse CCA and other multivariate integration methods are powerful tools to explore deep patterns among complex data. Resampling techniques can supply such methods with inferential measures to enhance their utility. Knowledge of how the bootstrap performs for sparse CCA could encourage researchers to pursue similar experiments for other methods. We encourage investigators to test and adopt bootstrapping approaches, especially in circumstances when sample size is large enough (i.e., when sample size approaches the number of variables).

References:

1. Chi EC, Allen GI, Zhou H, Kohannim O, Lange K, Thompson PM. Imaging genetics via sparse canonical correlation analysis. *Proc IEEE Int Symp Biomed Imaging*. 2013;2013:740-743. doi:10.1109/ISBI.2013.6556581.
2. Lin D, Calhoun VD, Wang Y-P. Correspondence between fMRI and SNP data by group sparse canonical correlation analysis. *Med Image Anal*. 2014;18(6):891-902. doi:10.1016/j.media.2013.10.010.
3. Lee G, Singanamalli A, Wang H, et al. Supervised Multi-View Canonical Correlation Analysis (sMVCCA): Integrating Histologic and Proteomic Features for Predicting Recurrent Prostate Cancer. *Med Imaging, IEEE Trans*. 2015;34(1):284-297. doi:10.1109/TMI.2014.2355175.
4. Wang H, Singanamalli A, Ginsburg S, Madabhushi A. Selecting features with group-sparse nonnegative supervised canonical correlation analysis: multimodal prostate cancer prognosis. *Med Image Comput Comput Assist Interv*. 2014;17(Pt 3):385-392. http://www.ncbi.nlm.nih.gov/pubmed/25320823. Accessed December 15, 2014.
5. Hotelling H. Relations between two sets of variates. *Biometrika*. 1936;28(3-4):321-377. doi:10.1093/biomet/28.3-4.321.
6. Waaijenborg S, Verselewel de Witt Hamer PC, Zwinderman AH. Quantifying the association between gene expressions and DNA-markers by penalized canonical correlation analysis. *Stat Appl Genet Mol Biol*. 2008;7(1):Article3. doi:10.2202/1544-6115.1329.
7. Parkhomenko E, Tritchler D, Beyene J. Sparse canonical correlation analysis with application to genomic data integration. *Stat Appl Genet Mol Biol*. 2009;8:Article 1. doi:10.2202/1544-6115.1406.
8. Bickel PJ, Li B, Tsybakov AB, et al. Regularization in statistics. *Test*. 2006;15(2):271-344.
9. Waaijenborg S, Zwinderman AH. Sparse canonical correlation analysis for identifying, connecting and completing gene-expression networks. *BMC Bioinformatics*. 2009;10:315. doi:10.1186/1471-2105-10-315.
10. Witten DM, Tibshirani R, Hastie T. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*. 2009;10(3):515-534. doi:10.1093/biostatistics/kxp008.
11. Lykou A, Whittaker J. Sparse CCA using a Lasso with positivity constraints. *Comput Stat Data Anal*. 2010;54(12):3144-3157.
12. Chen J, Bushman FD, Lewis JD, Wu GD, Li H. Structure-constrained sparse canonical correlation analysis with an application to microbiome data analysis. *Biostatistics*. 2013;14(2):244-258.
13. Wilms I, Croux C. Sparse canonical correlation analysis from a predictive point of view. *Biom J*. 2015;57(5):834-851. doi:10.1002/bimj.201400226.

14. Gao C, Ma Z, Zhou HH. An Efficient and Optimal Method for Sparse Canonical Correlation Analysis. 2014. http://arxiv.org/abs/1409.8565. Accessed December 15, 2014.
15. Lee W, Lee D, Lee Y, Pawitan Y. Sparse Canonical Covariance Analysis for High-throughput Data. *Stat Appl Genet Mol Biol*. 2011;10(1):1-24.
16. Johnson RA, Wichern DW. *Applied Multivariate Statistical Analysis*. 6th ed.; 2007. doi:10.1198/tech.2005.s319.
17. Fan X, Wang L. Comparability of Jackknife and Bootstrap Results: An Investigation for a Case of Canonical Correlation Analysis. *J Exp Educ*. 1996;64(2):173-189. doi:10.1080/00220973.1996.9943802.
18. Takane Y, Hwang H. Generalized Constrained Canonical Correlation Analysis. *Multivariate Behav Res*. 2002;37(2):163-195.
19. Sakar CO, Kursun O, Gurgen F. Ensemble canonical correlation analysis. *Appl Intell*. 2014;40(2):291-304.
20. Grellmann C, Bitzer S, Neumann J, et al. Comparison of variants of canonical correlation analysis and partial least squares for combined analysis of MRI and genetic data. *Neuroimage*. 2015;107:289-310. doi:10.1016/j.neuroimage.2014.12.025.
21. Helian S, Brumback BA, Cook RL. Sparse canonical correlation analysis between an alcohol biomarker and self-reported alcohol consumption. *Commun Stat - Simul Comput*. 2017;46(10):7924-7941. doi:10.1080/03610918.2016.1255971.
22. Szefer E, Lu D, Nathoo F, Beg MF, Graham J. Multivariate association between single-nucleotide polymorphisms in Alzgene linkage regions and structural changes in the brain: discovery, refinement and validation. *Stat Appl Genet Mol Biol*. 2017;16(5-6). doi:10.1515/sagmb-2016-0077.
23. Chu D, Liao L-Z, Ng MK, Zhang X. Sparse canonical correlation analysis: new formulation and algorithm. *IEEE Trans Pattern Anal Mach Intell*. 2013;35(12):3050-3065. doi:10.1109/TPAMI.2013.104.
24. Shen X, Sun Q. A novel semi-supervised canonical correlation analysis and extensions for multi-view dimensionality reduction. *J Vis Commun Image Represent*. 2014;25(8):1894-1904. doi:10.1016/j.jvcir.2014.09.004.
25. Yan J, Du L, Kim S, et al. Transcriptome-guided amyloid imaging genetic analysis via a novel structured sparse learning algorithm. *Bioinformatics*. 2014;30(17):i564-71. doi:10.1093/bioinformatics/btu465.
26. Witten DM, Tibshirani RJ. Extensions of Sparse Canonical Correlation Analysis with Applications to Genomic Data. *Stat Appl Genet Mol Biol*. 2009;8(1):29. doi:10.2202/1544-6115.1470.
27. Le Cao KA, Martin PGP, Robert-Granie C, Besse P. Sparse canonical methods for biological data integration: application to a cross-platform study. *BMC Bioinformatics*. 2009;10:17. doi:10.1186/1471-2105-10-34.
28. Hardoon DR, Shawe-Taylor J. Sparse canonical correlation analysis. *Mach Learn*.

2010;83(3):331-353. doi:10.1007/s10994-010-5222-7.
29. Lee I, Blom UM, Wang PI, Shim JE, Marcotte EM. Prioritizing candidate disease genes by network-based boosting of genome-wide association data. *Genome Res*. 2011;21(7):1109-1121. doi:10.1101/gr.118992.110.
30. Witten D, Tibshirani R, Gross S, Narasimhan B. PMA: Penalized Multivariate Analysis. 2013. https://cran.r-project.org/package=PMA.
31. Efron B, Tibshirani RJ. *An Introduction to the Bootstrap*. CRC press; 1994.
32. Mooney CZ, Duval RD. *Bootstrapping: A Nonparametric Approach to Statistical Inference*. Sage; 1993.
33. Davison AC, Hinkley D V. Bootstrap Methods and their Application. *Technometrics*. 1997. doi:10.2307/1271471.
34. R Core Team. R: A Language and Environment for Statistical Computing. 2017. https://www.r-project.org/.
35. Bunea F, She Y, Ombao H, Gongvatana A, Devlin K, Cohen R. Penalized least squares regression methods and applications to neuroimaging. *Neuroimage*. 2011;55(4):1519-1527. doi:10.1016/j.neuroimage.2010.12.028.
36. Yan J, Zhang H, Du L, Wernert E, Saykin AJ, Shen L. Accelerating Sparse Canonical Correlation Analysis for Large Brain Imaging Genetics Data. In: *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment - XSEDE '14*. New York, New York, USA: ACM Press; 2014:1-7. doi:10.1145/2616498.2616515.
37. Thompson B. Canonical correlation analysis: uses and interpretation. *Quant Appl Soc Sci*. 1984.

CHAPTER 5

5. SUMMARY AND CONCLUSIONS

As the pursuit of a system-wide understanding of complex diseases and traits continues, researchers are measuring omic data at a pace surpassing the development of statistical methodology we have to analyse it. The omic data is complex and diverse in structure, as well as large in dimension, positioning data analytics as a major bottleneck to discovery in the realm of health research.[78]

In this thesis, I applied and tested the performance of a variety of sparse multivariate statistical methods, to advance knowledge regarding their power and pitfalls for handling the integration of omic data. In Chapter 2, I focused on one set of data and an outcome, by analysing a toxicogenomic database that houses rich genomic information. By designing and performing an analysis pipeline involving sparse PCA and sparse regression, I extracted groups of genes that may be important to our understanding of the relative toxicity of drugs in humans. In Chapter 3, I expanded to two sets of data, and conducted extensive simulation experiments to compare the performance of multiple sparse CCA methods. After identifying the sparse CCA methods that perform best with real high-dimensional data, I applied them to find complex correlations between conventional toxicity assessment measures and gene expression. In Chapter 4, I investigated the performance and reliability of using bootstrapped measures of inference for sparse CCA. Using extensive simulations, I identified conditions for which bootstrap confidence intervals for the correlation coefficient perform well, and demonstrated that variables deemed most probable to be cross-correlated across bootstrap iterations can more accurately estimate underlying relationships than conventional sparse CCA estimates.

Work in this thesis informs health researchers aiming to infer complex relationships from large, high-dimensional ($n < p$) data. In particular, it promotes the class of sparse multivariate methods that can embrace the complex data structures presented by biological omic data. Although analysis strategies were applied to toxicogenomic data, they can also be used in other fields of study with equivalent success, so long as data properties remain similar. For instance, sparse PCA and CCA have recently experienced vibrant activity in neurology, where omic data have become seemingly ubiquitous.[54,79]

Results and conclusions from Chapters 3 and 4 should benefit statisticians who are currently working to improve sparse methodology. As more research teams adopt

sparse methods to analyze their data, this will further attract methods development. The extensive simulations presented here demonstrate how differently sparse CCA methods can perform with real data. They should encourage and provide a basis for developers of new approaches to thoroughly test their methods.

There are strengths that I have attempted to thread throughout the work presented in this thesis. First, I incorporated real data from a field of study that requires implementation. Toxicogenomic studies are in great need of innovative analysis strategies and, compared to other areas, have little application of the methods I've used.[80] Second, I've designed simulation experiments that mimic real data scenarios at high dimensions, but can also adapt to handle new methods and changes in data structure. The block-diagonal covariance design in my simulations is simple to adjust and generalize to different data. This contrasts more complicated simulation strategies that may depend on very specific data set ups.[81] Third, I attempted to make improvements to the coding behind sparse methods and simulation experiments; R code is available upon request and is planned to be made available through manuscripts as we continue to pursue publication. For example, I've sped up the tuning parameter selection procedures, where possible, by including a two-tiered grid search algorithm. This is less important for application, but helps greatly when conducting simulations or bootstrapping. Finally, I've attempted to carefully articulate each step of the sparse methodology implemented, including penalty functions, tuning parameter selection approaches, and summary measures. With many layers of methodology, reports of sparse multivariate method applications can lack in essential details for reproducibility.

There are also some limitations that come with certain methodological decisions made within this thesis. In the next few paragraphs, I summarize the main limitations alongside ideas to address them with future work. First, despite using real data to guide simulations, the scenarios still represent only a subset of the data structures that exist. For example, the toxicogenomic database included continuous measurements which made it convenient to parameterize simulations based on a multivariate normal distribution. Other omic data types, such as single-nucleotide polymorphism (SNP) data, involve non-continuous measures and could react very differently to sparse methodology than the results I presented in this thesis.

Second, the generalizability of my results is also limited by selecting a subset of methods available. Newer methods could outperform the methods investigated in this thesis. We focused mostly on methods that were popularized because they were published first and had readily available code. Other code packages and tools have emerged during the course of the work completed for this thesis and will likely promote

the adoption of more up to date methods.[82–84] For example, the R package 'mixOmics' has grown to include a variety of multivariate approaches, including regularized and sparse CCA and PLS.[83,85,86] This package is also home to some innovative visualization techniques that are otherwise in short supply for multivariate methods.[87]

Third, investigation of the performance of sparse CCA in Chapters 3 and 4 was restricted to the first canonical vectors only. In reality, subsequent CVs can carry important cross-correlations and should be examined by the investigator should there be reason to expect more than one set of correlated features. Though it is possible to extend both bodies of simulation work to test subsequent CVs, the effort would involve immense computation. To be accurate, tuning parameters need to be selected separately for each set of CVs, which would extend computation time significantly.[44] Another drawback to estimating more than one pair of CVs is the fact that sparsity constraints can intrude upon the orthogonality of CVs, meaning the cross-correlated groups of variables found in subsequent CVs could significantly overlap with the first.[36] This consequence is not necessarily a deterrent for implementation with real data, however, so the study of how accurate second, third, and so on, CVs are could be a valuable extension to the simulation studies presented in this thesis.

Finally, I had to make compromises while choosing tuning parameters. Determining the optimal ways to choose tuning parameters is still an area requiring dedicated research. Results from sparse PCA and sparse CCA can be sensitive to variations in the selected tuning parameter value.[20] Although isolating the best approaches was not the focus of my thesis, the simulations from my second project (Chapter 3) were able to demonstrate the effect tuning parameter selection could have. In other work, I attempted to either make informed decisions when choosing a selection approach (Chapter 1) or eliminate its influence entirely (Chapter 4). Investigators planning to use a sparse method should justify their choice of tuning parameter selection approach by consulting external simulations, such as the ones shown in this thesis, or by conducting their own. Certainly, at the very least, they should run the sparse method many times across the range of possible tuning parameters at a high resolution and investigate how the results change before accepting results.

Several attractive extensions to sparse CCA have emerged to handle specific data integration challenges and could be studied and tested in the future. Sparse 'multi-set' CCA has been the focus of a few groups, enabling estimation of cross-correlation between three or more data domains simultaneously.[44,88,89] By infusing penalty functions to conventional multiset CCA, these methods are poised to handle multi-omic databases like those emerging in toxicogenomics.[70,72,90,91] 'Supervised' sparse CCA can

incorporate an outcome to the objective of CCA, by influencing the estimation of cross-correlated variables to capture features that are jointly associated with the response.[44,92,93] Implementing such a method to find groups of cross-correlated genes and conventional measures for toxicity that are jointly associated with FDA-labeled DILI concern could help discern more concrete toxicity profiles in the TGP data, for example. A sparse CCA adaptation to find complex correlations between genomic data and longitudinal endpoints has recently been developed and could become pivotal as longitudinal studies ramp up the acquisition of omic data.[57] As well, the vast number of penalty functions available has propagated a wide array of sparse CCA methods that can handle more minor, yet important, structural intricacies of data domains.[30,81,94]

Sparse PCA, sparse CCA, and their variants are not the only methods to handle data integration problems. Borga described the link between objective functions for PCA, CCA, PLS, and multiple linear regression, by means of the generalized eigenproblem.[26] Sparse PLS methodology has been growing in tandem to sparse PCA and sparse CCA.[95,96] Although work in this thesis does not contribute directly to sparse PLS, it could naturally extend to involve it. For example, sparse PLS could be applied in Chapter 2, as a supervised alternative to the sparse PCA analysis. However, I would want to precede such an analysis with proper simulations first, to identify the most accurate sparse PLS method for the toxicogenomic data. Future work could include adapting the simulation infrastructure in my second project to test a variety of sparse PLS methods.

Between the alternating algorithms used to solve sparse CCA, grid-search tuning parameter selection methods that often involve cross-validation, and permutation tests for testing significance, there is a massive computational burden with executing sparse CCA. Though computation is a small cost for methodological rigour, it is certainly a restraint worth exploring solutions to. Yan et al., 2014 has discussed this issue and outlined strategies involving running sparse CCA algorithms in parallel to speed up the computation. Pursuing strategies such as this could catalyze further testing.[97]

Through more development, testing, and careful application, sparse methodology is maturing towards being the ideal toolset to handle massive data integration challenges. As data continues to barrage the health research landscape, we must arm ourselves with knowledge of the tools we have to analyze it. The work from this thesis should further our understanding of how sparse multivariate methods interact with high-dimensional omic data and offer some outlets for future work in this vibrant, cross-disciplinary research area.

References:

1. Jain SH, Rosenblatt M, Duke J. Is Big Data the New Frontier for Academic-Industry Collaboration? *JAMA*. 2014;311(21):2171. doi:10.1001/jama.2014.1845.

2. Chen M, Mao S, Liu Y. Big Data: A Survey. *Mob Networks Appl*. 2014;19(2):171-209. doi:10.1007/s11036-013-0489-0.

3. Murdoch TB, Detsky AS. The Inevitable Application of Big Data to Health Care. *JAMA*. 2013;309(13):1351. doi:10.1001/jama.2013.393.

4. Martin-Sanchez F, Verspoor K. Big Data in Medicine Is Driving Big Changes. *IMIA Yearb*. 2014;9(1):14-20. doi:10.15265/IY-2014-0020.

5. Metzker ML. Sequencing technologies the next generation. *Nat Rev Genet*. 2010. doi:10.1038/nrg2626.

6. Horgan RP, Kenny LC. 'Omic' technologies: genomics, transcriptomics, proteomics and metabolomics. *Obstet Gynaecol*. 2011;13(3):189-195. doi:10.1576/toag.13.3.189.27672.

7. Obermeyer Z, Emanuel EJ. Predicting the Future — Big Data, Machine Learning, and Clinical Medicine. *N Engl J Med*. 2016;375(13):1216-1219. doi:10.1056/NEJMp1606181.

8. Brazhnik O, Jones JF. Anatomy of data integration. *J Biomed Inform*. 2007;40(3):252-269. doi:10.1016/j.jbi.2006.09.001.

9. Louie B, Mork P, Martin-Sanchez F, Halevy A, Tarczy-Hornoch P. Data integration and genomic medicine. *J Biomed Inf*. 2007;40(1):5-16. doi:10.1016/j.jbi.2006.02.007.

10. Hamid JS, Hu P, Roslin NM, Ling V, Greenwood CMT, Beyene J. Data integration in genetics and genomics: methods and challenges. *Hum Genomics Proteomics*. 2009;2009. doi:10.4061/2009/869093.

11. Edgar R. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res*. 2002;30(1):207-210. doi:10.1093/nar/30.1.207.

12. Tomczak K, Czerwińska P, Wiznerowicz M. Review The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge. *Współczesna Onkol*. 2015;1A:68-77. doi:10.5114/wo.2014.47136.

13. Bickeböller H, Bailey JN, Beyene J, et al. Genetic Analysis Workshop 18: Methods and strategies for analyzing human sequence and phenotype data in members of

extended pedigrees. *BMC Proc*. 2014;8(Suppl 1):S1. doi:10.1186/1753-6561-8-S1-S1.

14. Ritchie MD, Holzinger ER, Li R, Pendergrass SA, Kim D. Methods of integrating data to uncover genotype-phenotype interactions. *Nat Rev Genet*. 2015;16(2):85-97. doi:10.1038/nrg3868.

15. Schumann G. Are we doing enough to extract genomic information from our data? *Psychophysiology*. 2014;51(12):1335-1336. doi:10.1111/psyp.12357.

16. Jolliffe I. Principal component analysis. In: *International Encyclopedia of Statistical Science*. Springer; 2011:1094-1096.

17. Wold H. Partial Least Squares. Kotz S, Johnson NL, eds. *Int J Cardiol*. 1985;147(2):581-591. doi:10.1016/j.ijcard.2010.12.060.

18. Thompson B. Canonical correlation analysis: uses and interpretation. *Quant Appl Soc Sci*. 1984.

19. Marian AJ. Molecular genetic studies of complex phenotypes. *Transl Res*. 2012;159(2):64-79. doi:10.1016/j.trsl.2011.08.001.

20. Waaijenborg S, Verselewel de Witt Hamer PC, Zwinderman AH. Quantifying the association between gene expressions and DNA-markers by penalized canonical correlation analysis. *Stat Appl Genet Mol Biol*. 2008;7(1):Article3. doi:10.2202/1544-6115.1329.

21. Bickel PJ, Li B, Tsybakov AB, et al. Regularization in statistics. *Test*. 2006;15(2):271-344.

22. Le Cao KA, Martin PGP, Robert-Granie C, Besse P. Sparse canonical methods for biological data integration: application to a cross-platform study. *BMC Bioinformatics*. 2009;10:17. doi:10.1186/1471-2105-10-34.

23. Pearson K. Principal components analysis. *London, Edinburgh, Dublin Philos Mag J Sci*. 1901;6(2):559.

24. Johnson RA, Wichern DW. *Applied Multivariate Statistical Analysis*. 6th ed.; 2007. doi:10.1198/tech.2005.s319.

25. Hotelling H. Relations between two sets of variates. *Biometrika*. 1936;28(3-4):321-377. doi:10.1093/biomet/28.3-4.321.

26. Borga M, Landelius T, Knutsson H. *A Unified Approach to Pca, Pls, Mlr and Cca*. Citeseer; 1997.

http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.1128.

27. Hoerl AE, Kennard RW. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*. 1970;12(1):55-67.

28. Tibshirani R. Regression shrinkage and selection via the lasso. *J R Stat Soc Ser B*. 1996:267-288.

29. Zou H, Hastie T. Regularization and variable selection via the elastic net. *J R Stat Soc Ser B-Statistical Methodol*. 2005;67:301-320. doi:10.1111/j.1467-9868.2005.00503.x.

30. Tibshirani R. Regression shrinkage and selection via the lasso: A retrospective. *J R Stat Soc Ser B Stat Methodol*. 2011;73(3):273-282.

31. Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. *J R Stat Soc Ser B Stat Methodol*. 2006;68:49-67. doi:10.1111/j.1467-9868.2005.00532.x.

32. Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K. Sparsity and smoothness via the fused lasso. *J R Stat Soc Ser B-Statistical Methodol*. 2005;67:91-108. doi:10.1111/j.1467-9868.2005.00490.x.

33. Zou H. The adaptive lasso and its oracle properties. *J Am Stat Assoc*. 2006;101(476):1418-1429.

34. Zou H, Hastie T, Tibshirani R. Sparse principal component analysis. *J Comput Graph Stat*. 2006;15(2):265-286. doi:10.1198/106186006x113430.

35. Shen H, Huang JZ. Sparse principal component analysis via regularized low rank matrix approximation. *J Multivar Anal*. 2008;99(6):1015-1034. doi:10.1016/j.jmva.2007.06.007.

36. Witten DM, Tibshirani R, Hastie T. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*. 2009;10(3):515-534. doi:10.1093/biostatistics/kxp008.

37. Leng C, Wang H. On General Adaptive Sparse Principal Component Analysis. *J Comput Graph Stat*. 2009;18(1):201-215. doi:10.1198/jcgs.2009.0012.

38. Lee D, Lee W, Lee Y, Pawitan Y. Super-sparse principal component analyses for high-throughput genomic data. *BMC Bioinformatics*. 2010;11:296. doi:10.1186/1471-2105-11-296.

39. Journee M, Nesterov Y, Richtarik P, Sepulchre R. Generalized Power Method for

Sparse Principal Component Analysis. *J Mach Learn Res*. 2010;11:517-553.

40. Xiao C. Two-Dimensional Sparse Principal Component Analysis for Palmprint Recognition. In: Huang DS, Zhao ZM, Bevilacqua V, Figueroa JC, eds. *Advanced Intelligent Computing Theories and Applications*.Vol 6215.; 2010:611-618.

41. Guo J, James G, Levina E, Michailidis G, Zhu J. Principal Component Analysis With Sparse Fused Loadings. *J Comput Graph Stat*. 2010;19(4):930-946. doi:10.1198/jcgs.2010.08127.

42. Lee S, Huang JZ, Hu J. Sparse logistic principal components analysis for binary data. *Ann Appl Stat*. 2010;4(3):1579-1601. doi:10.1214/10-aoas327.

43. Cai TT, Ma Z, Wu Y. Sparse PCA: optimal rates and adaptive estimation. *Ann Stat*. 2013;41(6):3074-3110. doi:10.1214/13-aos1178.

44. Witten DM, Tibshirani RJ. Extensions of Sparse Canonical Correlation Analysis with Applications to Genomic Data. *Stat Appl Genet Mol Biol*. 2009;8(1):29. doi:10.2202/1544-6115.1470.

45. Waaijenborg S, Zwinderman AH. Sparse canonical correlation analysis for identifying, connecting and completing gene-expression networks. *BMC Bioinformatics*. 2009;10:315. doi:10.1186/1471-2105-10-315.

46. Waaijenborg S, Zwinderman AH. Correlating multiple SNPs and multiple disease phenotypes: penalized non-linear canonical correlation analysis. *Bioinformatics*. 2009;25(21):2764-2771. doi:10.1093/bioinformatics/btp491.

47. Parkhomenko E, Tritchler D, Beyene J. Sparse canonical correlation analysis with application to genomic data integration. *Stat Appl Genet Mol Biol*. 2009;8:Article 1. doi:10.2202/1544-6115.1406.

48. Chu D, Liao L-Z, Ng MK, Zhang X. Sparse canonical correlation analysis: new formulation and algorithm. *IEEE Trans Pattern Anal Mach Intell*. 2013;35(12):3050-3065. doi:10.1109/TPAMI.2013.104.

49. Coleman J, Replogle J, Chandler G, Hardin J. Resistant Sparse Multiple Canonical Correlation. 2014. http://arxiv.org/abs/1410.3355.

50. Lee W, Lee D, Lee Y, Pawitan Y. Sparse Canonical Covariance Analysis for High-throughput Data. *Stat Appl Genet Mol Biol*. 2011;10(1):1-24.

51. Gao C, Ma Z, Zhou HH. An Efficient and Optimal Method for Sparse Canonical Correlation Analysis. 2014. http://arxiv.org/abs/1409.8565. Accessed December 15, 2014.

52. Wilms I, Croux C. Sparse canonical correlation analysis from a predictive point of view. *Biom J*. 2015;57(5):834-851. doi:10.1002/bimj.201400226.

53. Lin DD, Zhang JG, Li JY, Calhoun VD, Deng HW, Wang YP. Group sparse canonical correlation analysis for genomic data integration. *BMC Bioinformatics*. 2013;14:16. doi:10.1186/1471-2105-14-245.

54. Lin D, Calhoun VD, Wang Y-P. Correspondence between fMRI and SNP data by group sparse canonical correlation analysis. *Med Image Anal*. 2014;18(6):891-902. doi:10.1016/j.media.2013.10.010.

55. Lykou A, Whittaker J. Sparse CCA using a Lasso with positivity constraints. *Comput Stat Data Anal*. 2010;54(12):3144-3157.

56. Chen J, Bushman FD, Lewis JD, Wu GD, Li H. Structure-constrained sparse canonical correlation analysis with an application to microbiome data analysis. *Biostatistics*. 2013;14(2):244-258.

57. Hao X, Li C, Yan J, et al. Identification of associations between genotypes and longitudinal phenotypes via temporally-constrained group sparse canonical correlation analysis. *Bioinformatics*. 2017;33(14):i341-i349. doi:10.1093/bioinformatics/btx245.

58. Wang H, Singanamalli A, Ginsburg S, Madabhushi A. Selecting features with group-sparse nonnegative supervised canonical correlation analysis: multimodal prostate cancer prognosis. *Med Image Comput Comput Assist Interv*. 2014;17(Pt 3):385-392. http://www.ncbi.nlm.nih.gov/pubmed/25320823. Accessed December 15, 2014.

59. Witten D, Tibshirani R, Gross S, Narasimhan B. PMA: Penalized Multivariate Analysis. 2013. https://cran.r-project.org/package=PMA.

60. Bonner A. Sparse Principal Component Analysis for High-Dimensional Data: A Comparative Study. 2012.

61. Bonner A, Neupane B, Beyene J. Testing for associations between systolic blood pressure and single-nucleotide polymorphism profiles obtained from sparse principal component analysis. In: *BMC Proceedings*.Vol 8.; 2014:S95.

62. Lu AT-H, Austin E, Bonner A, Huang H-H, Cantor RM. Applications of machine learning and data mining methods to detect associations of rare and common variants with complex traits. *Genet Epidemiol*. 2014;38 Suppl 1:S81-5. doi:10.1002/gepi.21830.

63. Slater T, Bouton C, Huang ES. Beyond data integration. *Drug Discov Today*.

2008;13(13-14):584-589. doi:10.1016/j.drudis.2008.01.008.

64. Balmer N V., Dao T, Leist M, et al. Application of “Omics” Technologies to In Vitro Toxicology. In: ; 2014:399-432. doi:10.1007/978-1-4939-0521-8_18.

65. Uehara T, Ono A, Maruyama T, et al. The Japanese toxicogenomics project: application of toxicogenomics. *Mol Nutr Food Res*. 2010;54(2):218-227. doi:10.1002/mnfr.200900169.

66. Igarashi Y, Nakatsu N, Yamashita T, et al. Open TG-GATEs: a large-scale toxicogenomics database. *Nucleic Acids Res*. 2015;43(Database issue):D921-7. doi:10.1093/nar/gku955.

67. Ganter B, Snyder RD, Halbert DN, Lee MD. Toxicogenomics in drug discovery and development: mechanistic analysis of compound/class-dependent effects using the DrugMatrix database. *Pharmacogenomics*. 2006;7(7):1025-1044. doi:10.2217/14622416.7.7.1025.

68. Boitier E, Amberg A, Barbie V, et al. A comparative integrated transcript analysis and functional characterization of differential mechanisms for induction of liver hypertrophy in the rat. *Toxicol Appl Pharmacol*. 2011;252(2):85-96. doi:10.1016/j.taap.2011.01.021.

69. Suter L, Schroeder S, Meyer K, et al. EU framework 6 project: predictive toxicology (PredTox)--overview and outcome. *Toxicol Appl Pharmacol*. 2011;252(2):73-84. doi:10.1016/j.taap.2010.10.008.

70. Taboureau O, Hersey A, Audouze KML, et al. Toxicogenomics investigation under the eTOX Project. *J Pharmacogenomics Pharmacoproteomics*. 2012.

71. Briggs K, Cases M, Heard DJ, et al. Inroads to Predict in Vivo Toxicology-An Introduction to the eTOX Project. *Int J Mol Sci*. 2012;13(3):3820-3846. doi:10.3390/ijms13033820.

72. Ellinger-Ziegelbauer H, Adler M, Amberg A, et al. The enhanced value of combining conventional and “omics” analyses in early assessment of drug-induced hepatobiliary injury. *Toxicol Appl Pharmacol*. 2011;252(2):97-111. doi:10.1016/j.taap.2010.09.022.

73. Bonner AJ, Beyene J. Evaluating the Performance of Sparse Principal Component Analysis Methods in High-dimensional Data Scenarios. *Commun Stat - Simul Comput*. 2016:0-0. doi:10.1080/03610918.2015.1004268.

74. Takane Y, Hwang H, Abdi H. Regularized Multiple-Set Canonical Correlation Analysis. *Psychometrika*. 2008;73(4):753-775. doi:10.1007/s11336-008-9065-0.

75. Grellmann C, Bitzer S, Neumann J, et al. Comparison of variants of canonical correlation analysis and partial least squares for combined analysis of MRI and genetic data. *Neuroimage*. 2015;107:289-310. doi:10.1016/j.neuroimage.2014.12.025.

76. Helian S, Brumback BA, Cook RL. Sparse canonical correlation analysis between an alcohol biomarker and self-reported alcohol consumption. *Commun Stat - Simul Comput*. 2017;46(10):7924-7941. doi:10.1080/03610918.2016.1255971.

77. Szefer E, Lu D, Nathoo F, Beg MF, Graham J. Multivariate association between single-nucleotide polymorphisms in Alzgene linkage regions and structural changes in the brain: discovery, refinement and validation. *Stat Appl Genet Mol Biol*. 2017;16(5-6). doi:10.1515/sagmb-2016-0077.

78. Alyass A, Turcotte M, Meyre D. From big data analysis to personalized medicine for all: challenges and opportunities. *BMC Med Genomics*. 2015;8(1):33. doi:10.1186/s12920-015-0108-y.

79. Sui J, Adali T, Yu Q, Chen J, Calhoun VD. A review of multivariate methods for multimodal fusion of brain imaging data. *J Neurosci Methods*. 2012;204:68-81. doi:10.1016/j.jneumeth.2011.10.031.

80. Chalise P, Batzler A, Abo R, Wang L, Fridley BL. Simultaneous analysis of multiple data types in pharmacogenomic studies using weighted sparse canonical correlation analysis. *OMICS*. 2012;16(7-8):363-373. doi:10.1089/omi.2011.0126.

81. Chalise P, Fridley BL. Comparison of penalty functions for sparse canonical correlation analysis. *Comput Stat Data Anal*. 2012;56(2):245-254.

82. Lê Cao K-A, González I, Déjean S. integrOmics: an R package to unravel relationships between two omics datasets. *Bioinformatics*. 2009;25(21):2855-2856. doi:10.1093/bioinformatics/btp515.

83. Dejean S, Gonzalez I, with contributions from Pierre Monget K-ALC, et al. mixOmics: Omics Data Integration Project. 2014. http://cran.r-project.org/package=mixOmics.

84. Henry VJ, Bandrowski AE, Pepin A-S, Gonzalez BJ, Desfeux A. OMICtools: an informative directory for multi-omic data analysis. *Database (Oxford)*. 2014;2014. doi:10.1093/database/bau069.

85. Tenenhaus A, Tenenhaus M. Regularized Generalized Canonical Correlation Analysis. *Psychometrika*. 2011;76(2):257-284.

86. Tenenhaus A, Philippe C, Guillemot V, Le Cao K-A, Grill J, Frouin V. Variable

selection for generalized canonical correlation analysis. *Biostatistics*. 2014:1-15. doi:10.1093/biostatistics/kxu001.

87. Gonzalez I, Cao K-A Le, Davis MJ, Dejean S. Visualising associations between paired “omics” data sets. *BioData Min [electronic Resour*. 2012;5(1):19. doi:http://dx.doi.org/10.1186/1756-0381-5-19.

88. Shen X, Sun Q. A novel semi-supervised canonical correlation analysis and extensions for multi-view dimensionality reduction. *J Vis Commun Image Represent*. 2014;25(8):1894-1904. doi:10.1016/j.jvcir.2014.09.004.

89. Lee G, Singanamalli A, Wang H, et al. Supervised Multi-View Canonical Correlation Analysis (sMVCCA): Integrating Histologic and Proteomic Features for Predicting Recurrent Prostate Cancer. *Med Imaging, IEEE Trans*. 2015;34(1):284-297. doi:10.1109/TMI.2014.2355175.

90. Horst P. Relations among m sets of measures. *Psychometrika*. 1961;26:129-149. doi:10.1007/BF02289710.

91. Kettenring JR. Canonical analysis of several sets of variables. *Biometrika*. 1971;58(3):433-451.

92. Golugula A, Lee G, Master SR, et al. Supervised regularized canonical correlation analysis: integrating histologic and proteomic measurements for predicting biochemical recurrence following prostate surgery. *BMC Bioinformatics*. 2011;12:483. doi:10.1186/1471-2105-12-483.

93. Sun L, Ji S, Ye J. Canonical correlation analysis for multilabel classification: a least-squares formulation, extensions, and analysis. *IEEE Trans Pattern Anal Mach Intell*. 2011;33(1):194-200. doi:10.1109/TPAMI.2010.160.

94. Huang J, Zhang T. The benefit of group sparsity. *Ann Stat*. 2010;38(4):1978-2004. doi:10.1214/09-AOS778.

95. Le Cao K-A, Le Gall C. Integration and variable selection of ‘ omics ’ data sets with PLS : a survey. *J la Société Française Stat*. 2011;152(2):77–96. http://www.statistique-et-enseignement.fr/ojs/index.php/J-SFdS/article/viewFile/64/55.

96. Lê Cao K-A, Rossouw D, Robert-Granié C, Besse P. A sparse PLS for variable selection when integrating omics data. *Stat Appl Genet Mol Biol*. 2008;7(1):Article 35. doi:10.2202/1544-6115.1390.

97. Yan J, Zhang H, Du L, Wernert E, Saykin AJ, Shen L. Accelerating Sparse Canonical Correlation Analysis for Large Brain Imaging Genetics Data. In: *Proceedings of the*

2014 Annual Conference on Extreme Science and Engineering Discovery Environment - XSEDE '14. New York, New York, USA: ACM Press; 2014:1-7. doi:10.1145/2616498.2616515.

Appendix:

Search strategy of omic data integration articles:

The literature review I summarize in Chapter 1: Introduction is described in detail here. On September 14, 2018, using OVID, I searched the U.S. National Library of Medicine's database MEDLINE, from the year 1946 and onward, for articles that contained, within their title or abstract, terminology related to both data integration and omic data. I submitted the following keyword search strategy using OVID Medline:

1. data integration.ab,ti.
2. data fusion.ab,ti.
3. integrative analysis.ab,ti.
4. 1 or 2 or 3
5. genetics.ab,ti.
6. genom*.ab,ti.
7. transcriptom*.ab,ti.
8. proteome*.ab,ti.
9. metabolom*.ab,ti.
10. metabonom*.ab,ti.
11. methyl*.ab,ti.
12. microbiom*.ab,ti.
13. 5 or 6 or 7 or 8 or 9 or 10 or 11 or 12
14. 4 or 13

The final keyword search (14. from above) returned 1858 articles. Figure 1 in the Introduction section of this thesis presents the frequency of articles published per year; the increasing frequency of publications over time is an indicator of elevated interest in and application of data integration methodology.