Title

Large-Scale Interpretable Multi-View Learning for Very High-Dimensional Problems with Application to Multi-Omic Data

Permalink

https://escholarship.org/uc/item/3w7613s4

Author

Shams Solari, Omid

Publication Date

2019

Peer reviewed|Thesis/dissertation

Large-Scale Interpretable Multi-View Learning for Very High-Dimensional Problems with Application to Multi-Omic Data

by

Omid Shams Solari

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Peter J. Bickel, Chair
Professor Haiyan Huang
Professor Gary Karpen
Dr. James B. Brown

Fall 2019

Large-Scale Interpretable Multi-View Learning for Very High-Dimensional Problems with Application to Multi-Omic Data

Abstract

Large-Scale Interpretable Multi-View Learning for Very High-Dimensional Problems with Application to Multi-Omic Data

by

Omid Shams Solari

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Peter J. Bickel, Chair

We discuss the sparse Canonical Correlation Analysis (CCA) problem in the context of high-dimensional multi-view problems, where we aim to discover interpretable association structures among multiple random vectors via their respective *views* with an emphasis on setting where the number of observations is too few compared to the number of covariates. Throughout this text, we use the term *view* define as observations of a random vector on an ordered set of subjects, which is the same for observations of all other random vectors involved in the analysis. We denote each view by $\boldsymbol{X}_i \in \mathbb{R}^{n\times p_i}$, $i = 1, \ldots, m$, where m is the number of random vectors, or equivalently number of views.

In the first two chapters we consider linear association structures shared among multiple views, where the objective is to learn sparse linear combinations of multiple sets of covariates such that they are maximally correlated.

In the first chapter we introduce a new approach to the sparse CCA, where we learn the sparsity pattern of the canonical directions in the first stage by casting this problem as two successively shrinking concave minimization programs which are solved via a first-order algorithm, and in the second stage we solve a small CCA problem by considering the sparsity patterns estimated in the first stage. We demonstrate via simulations that, in comparison to other available methods, our approach demonstrates superior convergence properties and capability to recover the underlying sparsity patterns and the magnitudes of the non-zero elements of the canonical directions, as well as, significantly lower computational cost. We then apply our method to a multi-omic environmental genetics study on fruit flies, where we hypothesise about the mechanism of adaptation of this model organism to environmental pesticides.

In the second chapter we tackle a shared short-coming of sparse PCA and sparse CCA methods, which is that, in case of estimating multiple components or canonical directions

for each view, these directions are not orthogonal to each other, which diminishes interpretability. While all other approaches estimate canonical directions one-by-one via the contraction scheme, we offer a block scheme where we estimate the first d canonical directions simultaneously. In this setting, we can more easily impose orthogonality, and also encourage disjoint sets of non-zero elements within multiple directions, resulting in more interpretable models. We also extended our model to what we call sparse *Directed CCA*, where we use an accessory variable, defined in the text, to try to capture variations related to a certain hypothesis, rather than the dominant variations which might be proven irrelevant to the main hypothesis. As a validating example, we apply our method to the lung cancer multi-omics available on *The Cancer Genome Atlas*, using survival data as our accessory variable. While regular sparse CCA exclusively identified correlation structures dominated by and communities separated by *gender*, our directed sparse CCA correctly identified two underlying communities which were significantly separated by survival.

In the final chapter, we generalize our framework to discover non-linear association structures by proposing a two-stage *sparse kernel CCA* algorithm. We learn maximally aligned kernels in the first stage via sparse *Multiple Kernel Learning (MKL)*, and then solve a KCCA problem in the second stage using learned kernels. We perform sparse MKL by forming an alignment matrix where its elements are the sample *Hilbert Schmidt Independence Criterion* of base kernels of pairs of views. These base kernels are functions of small sets of covariates of each view; therefore our sparse MKL approach provides interpretable solutions, as sparse convex linear combinations of base kernels. We finally provide an `Apache Spark` implementation of our methods introduced throughout the dissertation which makes users capable of running our methods on very high-dimensional datasets, e.g. observations on millions of Single Nucleotide Polymorphism loci, using distributed computing. We call this package `SparKLe`.

`R` versions of our algorithms are also available. `MuLe`, `BLOCCS`, and `SparKLe-R` implements our methods presented in Chapters 1,2, and 3, respectively.

To my family,

That trained me with love!

Contents

Chapter 1

Sparse Canonical Correlation Analysis via Concave Minimization

Abstract

A new approach to the *sparse Canonical Correlation Analysis (sCCA)* is proposed with the aim of discovering interpretable associations in very high-dimensional multi-view, i.e. observations of multiple sets of variables on the same subjects, problems. Inspired by the sparse PCA approach of Journée et al. (2010), we also show that the sparse CCA formulation, while non-convex, is equivalent to a maximization program of a convex objective over a compact set for which we propose a first-order gradient method. This result helps us reduce the search space drastically to the boundaries of the set. Consequently, we propose a two-step algorithm, where we first infer the sparsity pattern of the canonical directions using our fast algorithm, then we shrink each view, i.e. observations of a set of covariates, to contain observations on the sets of covariates selected in the previous step, and compute their canonical directions via any CCA algorithm. We also introduce *Directed Sparse CCA*, which is able to find associations which are aligned with a specified experiment design, and *Multi-View sCCA* which is used to discover associations between multiple sets of covariates. Our simulations establish the superior convergence properties and computational efficiency of our algorithm as well as accuracy in terms of the canonical correlation and its ability to recover the supports of the canonical directions. We study the associations between metabolomics, trasncriptomics and microbiomics in a multi-omic study using `MuLe`, which is an `R` package that implements our approach, in order to form hypotheses on mechanisms of adaptations of *Drosophila Melanogaster* to high doses of environmental toxicants, specifically Atrazine, which is a commonly used chemical fertilizer.

1. Introduction

Canonical Correlation Analysis(CCA), Hotelling (1935) , is a powerful set of approaches for analyzing the relationship between two sets of random vectors, and discovering associations between elements of said vectors. Classical CCA is specifically concerned with finding linear combinations of the elements of each random vector such that they are maximally correlated estimated using observations of each random vector on matching subjects/individuals, i.e. different *views*, of the same latent random vector. In this article, we use the terms *view* and *dataset* interchangeably, denoted by $\boldsymbol{X}_i \in \mathbb{R}^{n \times p_i}$, to refer to n observations of a random vector of length p_i.

CCA has been widely used in various fields of data science and machine learning and has found successful applications in finance, neuro-imaging, computer vision, NLP, social sciences, geography, collaborative filtering, astronomy and a new surge in genomics, especially in recently popular multi-assay genetic/clinical population studies. After its proposition by Hotelling (1935), CCA was first applied in Waugh (1942) where he studied the relationship between the characteristics of wheat and the resulting flour. He demonstrated that desirable wheat is high in texture, density and protein content and low on damaged kernels and foreign materials. Other rather classic applications of CCA include: medical geography, where Monmonier and Finn (1973) showed direct association between the number of hospital beds per capita and physician ratios, socio-medical studies, e.g. Hopkins (1969) studies the relationship between housing and health in Baltimore, education, Dunham and Kravetz (1975) analyzes the association between measures of academic performance in college and exam scores in high school, economics, where Simonson et al. (1983) employs this technique to identify and describe hedging behavior between the asset side and the capital side of the balance sheets of a selection of US. banks, signal processing, e.g. Schell and Gardner (1995) introduces *Programmable CCA* to design filters to distinguish between desired signal and noise, time-series analysis, e.g. Heij and Roorda (1991) employs CCA for state-space modeling, geography, e.g. Ouarda et al. (2001) perform a regional flood frequency analysis using CCA by investigating the correlation structure between watershed characteristics and flood peaks, medical imaging, e.g. Friman et al. (2001) benefited from CCA in detecting activated brain regions based on physiological parameters such as temporal shape and delay of the hemodynamic response. There are plenty of other examples in the fields of chemistry, e.g. Tu et al. (1989), physics, e.g. Wong et al. (1980), dentistry, e.g. Lindsey et al. (1985) where CCA is utilized to discover complex yet meaningful associations between two sets of variables.

CCA and its variants have also found substantial grounds in modern fields of research such as artificial intelligence and statistical learning, neuro-imaging and human perception, context-based content retrieval, collaborative filtering, dimensionality reduction and feature selection, and spatial and temporal genome-wide association studies. Cao et al. (2015) and Nakanishi et al. (2015) used CCA in the area of Brain Computer Interface(BCI) to recognize the frequency components of target stimuli. In the area of image recognition, Hardoon et al. (2004) use a kernel CCA method to perform content-based image retrieval and learn semantics of multimedia content by combining image and text data. Ogura et al. (2013), Shen et al. (2013), and Wang et al. (2013) have employed CCA and its variants for the purpose of feature selection/extraction/fusion and dimensionality reduction.

Modern Canonical Correlation Analysis algorithms have had a significant surge in genomics esp. multi-omic genetic and environmental studies in the last few years mainly due to fast and efficient genome sequencing and measurement technologies becoming more accessible. Such studies typically involve two or more, usually high-dimensional, omic datasets, e.g. trascriptomic, metabolomic, microbiomic data. An instance of such study is Hyman et al. (2002) where they performed CGH analysis on cDNA microarrays in breast cancer and compared copy number and mRNA expression levels to infer the impact of genomic changes on gene expression. Yamanishi et al. (2003) successfully utilized this method to recognize the operons in *Escherichia Coli* genome by comparing three datasets corresponding to functional, locational and expression relationships between the genes. Morley et al. (2004), Pollack et al. (2002), Snijders et al. (2017), Orsini et al. (2018), Fang et al. (2016), Rousu et al. (2013), Seoane et al. (2014), Baur and Bozdag (2015), Sarkar and Chakraborty (2015), and Cichonska et al. (2016) are few other notable relevant works.

In the next section we provide an overview of the common approaches, but we first compile the notation used throughout the paper in the subsection below.

2. Notation

Each view, i.e. the observation matrix on random vector $X_i(\omega) : \Omega \to \mathbb{R}^{p_i}$, is denoted by $\boldsymbol{X}_i \in \mathbb{R}^{n \times p_i}$, $i = 1, \ldots, m$. n is reserved to denote the sample size and p_i to denote the length of each random vector $X_i, i = 1, \ldots, m$. Canonical directions are denoted by $\boldsymbol{z}_i \in \mathcal{B}^{p_i}$, or $\boldsymbol{z}_i \in \mathcal{S}^{p_i}$, and $\boldsymbol{Z}_i \in \mathcal{S}_d^{p_i}$, where $\mathcal{B} = \{\boldsymbol{x} \in \mathbb{R} | \|\boldsymbol{x}\|_2 \leq 1\}$ and $\mathcal{S} = \{\boldsymbol{x} \in \mathbb{R} | \|\boldsymbol{x}\|_2 = 1\}$. $l_x(\boldsymbol{z}) = \|\boldsymbol{z}\|_x : \mathbb{R}^p \to \mathbb{R}$ denotes any norm function, more specifically $l_{0/1}(\boldsymbol{z}) = \|\boldsymbol{z}\|_{0/1}$, and $\boldsymbol{\tau}^{(i)}$ refers to the $i-th$ non-zero element of the vector which is specifically used for the sparsity pattern vector. Sample covariance matrices corresponding to the i-th and j-th views is denoted by $\boldsymbol{C}_{ij}$. We drop the subscript when we only have two views. $max(x, 0)$ is also denoted by $[x]_+$. We also coin the term *accessory variables* in Section 5.2 to refer to the variables towards which we direct estimated canonical directions, disregarding their causal roles as covariates or dependent variables. We also use “program” to refer to “optimization programs”.

3. An Overview of Approaches to the CCA Problem

This subsection covers a literature review of Canonical Correlation Analysis, common approaches, and their statistical assumptions and approximations. While linear approaches and especially their regularized extensions are the main focus of this paper, we have also provided an overview of non-linear approaches, e.g. kernelized model of Lai and Fyfe (2000) and DeepCCA of Andrew et al. (2013).

3.1 CCA

Let $X(\omega) : \Omega \to \mathbb{R}^p$ be a random vector with covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$. Further assume that $\mathbb{E}X = \boldsymbol{0}$. Now partition X into $X_1 \in \mathbb{R}^{p_1}$ and $X_2 \in \mathbb{R}^{p_2}$. The covariance matrix can be partitioned accordingly.

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma_{11}} & \boldsymbol{\Sigma_{12}} \\ \boldsymbol{\Sigma_{21}} & \boldsymbol{\Sigma_{22}} \end{bmatrix} \tag{1}$$

Canonical Correlation Analysis, Hotelling (1935), identifies two weight vectors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ such that the Pearson correlation coefficient between the images $X_1\boldsymbol{z_1}$ and $X_2\boldsymbol{z_2}$ is maximized,

$$\begin{aligned}\rho(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*) &= \max_{\boldsymbol{z}_1 \in \mathbb{R}^{p_1}, \boldsymbol{z}_2 \in \mathbb{R}^{p_2}} \frac{\mathbb{E}[(X_1\boldsymbol{z_1})^\top X_2\boldsymbol{z_2}]}{\mathbb{E}[(X_1\boldsymbol{z_1})^2]^{1/2}\mathbb{E}[(X_2\boldsymbol{z_2})^2]^{1/2}} \\ &= \max_{\boldsymbol{z}_1 \in \mathbb{R}^{p_1}, \boldsymbol{z}_2 \in \mathbb{R}^{p_2}} \frac{\boldsymbol{z}_1^\top \boldsymbol{\Sigma}_{12}\boldsymbol{z}_2}{\sqrt{\boldsymbol{z}_1^\top \boldsymbol{\Sigma}_{11}\boldsymbol{z}_1}\sqrt{\boldsymbol{z}_2^\top \boldsymbol{\Sigma}_{22}\boldsymbol{z}_2}} \\ &= \max_{\substack{\boldsymbol{z}_1 \in \mathbb{R}^{p_1}, \boldsymbol{z}_2 \in \mathbb{R}^{p_2} \\ \boldsymbol{z}_1^T \boldsymbol{\Sigma}_{11}\boldsymbol{z}_1 = 1 \\ \boldsymbol{z}_2^T \boldsymbol{\Sigma}_{22}\boldsymbol{z}_2 = 1}} \boldsymbol{z}_1^T \boldsymbol{\Sigma}_{12}\boldsymbol{z}_2\end{aligned} \tag{2}$$

where the last line is due to scale-invariability of ρ.

The images $X_1\boldsymbol{z}_1$ and $X_2\boldsymbol{z}_2$ are called the *canonical variables* and the weights $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are the *canonical loading vectors* or the *canonical directions*. The loading vectors $(\boldsymbol{z}_1^{(1)}, \boldsymbol{z}_2^{(1)})$ obtained from optimizing Program 2 reveal the first canonical correlation. $(\boldsymbol{z}_1^{(2)}, \boldsymbol{z}_2^{(2)})$ that maximize 2 but with an added constraint that their corresponding images are respectively orthogonal to the first pair determine the second canonical correlation. This procedure is continued until no more pairs are found. The number $r \leq min\{p_1, p_2\}$ of pairs of canonical variables can be interpreted as the number of patterns in the correlation structure.

We estimate the population parameters by plugging in sample estimates of the expectations in Program 2. With $\boldsymbol{X}_1 \in \mathbb{R}^{n\times p_1}$ and $\boldsymbol{X}_2 \in \mathbb{R}^{n\times p_2}$ being the sample matrices corresponding to X_1 and X_2 respectively, $\boldsymbol{\Sigma}_{ij}, i, j \in \{1, 2\}$ is estimated by the sample covariance matrices $\boldsymbol{C}_{ij} = \frac{1}{n}\boldsymbol{X}_i^\top \boldsymbol{X}_j, i, j \in \{1, 2\}$.

Therefore the sample CCA optimization problem may be written as,

$$\max_{\substack{\boldsymbol{z}_1 \in \mathbb{R}^{p_1}, \boldsymbol{z}_2 \in \mathbb{R}^{p_2} \\ \boldsymbol{z}_1^\top \boldsymbol{C}_{11}\boldsymbol{z}_1 = 1 \\ \boldsymbol{z}_2^\top \boldsymbol{C}_{22}\boldsymbol{z}_2 = 1}} \boldsymbol{z}_1^\top \boldsymbol{C}_{12}\boldsymbol{z}_2 \tag{3}$$

Generally, this optimization problem is solved using one of the three classes of techniques. Hotelling (1935) solves this problem using Lagrange multipliers to obtain the characteristic equation which is a *standard eigenvalue problem*,

$$\boldsymbol{C}_{22}^{-1}\boldsymbol{C}_{21}\boldsymbol{C}_{11}^{-1}\boldsymbol{C}_{12}^{-1}\boldsymbol{z}_2 = \rho^2\boldsymbol{z}_2 \tag{4}$$

Bach and Jordan (2002) and Hardoon et al. (2004) form the following system of equations using the same Lagrange multiplier technique,

$$\begin{pmatrix} 0 & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & 0 \end{pmatrix}\begin{pmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{pmatrix} = \rho\begin{pmatrix} \boldsymbol{C}_{11} & 0 \\ 0 & \boldsymbol{C}_{22} \end{pmatrix}\begin{pmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{pmatrix} \tag{5}$$

Which can be regarded as a *generalized eigenvalue problem* and the positive generalized eigenvalues as the squared canonical correlations.

Healy (1957) and Ewerbring and Luk (1989) used *singular value decomposition* to find canonical correlations. In this approach, inverse square roots of the sample covariance matrices $\boldsymbol{C}_{11}^{-1/2}$ and $\boldsymbol{C}_{22}^{-1/2}$ are computed. Canonical loading vectors are computed using the following SVD,

$$\boldsymbol{C}_{11}^{-1/2}\boldsymbol{C}_{12}\boldsymbol{C}_{22}^{-1/2} = \boldsymbol{UDV}^{\top} \tag{6}$$

Where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal matrices and the non-zero elements of the diagonal matrix D correspond to the singular values which are equal to the canonical correlations. $\boldsymbol{z}_1^{(k)}$ and $\boldsymbol{z}_2^{(k)}$ are obtained using $\boldsymbol{C}_{11}^{-1/2}\boldsymbol{U}_{.k}$ and $\boldsymbol{C}_{22}^{-1/2}\boldsymbol{V}_{.k}$ respectively.

3.2 Regularized CCA

Techniques reviewed above are applicable in over-determined systems or low-dimensional regimes. However, in high-dimensional regimes where there are fewer observations than variables, $n \leq max\{p_1, p_2\}$, new approaches are needed to overcome the issues of singular covariance matrices and overfitting as well as lack of identifiability of original parameter. These approaches are also helpful in reducing the estimation variance, providing robustness to outliers, and, of special relevance to this paper, offering more interpretable models.

3.2.1 Ridge Regularization

So called *canonical ridge* was proposed in Vinod (1976) to address the problem of insufficient sample size. Here, the innvertibility of the sample covariance matrices C_{11} and C_{22} is improved by introducing ridge penalties, which comes at the cost of introducing two more hyper-parameters, $c_1, c_2 \geq 0$. Ultimately, the optimization constraints in Program 3 become

$$\begin{aligned} z_1^{\top}(C_{11} + c_1 I)z_1 =& 1 \\ z_2^{\top}(C_{22} + c_2 I)z_2 =& 1 \end{aligned} \tag{7}$$

Any of the three algorithms of Section 3.1 may be modified for solving this problem.

3.2.2 Lasso Regularization

LASSO or L_1 regularized CCA, which is one of the two main foci of this paper, is specifically useful when there are not nearly as many observations as covariates. In such high-dimensional settings ridge-regularized methods, although successfully reducing instability, lack interpretability and overfitting is still an issue. To this end, a school of methods exist which does both variable selection and estimation simultaneously or sequentially through sparsity inducing regularization. Parkhomenko et al. (2007), Parkhomenko et al. (2009) , and Witten and Tibshirani (2009) advise a simple soft-thresholding algorithm to enforce sparsity. They apply *sparse CCA* methods to find meaningful associations between genomic datasets, be it RNA expression datasets, single-loci DNA modifications or regions of loss/gain within the genome. Waaijenborg et al. (2008) incorporates a combination of L_1 and L_2 penalties into the CCA model to identify gene networks that are influenced by multiple genetic changes. Hardoon and Shawe-Taylor (2011) offers a different formulation using convex least squares. In their approach the association between the linear combination of one view and the Gram matrix of the other view is computed. They demonstrate that in cases when the observations are very high-dimensional, their sparse CCA approach outperforms KCCA significantly.

The approaches to the L_1 regularized CCA proposed in the literature referenced above are almost identical, except for that of Hardoon and Shawe-Taylor (2011). Despite small differences, e.g. Waaijenborg et al. (2008) uses elastic net which is a mixture of LASSO and ridge penalties, they all solve a regularized SVD using alternating maximization of slightly different optimization programs. *Penalized Matrix Decomposition(PMD)* algorithm which was first introduced in Witten et al. (2009), then extended in Witten and Tibshirani (2009) estimates the sample covariance matrix $\boldsymbol{C}_{12}$ with closest rank-one matrix in a Frobenius norm sense under some constraints.

$$(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*) = \underset{\substack{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}, \boldsymbol{z}_2 \in \mathcal{B}^{p_2} \\ \|\boldsymbol{z}_1\|_1 \leq c_1, \|\boldsymbol{z}_2\|_1 \leq c_2, \sigma \geq 0}}{\arg\min} \|\boldsymbol{C}_{12} - \sigma \boldsymbol{z}_1 \boldsymbol{z}_2^\top\|_F^2 = \underset{\substack{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}, \boldsymbol{z}_2 \in \mathcal{B}^{p_2} \\ \|\boldsymbol{z}_1\|_1 \leq c_1, \|\boldsymbol{z}_2\|_1 \leq c_2}}{\arg\max} \boldsymbol{z}_1^\top \boldsymbol{C}_{12} \boldsymbol{z}_2 \tag{8}$$

where $c_i \geq 0, i = 1, 2$ are sparsity parameters. The last statement in Program 8 is of course a penalized SVD.

3.2.3 Cardinality Regularization

Most approaches to the sparse CCA problem involve the LASSO regularization which was reviewed in Section 3.2.2. However, few greedy approaches were also developed cardinality or L_0 regularized case.

$$(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*) = \underset{\substack{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}, \boldsymbol{z}_2 \in \mathcal{B}^{p_2} \\ \|\boldsymbol{z}_1\|_0 \leq c_1, \|\boldsymbol{z}_2\|_0 \leq c_2}}{\arg\max} \boldsymbol{z}_1^\top \boldsymbol{C}_{12} \boldsymbol{z}_2 \tag{9}$$

where as before the sparsity parameters are non-negative. Wiesel et al. (2008) develop a greedy algorithm which is based on the sparse PCA approach of d'Aspremont et al. (2008), which we also base our L_0 regularized algorithm on, and demonstrate the effectiveness of their backward greedy algorithm in high-dimensional settings.

3.3 Bayesian CCA

Bayesian approaches to CCA were introduced to increase the robustness of the model in low sample size scenarios and improve the validity of the model by allowing different distributions. Klami et al. (2012) offer a detailed review of Bayesian approaches to CCA, and Bach and Jordan (2005) offer a formalization of this problem within a probabilistic framework. In these models latent variables $U \sim \mathcal{N}(0, I_l)$ where $l \leq min\{p_1, p_2\}$ are assumed to generate the observations $\boldsymbol{x}_1^{(i)} \in \mathbb{R}^{p_1}$ and $\boldsymbol{x}_2^{(i)} \in \mathbb{R}^{p_2}$ through

$$\begin{aligned} X_1 | U &\sim \mathcal{N}(\boldsymbol{S}_1 U + \boldsymbol{\mu}_1, \boldsymbol{\Psi}_1) \\ X_2 | U &\sim \mathcal{N}(\boldsymbol{S}_2 U + \boldsymbol{\mu}_2, \boldsymbol{\Psi}_2) \end{aligned} \tag{10}$$

where $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are transform matrices and $\boldsymbol{\Psi}_1$ and $\boldsymbol{\Psi}_1$ noise covariance matrices. Maximum likelihood estimates of model parameters are used to estimate the posterior expectation of U.

3.4 Non-Linear Transformations

So far, our discussion of CCA and its extensions were constrained to linear transformations of observed random variables. Analyzing non-linear correlation structures, however, requires further innovation. *(Deep) neural networks(DNN)* based CCA and *kernel CCA* are reviewed as the two main schools of methods for uncovering non-linear canonical correlations.

3.4.1 DNN-Based CCA

Lai and Fyfe (1999) used neural networks to find non-linear canonical correlation and detect shift information in a random dot stereogram data. Lai and Fyfe (2000) extends this by adding a non-linearity to their network and also by non-linearly transforming the data to a feature space and then performing linear CCA. Andrew et al. (2013) developed the package `deepCCA`, which will be explained here briefly. In this approach, each dataset, $\boldsymbol{X}_i$, is transformed through multiple layers by applying sigmoid functions on linear transformation of the input to the layer $j = 1, \dots, J$ of network $i = 1, \dots, I$,

$$\boldsymbol{a}_i^j = \sigma(\boldsymbol{Z}_i^j \boldsymbol{x}_i + \boldsymbol{b}_i^j), \quad i = 1, \dots, I, j = 1, \dots, J \tag{11}$$

where σ is a nonlinear sigmoid function and $\boldsymbol{Z}_i^j$ and $\boldsymbol{b}_i^j$ are the weight matrices and bias vectors respectively that need to be learned such that some cost function is minimized. The cost function they defined was the correlation between the output views of all I datasets. Assuming output matrices $\boldsymbol{H}_1 \in \mathbb{R}^{o \times n}$ and $\boldsymbol{H}_2 \in \mathbb{R}^{o \times n}$, define $\boldsymbol{C}_{12} = \frac{1}{n-1}\tilde{\boldsymbol{H}}_1\tilde{\boldsymbol{H}}_2^\top$, $\boldsymbol{C}_{11} = \frac{1}{n-1}\tilde{\boldsymbol{H}}_1\tilde{\boldsymbol{H}}_1^\top + \gamma_1\boldsymbol{I}$ and $\boldsymbol{C}_{22} = \frac{1}{n-1}\tilde{\boldsymbol{H}}_2\tilde{\boldsymbol{H}}_2^\top + \gamma_2\boldsymbol{I}$, where $\tilde{\boldsymbol{H}}_i = \boldsymbol{H}_i - \frac{1}{n}\boldsymbol{H}_i\boldsymbol{1}$ are the centered output matrices. Also define $\boldsymbol{T} = \boldsymbol{C}_{11}^{-1/2}\boldsymbol{C}_{12}\boldsymbol{C}_{22}^{-1/2}$. Then the correlation objective to be maximized can be written as the trace norm of $\boldsymbol{T}$.

$$corr(\boldsymbol{H}_1, \boldsymbol{H}_2) = tr(\boldsymbol{T}^\top\boldsymbol{T})^{1/2} \tag{12}$$

Obviously $H_i = f(\boldsymbol{z}_i^j, b_i^j), j = 1, \dots, J$.

Using *DNN*s for multi-view learning is a very active line of research. Recently, models based on *Variational Auto-Encoders(VAE)* have become popular[Wang et al. (2016)].

3.4.2 Kernel CCA & The Kernel Trick

Kernel methods are more popular for analyzing non-linear associations[Lai and Fyfe (2000)]. This is for the most part due to the vast theoretical literature on kernel methods, mainly from SVM literature, [Gestel et al. (2001); Cai (2013); Blaschko et al. (2008); Hardoon and Shawe-Taylor (2009); Alam et al. (2008)] and part due to the significantly fewer number of parameters to be estimated compared to DNNs[Akaho (2001)]. Melzer et al. (2001) applies non-linear feature extraction to object recognition and compares it to non-linear PCA. Bach and Jordan (2002) uses CCA based methods in kernel Hilbert spaces for *Independent Component Analysis(ICA)* and present efficient computation of their derivatives. Larson et al. (2014) utilizes kernel CCA to discover complex multi-loci disease-inducing SNPs related to ovarian cancer.

Kernelized methods use non-linear mappings,$\phi_1(\boldsymbol{X}_1)$ and $\phi_2(\boldsymbol{X}_2)$, of observations to non-Euclidean spaces, $\mathcal{H}_1$ and $\mathcal{H}_1$, where the measures of similarity between images are no longer linear. The similarity may be captured by a symmetric positive semi-definite kernel, which corresponds to the inner product in Hilbert spaces. In essence, KCCA first transforms the observations into Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ using PSD kernels,

$$k_1(\boldsymbol{x}_{1i}, \boldsymbol{x}_{1j}) = \langle \phi_1(\boldsymbol{x}_{1i}), \phi_1(\boldsymbol{x}_{1j}) \rangle_{\mathcal{H}_1}, \quad k_2(\boldsymbol{x}_{2i}, \boldsymbol{x}_{2j}) = \langle \phi_2(\boldsymbol{x}_{2i}), \phi_2(\boldsymbol{x}_{2j}) \rangle_{\mathcal{H}_2} \tag{13}$$

In practice, we don't need to specify the mappings $\phi_i(\boldsymbol{x}_{i,j})$. *Mercer's theorem*[Mercer (1909)] guarantees that as long as $k_1(\boldsymbol{x}_{ij}, \boldsymbol{x}'_{ij})$ is a positive semi-definite inner-product kernel, there is a corresponding $\phi_i : \mathbb{R}^{p_i} \to \mathcal{H}$ equipped with inner-product $< ., . >_{\mathcal{H}}$. This permits us to bypass evaluating ϕ_i and go straight to evaluating inner-product kernels $k_i, 1, \ldots, I$. The rest of the analysis will be quite similar to the CCA problem except that the observation matrices $\boldsymbol{X}_i$ are replaced by their corresponding Gram matrices K_i for $i = 1, \ldots, I$. For a more comprehensive treatment, refer to Hardoon et al. (2004) and Bach and Jordan (2002).

The remainder of this paper is organized as follows: In Section 4 we introduce the optimization problems corresponding to L_0/L_1*regularized CCA* which are then extended to *Multi-View Sparse CCA* and *Directed Sparse CCA* in Section 5. In Section 6, we propose algorithms that solve the optimization programs of Sections 4 and 5. In Section 7 we apply `MuLe`, the `R`-package that implements our algorithms, to simulated data, where we benchmark our method and also compare it to several other available approaches. We also utilize it in Section 8 to discover and interpret multi-omic associations which explain the mechanisms of adaptations of *Dropsophila Melanogaster* to environmental pesticides. We conclude this paper in Chapter 9. Appendices are referenced in the text wherever applicable.

4. Sparse Canonical Correlation Analysis

We consider *sparse CCA* formulations of the following form,

$$\phi_{l_x,l_x}(\gamma_1, \gamma_2) = \max_{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}} \max_{\boldsymbol{z}_2 \in \mathcal{B}^{p_2}} \boldsymbol{z}_1^T \boldsymbol{C}_{12} \boldsymbol{z}_2 - \gamma_1 l_x(\boldsymbol{z}_1) - \gamma_2 l_x(\boldsymbol{z}_2) \tag{14}$$

where $l_x = l_x(\boldsymbol{z})$ is a sparsity-inducing norm function, $\gamma_i \geq 0$, $i = 1, 2$ are regularization parameters, and $\boldsymbol{C}_{12} = 1/n \boldsymbol{X}_1^\top \boldsymbol{X}_2$ is the sample covariance matrix.

4.1 L_1 Regularization

Consider $x = 1$ in Program 14,

$$\phi_{l_1,l_1}(\gamma_1, \gamma_2) = \max_{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}} \max_{\boldsymbol{z}_2 \in \mathcal{B}^{p_2}} \boldsymbol{z}_1^T \boldsymbol{C}_{12} \boldsymbol{z}_2 - \gamma_1 \|\boldsymbol{z}_1\|_1 - \gamma_2 \|\boldsymbol{z}_2\|_1 \tag{15}$$

This optimization program is equivalent[1] to the one in 8.

Theorem 1 *Maximizers, $(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*)$, of $\phi_{l_1,l_1}(\gamma_1, \gamma_2)$ in Program 15 are given by,*

$$\boldsymbol{z}_1^* = \arg\max_{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}} \sum_{i=1}^{p_2} [|\boldsymbol{c}_i^T \boldsymbol{z}_1| - \gamma_2]_+^2 - \gamma_1 \|\boldsymbol{z}_1\|_1 \tag{16}$$

[1] Optimization programs $\psi_{\boldsymbol{x}}(\boldsymbol{\lambda})$ and $\eta_{\boldsymbol{y}}(\boldsymbol{\mu})$ are called *equivalent* if there is a one-to-one mapping $g : \mathcal{D}_{\boldsymbol{\lambda}} \to \mathcal{D}_{\boldsymbol{\mu}}$ such that $\boldsymbol{x}^* = \boldsymbol{y}^*$ if $\boldsymbol{\lambda} = g(\boldsymbol{\mu})$.

and

$$z^*_{2i} = z^*_{2i}(\gamma_2) = \frac{sgn(\boldsymbol{c}_i^T \boldsymbol{z}_1)[|\boldsymbol{c}_i^T \boldsymbol{z}_1| - \gamma_2]_+}{\sqrt{\sum_{k=1}^{p_2}[|\boldsymbol{c}_k^T \boldsymbol{z}_1| - \gamma_2]_+^2}}, \quad i = 1, \ldots, p_2. \tag{17}$$

Proof [2]

$$\begin{aligned}\phi_{l_1,l_1}(\gamma_1,\gamma_2) &= \max_{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}} \max_{\boldsymbol{z}_2 \in \mathcal{B}^{p_2}} \boldsymbol{z}_1^\top \boldsymbol{C}_{12} \boldsymbol{z}_2 - \gamma_1 \|\boldsymbol{z}_1\|_1 - \gamma_2 \|\boldsymbol{z}_2\|_1 \\ &= \max_{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}} \max_{\boldsymbol{z}_2 \in \mathcal{B}^{p_2}} \sum_{i=1}^{p_2} z_{2i}(\boldsymbol{c}_i^\top \boldsymbol{z}_1) - \gamma_2 \|\boldsymbol{z}_2\|_1 - \gamma_1 \|\boldsymbol{z}_1\|_1 \\ &= \max_{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}} \max_{\boldsymbol{z}'_2 \in \mathcal{B}^{p_2}} \sum_{i=1}^{p_2} |z'_{2i}|(|\boldsymbol{c}_i^\top \boldsymbol{z}_1| - \gamma_2) - \gamma_1 \|\boldsymbol{z}_1\|_1\end{aligned} \tag{18}$$

where we used the following change-of-variable $\boldsymbol{z}_{2i} = sgn(\boldsymbol{c}_i^\top \boldsymbol{z}_1)\boldsymbol{z}'_{2i}$. We optimize 18 for $\boldsymbol{z}'_2$ for fixed $\boldsymbol{z}_1$ and change it back to $\boldsymbol{z}_2$ to get the result in Equation 17. Substituting this result back in 18,

$$\phi^2_{l_1,l_1}(\gamma_1,\gamma_2) = \arg\max_{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}} \sum_{i=1}^{p_2} [|\boldsymbol{c}_i^T \boldsymbol{z}_1| - \gamma_2]_+^2 - \gamma_1 \|\boldsymbol{z}_1\|_1 \tag{19}$$

■

The following corollary asserts that we can provide the necessary and sufficient conditions based on the solution $\boldsymbol{z}^*_1$ in order to find the sparsity pattern of $\boldsymbol{z}^*_2$, i.e. $supp(\boldsymbol{z}^*_2)$, denoted in this paper as $\boldsymbol{\tau}_2 \in \{0,1\}^{p_2}$.

Corollary 2 *Given the sparsity parameter γ_2 and maximizer $\boldsymbol{z}^*_1$ of the program 19, entries z^*_{2i}, refer to 17, for which $|\boldsymbol{c}_i^\top \boldsymbol{z}^*_1| \leq \gamma_2$ are identically zero.*

Proof According to Equation 17 of Theorem 1,

$$z^*_{2i} = 0 \Leftrightarrow [|\boldsymbol{c}_i^T \boldsymbol{z}^*_1| - \gamma_2]_+ = 0 \Leftrightarrow |\boldsymbol{c}_i^T \boldsymbol{z}^*_1| \leq \gamma_2 \tag{20}$$

We can go further and show that we can talk about $\boldsymbol{\tau}_2$ without solving for $\boldsymbol{z}^*_1$. Consider Equation 17 once again,

$$|\boldsymbol{c}_i^T \boldsymbol{z}_1| \leq \|\boldsymbol{c}_i\|_2 \|\boldsymbol{z}_1\|_2 = \|\boldsymbol{c}_i\|_2 \tag{21}$$

Hence, $z_{2i} = 0$ for $i \in 1, \ldots, p_2$ if $\|\boldsymbol{c}_i\|_2 \leq \gamma_2$ without regard to $\boldsymbol{z}^*_1$. ■

Program 16 can be viewed as a L_1 regularized maximization of a quadratic function over a compact set. Obviously the objective is not convex, since it's the difference of two convex functions.

[2] We use the technique introduced in Journée et al. (2010) for sparse PCA to carry out the proofs of Theorems 1 and 5

However, as we will elaborate more Chapter 6 where we propose our two-stage algorithm, `MuLe`, we are only interested in $\boldsymbol{z}_1^*$ for the purpose of inferring $\boldsymbol{\tau}_2$. Hence we will optimize Program 19 with no regularization term in the first stage.

$$\phi^2_{l_1,l_1}(\gamma_1,\gamma_2) \approx \max_{\boldsymbol{z}_1\in\mathcal{B}^{p_1}} \sum_{i=1}^{p_2}[|\boldsymbol{c}_i^T\boldsymbol{z}_1| - \gamma_2]_+^2 = \max_{\boldsymbol{z}_1\in\mathcal{S}^{p_1}} \sum_{i=1}^{p_2}[|\boldsymbol{c}_i^T\boldsymbol{z}_1| - \gamma_2]_+^2 \tag{22}$$

Remark 3 *As a result of this approximation, as stated in Program 22, the search space is drastically shrunk from a p_1-dimensional Euclidean ball to a p_1-dimensional sphere. This is as a result of maximizing a convex function over a compact set.*

Remark 4 *Program 22 is a valid approximation of the Program 19. Beside our simulation results in Section 7, we can see that there is a one-to-one mapping $\gamma_1 = h(\gamma_2)$ in light of Equation 20; in other words, for every γ_1 for which $z^*_{1i} = 0$ there is a γ_2 for which the last inequality in 20 is true.*

4.2 L_0 Regularization

Adapting formulation 9 of Wiesel et al. (2008) to our approach is equivalent to setting $x = 0$ in 14,

$$\phi_{l_0,l_0}(\gamma_1,\gamma_2) = \max_{\boldsymbol{z}_1\in\mathcal{B}^{p_1}} \max_{\boldsymbol{z}_2\in\mathcal{B}^{p_2}} \boldsymbol{z}_1^T\boldsymbol{C}_{12}\boldsymbol{z}_2 - \gamma_1\|\boldsymbol{z}_1\|_0 - \gamma_2\|\boldsymbol{z}_2\|_0 \tag{23}$$

However, to make use of the results in the previous section, we consider the following program instead,

$$\phi'_{l_0,l_0}(\gamma_1,\gamma_2) = \max_{\boldsymbol{z}_1\in\mathcal{B}^{p_1}} \max_{\boldsymbol{z}_2\in\mathcal{B}^{p_2}} (\boldsymbol{z}_1^T\boldsymbol{C}_{12}\boldsymbol{z}_2)^2 - \gamma_1\|\boldsymbol{z}_1\|_0 - \gamma_2\|\boldsymbol{z}_2\|_0 \tag{24}$$

Theorem 5 *Maximizers, $(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*)$, to $\phi_{l_0,l_0}(\gamma_1,\gamma_2)$ in Program 23 are given by,*

$$\boldsymbol{z}_1^* = \arg\max_{\boldsymbol{z}_1\in\mathcal{B}^{p_1}} \sum_{i=1}^{p_2}[(\boldsymbol{c}_i^T\boldsymbol{z}_1)^2 - \gamma_2]_+ - \gamma_1\|\boldsymbol{z}_1\|_0 \tag{25}$$

and

$$z^*_{2i} = z^*_{2i}(\gamma_2) = \frac{[sgn((\boldsymbol{c}_i^T\boldsymbol{z}_1)^2 - \gamma_2)]_+\boldsymbol{c}_i^\top\boldsymbol{z}_1}{\sqrt{\sum_{k=1}^{p_2}[sgn((\boldsymbol{c}_k^T\boldsymbol{z}_1)^2 - \gamma_2)]_+(\boldsymbol{c}_k^\top\boldsymbol{z}_1)^2}}, \quad i = 1,\ldots,p_2. \tag{26}$$

Proof Consider optimizing over $\boldsymbol{z}_2$ while keeping $\boldsymbol{z}_1$ fixed. First, assume $\gamma_2 = 0$. Obviously, $\phi_{l_0,l_0}(\gamma_1,0)|_{\boldsymbol{z}_1=const.}$ is maximized at $\boldsymbol{z}_2^* = \boldsymbol{c}_i^\top\boldsymbol{z}_1$. Now, considering the case for $\gamma_2 > 0$, for which $z^*_{2i} = 0$ for any $\boldsymbol{z}_1$ such that $\phi_{l_0,l_0}(\gamma_1,0)|_{\boldsymbol{z}_1=const.} = (\boldsymbol{c}_i^T\boldsymbol{z}_1)^2 \leq \gamma_2$. Considering this analysis and normalizing we obtain Equation 26. Substituting back in 24, we arrive at 25.

■

Similar to the L_1 regularized case, the following corollary formalizes the relationship between $\boldsymbol{z}_1^*$ and the sparsity pattern $\boldsymbol{\tau}_2 \in \{0,1\}^{p_2}$ of $\boldsymbol{z}_2^*$.

Corollary 6 *Given the sparsity parameter γ_2 and solution $\boldsymbol{z}_1^*$ to the program 25,*

$$\boldsymbol{\tau}_{2i} = \begin{cases} 0 & -\sqrt{\gamma_2} \leq \boldsymbol{c}_i^\top \boldsymbol{z}_1^* \leq \sqrt{\gamma_2} \\ 1 & otherwise \end{cases} \tag{27}$$

Proof According to Equation 26 of Theorem 5,

$$z_{2i}^* = 0 \Leftrightarrow sgn((\boldsymbol{c}_i^T \boldsymbol{z}_1^*)^2 - \gamma_2) \leq 0 \Leftrightarrow (\boldsymbol{c}_i^T \boldsymbol{z}_1^*)^2 \leq \gamma_2 \tag{28}$$

Again, even without solving for $\boldsymbol{z}_1^*$ we can show that

$$(\boldsymbol{c}_i^T \boldsymbol{z}_1)^2 \leq \|\boldsymbol{c}_i\|_2^2 \|\boldsymbol{z}_1\|_2^2 = \|\boldsymbol{c}_i\|_2^2 \tag{29}$$

Hence, in light of 26, $z_{2i} = 0$ for $i \in 1, \ldots, p_2$ if $\|\boldsymbol{c}_i\|_2^2 \leq \gamma_2$ without regards to $\boldsymbol{z}_1^*$. ■

As before, Program 25 can be viewed as a L_0 regularized maximization of a quadratic function over a compact set. Also, we are only interested in $\boldsymbol{z}_1^*$ for the purpose of inferring $\boldsymbol{\tau}_2$. Therefore, to be able to use the previous result in shrinking the search domain, we will optimize Program 25 with no regularization in the first stage.

$$\phi'_{l_0,l_0}(\gamma_1, \gamma_2) \approx \max_{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}} \sum_{i=1}^{p_2} [(\boldsymbol{c}_i^\top \boldsymbol{z}_1)^2 - \gamma_2]_+ = \max_{\boldsymbol{z}_1 \in \mathcal{S}^{p_1}} \sum_{i=1}^{p_2} [(\boldsymbol{c}_i^\top \boldsymbol{z}_1)^2 - \gamma_2]_+ \tag{30}$$

The same justifications as presented in Remarks 3 and 4 apply here analogously.

So far we proposed methods to infer the sparsity patterns $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ which can be used to shrink the covariance matrix drastically, as explain in Section 6. Now, efficient CCA algorithms may be used to estimate the active entries of $\boldsymbol{z}_1^*$ and $\boldsymbol{z}_2^*$. Assuming we have estimated the $i-th$ pair of canonical loading vectors, $(\boldsymbol{z}_1, \boldsymbol{z}_2)^{(i)}, i = 1, \ldots, I$, where $I = rank(\boldsymbol{C}_{12}) \leq n$ assuming $n << min\{p_1, p_2\}$, we define the *i-th Residual Covariance Matrix* as,

$$\boldsymbol{C}_{12}^{(i)} = \boldsymbol{C}_{12} - \sum_{k=1}^{i} (\boldsymbol{z}_1^{(k)*\top} \boldsymbol{C}_{12}^{(k-1)} \boldsymbol{z}_2^{(k)*}) \boldsymbol{z}_1^{(k)*} \boldsymbol{z}_2^{(k)*\top} \quad 1 \leq i \leq I \tag{31}$$

The $(i+1)-th$ pair of canonical loading vectors are estimated by the leading canonical loading vectors of $\boldsymbol{C}_{12}^{(i)}$, using any of the previous two methods. Refer to Algorithm 9 in Appendix B.1 for more details.

5. Further Applications and Extensions

In this section we further extend the methods developed in Section 4. In 5.1 we introduce our approach to *Multi-View Sparse CCA*, where more than two views are available. In 5.2 we extend our approach to *Directed Sparse CCA*, where an observed variable, other than the observed views, is available, towards which we direct the canonical directions.

5.1 Multi-View Sparse CCA

So far we limited ourselves to a pair of views in discussing the sub-space learning problem. In this section we extend our approach to learning sub-spaces from multiple views, i.e. when we have multiple groups of observations, $\boldsymbol{X}_i \in \mathbb{R}^{n \times p_i}, i = 1, \ldots, m$ on matching samples. An example of this problem is multi-omic genetic studies where transcriptomic, metabolomic, and microbiomic data are collected from a single group of individuals. Thus, we try to discover the association structures between random vectors X_i by estimating $\boldsymbol{z}_i$ such that $\boldsymbol{X}_i \boldsymbol{z}_i$ are maximally correlated in pairs. Here, we propose a solution to the following optimization program which is equivalent to the one proposed in Witten and Tibshirani (2009),

$$\phi_{l_x}^{M}(\boldsymbol{\Gamma}) = \max_{\substack{\boldsymbol{z}_i \in \mathcal{B}^{p_i} \\ \forall i=1,\ldots,m}} \sum_{r<s=2}^{m} \boldsymbol{z}_r^T \boldsymbol{C}_{rs} \boldsymbol{z}_s - \sum_{s=2}^{m} \sum_{\substack{r=1 \\ r \neq s}}^{s-1} \Gamma_{sr} \|\boldsymbol{z}_s\|_1 \tag{32}$$

where m is the total number of available views, $\boldsymbol{\Gamma} \in \mathbb{R}^{m \times m}$, $\Gamma_{ij} \geq 0$ is a Lagrange multiplier matrix, and $\boldsymbol{C}_{rs} = 1/n \boldsymbol{X}_r^T \boldsymbol{X}_s$ is the sample covariance matrix of the (r, s) pairs of views. Following similar procedure as in 4.1, we analyze the solution to Program 32.

Theorem 7 *The local optima $\boldsymbol{z}_1^*, \ldots, \boldsymbol{z}_m^*$ of the optimization problem 32 is given by,*

$$z_{si}^* = z_{si}^*(\boldsymbol{\Gamma}) = \frac{sgn(\sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^{\top} \boldsymbol{z}_r)[|\sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^{\top} \boldsymbol{z}_r| - \sum_{\substack{r=1 \\ r \neq s}}^{m} \Gamma_{sr}]_+}{\sqrt{\sum_{k=1}^{p_2} [|\sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsk}^{\top} \boldsymbol{z}_r| - \sum_{\substack{r=1 \\ r \neq s}}^{m} \Gamma_{sr}]_+^2}} \tag{33}$$

and for $r = 1, \ldots, m$ and $r \neq s$,

$$\begin{aligned} \boldsymbol{z}_r(\boldsymbol{\Gamma}) = \max_{\substack{\boldsymbol{z}_r \in \mathcal{B}^{p_r} \\ r \neq s, r=1,\ldots,m}} & \sum_{i=1}^{p_s} [|\sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^{\top} \boldsymbol{z}_r| - \sum_{\substack{r=1 \\ r \neq s}}^{m} \Gamma_{sr}]_+^2 + \\ & \sum_{\substack{i<j=2 \\ i,j \neq s}}^{m} \boldsymbol{z}_i^{\top} \boldsymbol{C}_{ij} \boldsymbol{z}_j - \sum_{\substack{i=1 \\ i \neq s}}^{m} \sum_{\substack{j=1 \\ i \neq j}}^{m-1} \Gamma_{ij} \|\boldsymbol{z}_i\|_1 \end{aligned} \tag{34}$$

Proof Here we follow a progression similar to the proof of Theorem 1.

$$
\begin{aligned}
\phi_{l_1}^{m}(\boldsymbol{\Gamma}) &= \max_{\substack{\boldsymbol{z}_r \in \mathcal{B}^{p_r} \\ r \neq s, r=1,\ldots,m}} \max_{\boldsymbol{z}_s \in \mathcal{B}^{p_s}} \sum_{r<s=2}^{m} \boldsymbol{z}_r^{\top} \boldsymbol{C}_{rs} \boldsymbol{z}_s - \sum_{s=1}^{m} \sum_{\substack{r=1 \\ r \neq s}}^{m-1} \Gamma_{sr} \|\boldsymbol{z}_s\|_1 && (35) \\
&= \max_{\substack{\boldsymbol{z}_r \in \mathcal{B}^{p_r} \\ r \neq s, r=1,\ldots,m}} \max_{\boldsymbol{z}_s \in \mathcal{B}^{p_s}} \sum_{i=1}^{p_s} z_{si} \Big(\sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^{\top} \boldsymbol{z}_r \Big) - \sum_{\substack{r=1 \\ r \neq s}}^{m} \Gamma_{sr} \|\boldsymbol{z}_s\|_1 + \\
&\qquad \overbrace{\sum_{\substack{i<j=2 \\ i,j \neq s}}^{m} \boldsymbol{z}_i^{\top} \boldsymbol{C}_{ij} \boldsymbol{z}_j - \sum_{\substack{i=1 \\ i \neq s}}^{m} \sum_{\substack{j=1 \\ i \neq j}}^{i-1} \Gamma_{ij} \|\boldsymbol{z}_i\|_1}^{I} && (36) \\
&= \max_{\substack{\boldsymbol{z}_r \in \mathcal{B}^{p_r} \\ r \neq s, r=1,\ldots,m}} \max_{\boldsymbol{z}_s \in \mathcal{B}^{p_s}} \sum_{i=1}^{p_s} |z'_{si}| \Big(\Big| \sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^{\top} \boldsymbol{z}_r \Big| - \sum_{\substack{r=1 \\ r \neq s}}^{m} \Gamma_{sr} \Big) + I && (37)
\end{aligned}
$$

The last line follows from $z_{si} = sgn(\sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^{\top} \boldsymbol{z}_r) z'_{si}$. $\tilde{\boldsymbol{c}}_{rsi} = \boldsymbol{c}_{rsi}$ if $r < s$, and $\tilde{\boldsymbol{c}}_{rsi} = \boldsymbol{c}_{rsi}^{\top}$ if $r > s$ where $\boldsymbol{c}_{rsi}$ is the ith row of $\boldsymbol{C}_{rs} = 1/n \boldsymbol{X}_r^{T} \boldsymbol{X}_s$. Solving for $\boldsymbol{z}'_s$ and converting back to $\boldsymbol{z}_s$, using the aforementioned change-of-variable and normalizing, we get the local optimum in 33. Substituting back to 37,

$$
\begin{aligned}
\phi_{l_1}^{m2}(\boldsymbol{\Gamma}) = \max_{\substack{\boldsymbol{z}_r \in \mathcal{B}^{p_r} \\ r \neq s, r=1,\ldots,m}} & \sum_{i=1}^{p_s} \Big[\Big| \sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsk}^{\top} \boldsymbol{z}_r \Big| - \sum_{\substack{r=1 \\ r \neq s}}^{m} \Gamma_{sr} \Big]_{+}^{2} + \\
& \sum_{\substack{i<j=2 \\ i,j \neq s}}^{m} \boldsymbol{z}_i^{\top} \boldsymbol{C}_{ij} \boldsymbol{z}_j - \sum_{\substack{i=1 \\ i \neq s}}^{m} \sum_{\substack{j=1 \\ i \neq j}}^{m-1} \Gamma_{ij} \|\boldsymbol{z}_i\|_1
\end{aligned} \qquad (38)
$$

■

As pointed out in Section 4.1, we're only interested in the optimizing 38 in order to find the sparsity pattern $\boldsymbol{\tau}_s \in \{0,1\}^{p_s}$. Per Remark 4, we can make a good approximation by not considering the regularization terms, simplifying the problem to,

$$
\phi_{l_1}^{m2}(\boldsymbol{\Gamma}) = \max_{\substack{\boldsymbol{z}_r \in \mathcal{B}^{p_r} \\ r \neq s, r=1,\ldots,m}} \sum_{i=1}^{p_s} \Big[\Big| \sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^{\top} \boldsymbol{z}_r \Big| - \sum_{\substack{r=1 \\ r \neq s}}^{m} \Gamma_{sr} \Big]_{+}^{2} + \sum_{\substack{i<j=2 \\ i,j \neq s}}^{m} \boldsymbol{z}_i^{\top} \boldsymbol{C}_{ij} \boldsymbol{z}_j \qquad (39)
$$

As before, we can talk about $\boldsymbol{\tau}_s$, by just looking at $\boldsymbol{z}_r^{*}$ for $r = 1, \ldots, m$ and $r \neq s$.

Corollary 8 *For a sparsity parameter matrix* $\mathbf{\Gamma}$ *and the solution,* $\boldsymbol{z}_r^*$ *for* $r = 1, \dots, m$ *and* $r \neq s$*, to the Program [39](#),*

$$\boldsymbol{\tau}_{2i} = \begin{cases} 0 & |\sum_{\substack{r=1 \\ r\neq s}}^m \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_r| \leq \sum_{\substack{r=1 \\ r\neq s}}^m \Gamma_{sr} \\ 1 & \textit{otherwise} \end{cases} \tag{40}$$

Proof Scanning Equation 33,

$$z_{si}^* = 0 \Leftrightarrow [|\sum_{\substack{r=1 \\ r\neq s}}^m \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_r| - \sum_{\substack{r=1 \\ r\neq s}}^m \Gamma_{sr}]_+^2 = 0 \Leftrightarrow |\sum_{\substack{r=1 \\ r\neq s}}^m \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_r| \leq \sum_{\substack{r=1 \\ r\neq s}}^m \Gamma_{sr} \tag{41}$$

Regardless of $\boldsymbol{z}_r^*$ we have,

$$|\sum_{\substack{r=1 \\ r\neq s}}^m \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_r| \leq \sum_{\substack{r=1 \\ r\neq s}}^m \|\tilde{\boldsymbol{c}}_{rsi}\|_2 \|\boldsymbol{z}_r\|_2 = \sum_{\substack{r=1 \\ r\neq s}}^m \|\tilde{\boldsymbol{c}}_{rsi}\|_2 \tag{42}$$

Hence, $\tau_{si} = 0$ for $i \in 1, \dots, p_s$ if $\sum_{\substack{r=1 \\ r\neq s}}^m \|\tilde{\boldsymbol{c}}_{rsi}\|_2 \leq \sum_{\substack{r=1 \\ r\neq s}}^m \Gamma_{sr}$ regardless of $\boldsymbol{z}_r^*$. ■

Computing $\boldsymbol{\tau}_i$ is the first stage of our two-stage multi-modal sCCA approach, for which a fast algorithm is proposed in 6.4 as part of our proposed `MuLe` framework. The second stage of our approach consists of estimating the active elements of $\boldsymbol{z}_i^*$, for which we use two methods, one is to frame the multi-modal CCA problem as a generalized eigenvalue problem as originally proposed in Kettenring (1971), see Appendix B.2, and the other one is a more algorithmic approach of extending SVD via power iterations to multiple views, refer to Appendix B.3.

5.2 Directed Sparse CCA

Consider a setting where in addition to the views $\boldsymbol{X}_i \in \mathbb{R}^{n \times p_i}$, some *accessory variable*[3], $Y(\omega) : \omega \to \mathbb{R}$ $\boldsymbol{y} \in \mathbb{R}^n$, is also observed. We also term the observed accessory variable the *Accessory Direction*, $\boldsymbol{y} \in \mathbb{R}^n$. Having observed $\boldsymbol{y}$, the objective is to find linear combinations of the covariates in each view which are highly correlated with each other and also "associated" with the accessory direction. This is useful in high-dimensional settings where rank-deficient covariance matrices lead to over-fitting, and small sample sizes are not representative of the direction of variance within each population, and particularly useful in hypotheses generation where we're interested in correlation structures associated with a specific experiment design, e.g. association mechanisms corresponding to a certain treatment effect. Here we compare two approaches to this problem,

5.2.1 Two-Step Formulation

Witten and Tibshirani (2009) propose *Sparse Supervised CCA*, where they consider an extra observed outcome. Their approach consists of two sequential steps where the first step, which is completely separate from the second step, involves finding subsets Q_i of each random vector X_i

[3] We coined the term *Accessory Variable* to prevent confusion about the causal role of $\boldsymbol{y}$, and to emphasize that independent from their role, whether dependent or independent variable, we are solely utilizing them as a direction towards which we're directing the canonical directions.

using a conventional variable selection method, e.g. LASSO regression. In the second step, they utilize sparse CCA where the scope of search and estimation of the canonical directions is limited to the subspaces defined by $X_{ij}, j \in Q_i$,

$$\phi_{l_1,l_1}(\gamma_1,\gamma_2) = \max_{\substack{\boldsymbol{z}_1\in\mathcal{B}^{p_1}\\ z_{1j}=0,\forall j\in Q_1}} \max_{\substack{\boldsymbol{z}_2\in\mathcal{B}^{p_2}\\ z_{2j}=0,\forall j\in Q_2}} \boldsymbol{z}_1^T\boldsymbol{C}_{12}\boldsymbol{z}_2 - \gamma_1\|\boldsymbol{z}_1\|_1 - \gamma_2\|\boldsymbol{z}_2\|_1 \tag{43}$$

In Appendix B.4 a simple algorithm to optimize 43 is introduced. This approach, however, has two considerable shortcomings:

1. Although the scopes of canonical directions are limited to the subspace spanned by $\boldsymbol{z}_i \in \mathcal{B}^{p_i}, z_{ij} = 0, \forall j \in Q_i$, the active elements of these directions are estimated to maximize the sCCA criterion. The estimated direction may well not be associated to the outcome vector anymore, which misses the point.

2. Computing Q_i requires some parameter tuning, e.g. sparsity parameters, which is blind to the CCA criterion; as a result, Q_i might exclude covariates which are moderately correlated with $\boldsymbol{y}$ but highly associated with covariates in other views.

To bridge the gap between the two stages, we propose an approach where $\boldsymbol{z}_i$ are estimated in one stage such that the canonical covariates are highly correlated with each other and also associated with the accessory variable.

5.2.2 Single-Stage Formulation

The following optimization problem tends to perform the two stages of variable selection and performing sCCA in one stage simultaneously,

$$\phi^D_{l_1,l_1}(\boldsymbol{\gamma},\boldsymbol{\epsilon}) = \max_{\boldsymbol{z}_1\in\mathcal{B}^{p_1}} \max_{\boldsymbol{z}_2\in\mathcal{B}^{p_2}} \boldsymbol{z}_1^T\boldsymbol{C}_{12}\boldsymbol{z}_2 - \sum_{i=1}^{2}[\epsilon_i\mathcal{L}_i(\boldsymbol{X}_i\boldsymbol{z}_i,\boldsymbol{y}) + \gamma_i\|\boldsymbol{z}_i\|_1] \tag{44}$$

where $\mathcal{L}_i$ is some loss function which directs our canonical directions to be associated with the accessory direction $\boldsymbol{y}$, and $\gamma_i, \epsilon_i \in \mathbb{R}, i = 1, 2$ are non-negative Lagrange multipliers. Here we analyze two scenarios,

a. Let's consider the case where $\boldsymbol{y}$ is another separate explanatory variable. Here, one possible utility function is the dot-product between the canonical covariates and the explanatory variable, i.e. $\mathcal{L}(\boldsymbol{X}_i\boldsymbol{z}_i,\boldsymbol{y}) = -\langle\boldsymbol{X}_i\boldsymbol{z}_i,\boldsymbol{y}\rangle$. Replacing in 44, we have,

$$\phi^D_{l_1,l_1}(\boldsymbol{\gamma},\boldsymbol{\epsilon}) = \max_{\boldsymbol{z}_1\in\mathcal{B}^{p_1}} \max_{\boldsymbol{z}_2\in\mathcal{B}^{p_2}} \boldsymbol{z}_1^T\boldsymbol{C}_{12}\boldsymbol{z}_2 + \sum_{i=1}^{2}[\epsilon_i\boldsymbol{y}^\top\boldsymbol{X}_i\boldsymbol{z}_i - \gamma_i\|\boldsymbol{z}_i\|_1] \tag{45}$$

Theorem 9 *The local optima, $(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*)$, to $\phi^D_{l_1,l_1}(\boldsymbol{\gamma},\boldsymbol{\epsilon})$ in optimization program 45 is given by,*

$$\boldsymbol{z}_1^* = \arg\max_{\boldsymbol{z}_1\in\mathcal{B}^{p_1}} \sum_{i=1}^{p_2}[|\boldsymbol{c}_i^T\boldsymbol{z}_1 + \epsilon_2\boldsymbol{x}_{2i}^\top\boldsymbol{y}| - \gamma_2]_+^2 + \epsilon_1\boldsymbol{y}\boldsymbol{X}_1\boldsymbol{z}_1 - \gamma_1\|\boldsymbol{z}_{\boldsymbol{1}}\|_1 \tag{46}$$

and

$$z^*_{2i} = z^*_{2i}(\gamma_2, \epsilon_2) = \frac{sgn(\boldsymbol{c}_i^T \boldsymbol{z}_1 + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y})[|\boldsymbol{c}_i^T \boldsymbol{z}_1 + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y}| - \gamma_2]_+}{\sqrt{\sum_{k=1}^{p_2} [|\boldsymbol{c}_i^T \boldsymbol{z}_1 + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y}| - \gamma_2]_+^2}}, \quad i = 1, \ldots, p_2. \tag{47}$$

Proof

$$\begin{aligned}
\phi^D_{l_1,l_1}(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) &= \max_{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}} \max_{\boldsymbol{z}_2 \in \mathcal{B}^{p_2}} \boldsymbol{z}_1^T \boldsymbol{C}_{12} \boldsymbol{z}_2 + \sum_{i=1}^{2} [\epsilon_i \boldsymbol{y}^\top \boldsymbol{X}_i \boldsymbol{z}_i - \gamma_i \|\boldsymbol{z}_i\|_1] \\
&= \max_{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}} \max_{\boldsymbol{z}_2 \in \mathcal{B}^{p_2}} \sum_{i=1}^{p_2} z_{2i}(\boldsymbol{c}_i^T \boldsymbol{z}_1 + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y}) - \gamma_2 \|\boldsymbol{z}_2\|_1 + \epsilon_1 \boldsymbol{y} \boldsymbol{X}_1 \boldsymbol{z}_1 - \gamma_1 \|\boldsymbol{z}_1\|_1 \qquad (48) \\
&= \max_{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}} \max_{\boldsymbol{z}_2 \in \mathcal{B}^{p_2}} \sum_{i=1}^{p_2} |z'_{2i}|(|\boldsymbol{c}_i^T \boldsymbol{z}_1 + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y}| - \gamma_2) + \epsilon_1 \boldsymbol{y} \boldsymbol{X}_1 \boldsymbol{z}_1 - \gamma_1 \|\boldsymbol{z}_1\|_1
\end{aligned}$$

As before we used a simple change of variable, $\boldsymbol{z}_{2i} = sgn(\boldsymbol{c}_i^T \boldsymbol{z}_1 + \epsilon_2 \boldsymbol{x}_i^\top \boldsymbol{y})\boldsymbol{z}'_{2i}$. We solve 48 for $\boldsymbol{z}'_2$ for fixed $\boldsymbol{z}_1$ and convert it back, using the aforementioned change-of-variable, to $\boldsymbol{z}_2$ to get the result in Equation 47. Substituting this result back in 48,

$$\phi^D_{l_1,l_1}{}^2(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) = \max_{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}} \sum_{i=1}^{p_2} [|\boldsymbol{c}_i^T \boldsymbol{z}_1 + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y}| - \gamma_2]_+^2 + \epsilon_1 \boldsymbol{y} \boldsymbol{X}_1 \boldsymbol{z}_1 - \gamma_1 \|\boldsymbol{z_1}\|_1 \tag{49}$$

■

Quite similar to our sCCA formulation we can find the sparsity pattern, $\boldsymbol{\tau}_2$ of $\boldsymbol{z}_2^*$ by looking at $\boldsymbol{z}_1^*$.

Corollary 10 *Given hyperparameters* γ_2, ϵ_2, *and* $\boldsymbol{z}_1^*$ *from program 46,* $\tau_{2i} = 0$ *if* $|\boldsymbol{c}_i^T \boldsymbol{z}_1^* + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y}| \leq \gamma_2$.

Proof According to Equation 47 of Theorem 9,

$$z^*_{2i} = 0 \Leftrightarrow [|\boldsymbol{c}_i^T \boldsymbol{z}_1^* + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y}| - \gamma_2]_+ = 0 \Leftrightarrow |\boldsymbol{c}_i^T \boldsymbol{z}_1^* + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y}| \leq \gamma_2 \tag{50}$$

We can go further and show that we can talk about $\boldsymbol{\tau}_2$ without solving for $\boldsymbol{z}_1^*$,

$$|\boldsymbol{c}_i^T \boldsymbol{z}_1 + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y}| \leq \|\boldsymbol{c}_i\|_2 \|\boldsymbol{z}_1\|_2 + \epsilon_2 \|\boldsymbol{x}_{2i}\|_2 \|\boldsymbol{y}\|_2 = \|\boldsymbol{c}_i\|_2 + \epsilon_2 \|\boldsymbol{x}_{2i}\|_2 \tag{51}$$

Hence, $z_{2i} = 0$ for $i \in 1, \ldots, p_2$ if $\|\boldsymbol{c}_i\|_2 + \epsilon_2 \|\boldsymbol{x}_{2i}\|_2 \leq \gamma_2$ regardless of $\boldsymbol{z}_1^*$. ■

b. Let's examine a setting where $\boldsymbol{y}$ is an outcome variable. Here the objective is to ideally find a common low-dimensional subspace in which the projections of $\boldsymbol{X}_i$ are as correlated as possible and also descriptive/predictive of the outcome $\boldsymbol{y}$. Being confined to linear projections, we can choose $\mathcal{L}_i(\boldsymbol{X}_i \boldsymbol{z}_i, \boldsymbol{y}) = \|\boldsymbol{y} - \boldsymbol{X}_i \boldsymbol{z}_i\|_2^2$, i.e. sum of squared errors loss. Rewriting 44 with this choice,

$$\phi^D_{l_1,l_1}(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) = \max_{\boldsymbol{z}_1 \in \mathcal{B}^{p_1}} \max_{\boldsymbol{z}_2 \in \mathcal{B}^{p_2}} \boldsymbol{z}_1^T \boldsymbol{C}_{12} \boldsymbol{z}_2 - \sum_{i=1}^{2} [\epsilon_i \|\boldsymbol{y} - \boldsymbol{X}_i \boldsymbol{z}_i\|_2^2 + \gamma_i \|\boldsymbol{z}_i\|_1] \tag{52}$$

Theorem 11 *The optimization program in 52 is equivalent to the following program,*

$$\phi^D_{l_1,l_1}(\boldsymbol{\gamma},\boldsymbol{\epsilon}) = \max_{\boldsymbol{z}\in\mathcal{B}^p} \boldsymbol{z}^\top\tilde{\boldsymbol{C}}\boldsymbol{z} + 2\boldsymbol{y}^\top\tilde{\boldsymbol{X}}\boldsymbol{z} - \gamma_1\|\boldsymbol{z}_1\|_1 - \gamma_1\|\boldsymbol{z}_2\|_1 \tag{53}$$

where,

$$\tilde{\boldsymbol{z}} = \begin{bmatrix}\boldsymbol{z}_1\\ \boldsymbol{z}_2\end{bmatrix}, \quad \tilde{\boldsymbol{C}} = \begin{bmatrix}\epsilon_1\boldsymbol{C}_{11} & \boldsymbol{C}_{12}\\ \boldsymbol{C}_{12}^\top & \epsilon_2\boldsymbol{C}_{22}\end{bmatrix}, \quad \tilde{\boldsymbol{X}} = \begin{bmatrix}\epsilon_1\boldsymbol{X}_1 & \epsilon_2\boldsymbol{X}_2\end{bmatrix}, \tag{54}$$

and $p = p_1 + p_2$. The solution, $(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*)$, to $\phi^D_{l_1,l_1}(\boldsymbol{\gamma},\boldsymbol{\epsilon})$ in Program 52 is given by,

$$\boldsymbol{v}^* = \arg\max_{\boldsymbol{v}\in\mathcal{B}^p} \sum_{i=1}^{p_2} [|\tilde{\boldsymbol{c}}_i^T\boldsymbol{v} + 2\tilde{\boldsymbol{x}}_i^\top\boldsymbol{y}| - \gamma_1 I_{(i\le p_1)} - \gamma_2 I_{(p_1<i)}]_+^2 \tag{55}$$

and

$$z_i^* = z_i^*(\boldsymbol{\gamma},\boldsymbol{\epsilon}) = \frac{sgn(\tilde{\boldsymbol{c}}_i^\top\boldsymbol{v} + 2\tilde{\boldsymbol{x}}_i^\top\boldsymbol{y})[|\tilde{\boldsymbol{c}}_i^T\boldsymbol{v} + 2\tilde{\boldsymbol{x}}_i^\top\boldsymbol{y}| - \gamma_1 I_{(i\le p_1)} - \gamma_2 I_{(p_1<i)}]_+}{\sqrt{\sum_{k=1}^p [|\tilde{\boldsymbol{c}}_k^T\boldsymbol{v} + 2\tilde{\boldsymbol{x}}_k^\top\boldsymbol{y}| - \gamma_1 I_{(k\le p_1)} - \gamma_2 I_{(p_1<k)}]_+^2}}, \quad i = 1,\ldots,p_2. \tag{56}$$

Proof Let $\boldsymbol{R} = \tilde{\boldsymbol{C}}^{1/2}$.

$$\begin{aligned}
\phi^D_{l_1,l_1}(\boldsymbol{\gamma},\boldsymbol{\epsilon}) &= \max_{\boldsymbol{z}\in\mathcal{B}^p}\max_{\boldsymbol{v}\in\mathcal{B}^p} \boldsymbol{v}^\top\tilde{\boldsymbol{C}}^{1/2}\boldsymbol{z} + 2\boldsymbol{y}^\top\tilde{\boldsymbol{X}}\boldsymbol{z} - \gamma_1\|\boldsymbol{z}_1\|_1 - \gamma_2\|\boldsymbol{z}_2\|_1\\
&= \max_{\boldsymbol{v}\in\mathcal{B}^p}\max_{\boldsymbol{z}\in\mathcal{B}^p} \sum_{i=1}^p z_i(\tilde{\boldsymbol{c}}_i^\top\boldsymbol{v} + 2\tilde{\boldsymbol{x}}_i^\top\boldsymbol{y}) - \gamma_2\|\boldsymbol{z}_2\|_1 - \gamma_1\|\boldsymbol{z}_1\|_1\\
&= \max_{\boldsymbol{v}\in\mathcal{B}^p}\max_{\boldsymbol{z}\in\mathcal{B}^p} \sum_{i=1}^p |z_i'|(|\tilde{\boldsymbol{c}}_i^\top\boldsymbol{v} + 2\tilde{\boldsymbol{x}}_i^\top\boldsymbol{y}| - \gamma_1 I_{(i\le p_1)} - \gamma_2 I_{(p_1<i)})
\end{aligned} \tag{57}$$

where $z_i = sgn(\tilde{\boldsymbol{c}}_i^\top\boldsymbol{v} + 2\tilde{\boldsymbol{x}}_i^\top\boldsymbol{y})z_i'$. We optimize 57 for $\boldsymbol{z}'$ for fixed $\boldsymbol{v}$ and express it in terms of $\boldsymbol{z}$ to get the result in Equation 56. Substituting this result back in 57,

$$\begin{aligned}
{\phi^D_{l_1,l_1}}^2(\boldsymbol{\gamma},\boldsymbol{\epsilon}) &= \max_{\boldsymbol{v}\in\mathcal{B}^p} \sum_{i=1}^p [|\tilde{\boldsymbol{c}}_i^\top\boldsymbol{v} + 2\tilde{\boldsymbol{x}}_i^\top\boldsymbol{y}| - \gamma_1 I_{(i\le p_1)} - \gamma_2 I_{(p_1<i)}]_+^2\\
&= \max_{\boldsymbol{v}\in\mathcal{S}^p} \sum_{i=1}^p [|\tilde{\boldsymbol{c}}_i^\top\boldsymbol{v} + 2\tilde{\boldsymbol{x}}_i^\top\boldsymbol{y}| - \gamma_1 I_{(i\le p_1)} - \gamma_2 I_{(p_1<i)}]_+^2
\end{aligned} \tag{58}$$

The last line follows from the fact that the objective function is convex, and the maximization is over a convex set, therefore the maxima are located on the boundary.

■

Parallel to the Corollary 10, we can find the relationship between the sparsity pattern $\boldsymbol{\tau} \in \mathbb{R}^p$, and $\boldsymbol{v}^*$.

Corollary 12 *Solving 58 for $\boldsymbol{v}^*$ given $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$,*

$$|\tilde{\boldsymbol{c}}_i^\top \boldsymbol{v}^* + 2\tilde{\boldsymbol{x}}_i^\top \boldsymbol{y}| \leq \gamma_1 I_{(i \leq p_1)} + \gamma_2 I_{(p_1 < i)} \Rightarrow \tau_i = 0$$

Proof According to Equation 56 of Theorem 11,

$$\begin{aligned} &|\boldsymbol{c}_i^T \boldsymbol{z}_1^* + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y}| \leq \gamma_1 I_{(i \leq p_1)} + \gamma_2 I_{(p_1 < i)} \\ &\Rightarrow [|\tilde{\boldsymbol{c}}_i^T \boldsymbol{v}^* + 2\tilde{\boldsymbol{x}}_i^\top \boldsymbol{y}| - \gamma_1 I_{(i \leq p_1)} - \gamma_2 I_{(p_1 < i)}]_+ = 0 \\ &\Rightarrow \tau_i^* = 0 \end{aligned} \tag{59}$$

We can go further and show that we can talk about $\boldsymbol{\tau}$ without solving for $\boldsymbol{v}^*$,

$$|\tilde{\boldsymbol{c}}_i^T \boldsymbol{v}^* + 2\tilde{\boldsymbol{x}}_i^\top \boldsymbol{y}| \leq \|\tilde{\boldsymbol{c}}_i\|_2 + 2\|\tilde{\boldsymbol{x}}_i\|_2 \tag{60}$$

Hence,

$$\tau_i = 0 \quad if \quad \|\tilde{\boldsymbol{c}}_i\|_2 + 2\|\tilde{\boldsymbol{x}}_i\|_2 \leq \gamma_1 I_{(i \leq p_1)} + \gamma_2 I_{(p_1 < i)}, \quad for \quad i = 1, \ldots, p. \tag{61}$$

■

So far in Sections 5.1 and5.2, new approaches to Multi-View sCCA and Directed sCCA were introduced. The former was proposed to compute the canonical directions when we have more than two sets of variables, while the latter was proposed to direct the canonical directions towards an accessory direction.

Proposition 13 *The Directed sCCA approach in 5.2.2.a is equivalent to the approach in 5.2.2.b assuming an orthogonal design matrix, i.e. cov($\boldsymbol{X}_i$) = $\boldsymbol{I}_{p_i}$, and both are equivalent to the Multi-View sCCA approach where the inputs are three views $\boldsymbol{X}_1, \boldsymbol{X_2}$ and $\boldsymbol{y}$.*

Proof Assuming an orthogonal design,

$$\min_{\boldsymbol{z} \in \mathcal{B}^p} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{z}\|_2^2 = \min_{\boldsymbol{z} \in \mathcal{B}^p} \boldsymbol{y}^\top \boldsymbol{y} - 2\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{z} + \boldsymbol{z}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{z} = \max_{\boldsymbol{z} \in \mathcal{B}^p} \boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{z} = \max_{\boldsymbol{z} \in \mathcal{B}^p} \langle \boldsymbol{y}, \boldsymbol{X}\boldsymbol{z} \rangle \tag{62}$$

Hence programs 45 and 52 are equivalent. Now considering the multi-view approach for this problem,

$$\begin{aligned} \phi_{l_x}^M(\boldsymbol{\Gamma}) &= \max_{\substack{\boldsymbol{z}_i \in \mathcal{B}^{p_i} \\ \forall i=1,\ldots,3}} \sum_{r<s=2}^{3} \boldsymbol{z}_r^T \boldsymbol{C}_{rs} \boldsymbol{z}_s - \sum_{s=1}^{3} \sum_{\substack{r=1 \\ r \neq s}}^{2} \Gamma_{sr} \|\boldsymbol{z}_s\|_1 \\ &= \max_{\substack{\boldsymbol{z}_i \in \mathcal{B}^{p_i} \\ \forall i=1,2,3}} \boldsymbol{z}_1^T \boldsymbol{X}_1^\top \boldsymbol{X}_2 \boldsymbol{z}_2 + \boldsymbol{z}_1^T \boldsymbol{X}_1^\top \boldsymbol{y} \boldsymbol{z}_3 + \boldsymbol{z}_2^T \boldsymbol{X}_2^\top \boldsymbol{y} \boldsymbol{z}_3 - \sum_{s=1}^{3} \sum_{\substack{r=1 \\ r \neq s}}^{2} \Gamma_{sr} \|\boldsymbol{z}_s\|_1 \\ &= \max_{\substack{\boldsymbol{z}_i \in \mathcal{B}^{p_i} \\ \forall i=1,2}} \boldsymbol{z}_1^T \boldsymbol{X}_1^\top \boldsymbol{X}_2 \boldsymbol{z}_2 + \boldsymbol{z}_1^T \boldsymbol{X}_1^\top \boldsymbol{y} + \boldsymbol{z}_2^T \boldsymbol{X}_2^\top \boldsymbol{y} - \Gamma_{12} \|\boldsymbol{z}_1\|_1 - \Gamma_{21} \|\boldsymbol{z}_2\|_1 \end{aligned} \tag{63}$$

where the last line follows from the fact that $p_3 = 1$, so $\boldsymbol{z}_3^* = 1$. Equation 63 is identical to 45 for $\epsilon_1 = \epsilon_2 = 1$.

■

6. MuLe

In this section we propose algorithms to solve the optimization programs introduced in Sections 4 and 5. We also address the problem of initialization and hyper-parameter tuning. Our proposed algorithms are generally two-stage algorithms; in the first stage we find the sparsity patterns, $\boldsymbol{\tau}_i \in \{0,1\}^{p_i}, \quad i = 1, \dots, m$, of the optimal canonical directions via concave minimization programs introduced before, and in the second stage we shrink the covariance matrices using the sparsity patterns, $[\boldsymbol{C}'_{ij}]_{rs} = [\boldsymbol{C}_{ij}]_{\tau_i^{(r)}\tau_j^{(s)}}$, where $\tau_i^{(r)}$ is the $r-th$ non-zero element of $\boldsymbol{\tau}_i$ or $r-th$ active element of $\boldsymbol{z}_i^*$, and solve the CCA problem using any *Generalized Rayleigh Quotient* maximizer.

Remark 14 *In order to compute $\boldsymbol{\tau}_i$ for $i = 1, \dots, m$, we start by computing $\boldsymbol{\tau}_m$, using which we shrink $\boldsymbol{C}_{im} \quad \forall i \neq m$ to $[\boldsymbol{C}'_{im}]_{rs} = [\boldsymbol{C}_{im}]_{r\tau_m^{(s)}}$. This in turn shrinks the search space on $\boldsymbol{z}_m$ when computing $\boldsymbol{\tau}_i$, $i \neq m$. We perform the same shrinkage sequentially as we move down towards $\boldsymbol{\tau}_1$, shrinking the search space significantly each time. This sequential shrinkage, not only decreases computational cost drastically, it is also very useful in specifically very high-dimensional settings, since as with each shrinkage, we are directing successive solutions away from the normal cones of the preceding one. This might explain superior stability of our algorithm demonstrated in Section 7.*

Collecting from previous sections, the main differentiating characteristic of our approach is that we cast the problem of finding the sparsity patterns of the canonical directions as a maximization of a convex objective over a convex set, which is equivalent to the following *Concave Minimization* problem,

$$\phi^* = \max_{\boldsymbol{z} \in \mathbb{R}^p} f(\boldsymbol{z}) = \min_{\boldsymbol{z} \in \mathbb{R}^p} -f(\boldsymbol{z}) \tag{64}$$

where $f : \mathbb{R}^p \to \mathbb{R}$ is a convex function. Consult Mangasarian (1996) and Benson (1995) for an in-depth treatment of this class of programs. Journée et al. (2010) propose a simple gradient ascent algorithm for this problem, for which they provide step-size convergence results. Considering these results as well as its empirical performance in terms of convergence and small memory foot-ptint, we also decided to use the following first-order method,

```
Algorithm 1: A first-order optimization method.
  Data: $\boldsymbol{z}_0 \in \mathcal{Q}$
  Result: $\boldsymbol{z}^* = \arg\max_{\boldsymbol{z} \in \mathcal{Q}} f(\boldsymbol{z})$
  $k \leftarrow 0$
  while convergence criterion is not met do
      $\boldsymbol{z}_{k+1} \leftarrow \arg\max_{x \in \mathcal{Q}}(f(z_k) + (x - z_k)^T f'(z_k))$
      $k \leftarrow k + 1$
```

What follows in this section, is the application of Algorithm 1 to the programs proposed so far in this paper.

6.1 l_1-Regularized Algorithm

Applying algorithm 1 to the problem in Program 22.

Algorithm 2: `MuLe` algorithm for optimizing Program 22

```
Data: Sample Covariance Matrix $\boldsymbol{C}_{12}$
      $l_1$-penalty parameter $\gamma_2$
      Initial value $\boldsymbol{z}_1 \in \mathcal{S}^{p_1}$
Result: $\boldsymbol{\tau}_2$, optimal sparsity pattern for $\boldsymbol{z}_2^*$
initialization;
while convergence criterion is not met do
    $\boldsymbol{z}_1 \leftarrow \sum_{i=1}^{p_2} [|\boldsymbol{c}_i^\top \boldsymbol{z}_1| - \gamma_2]_+ sgn(\boldsymbol{c}_i^\top \boldsymbol{z}_1)\boldsymbol{c}_i$
    $\boldsymbol{z}_1 \leftarrow \frac{\boldsymbol{z}_1}{\|\boldsymbol{z}_1\|_2}$
Output $\boldsymbol{\tau}_2 \in \{0,1\}^{p_2}$ where $\tau_{2i} = 0$ if $|\boldsymbol{c}_i^\top \boldsymbol{z}_1^*| \leq \gamma_2$ and 1 otherwise.
```

Once the sparsity pattern $\boldsymbol{\tau}_2$ is found, we shrink the covariance matrix to $\boldsymbol{C}'_{12} \in \mathbb{R}^{p_1 \times |\boldsymbol{\tau}_2|}$, as prescribed at the beginning of this section, and apply Algorithm 1 to ${\boldsymbol{C}'_{12}}^\top$ to find $\boldsymbol{\tau}_1$. Now we shrink the sample covariance matrix once more to $\boldsymbol{C}''_{12} \in \mathbb{R}^{|\boldsymbol{\tau}_1| \times |\boldsymbol{\tau}_2|}$. For large enough sparsity parameters, this matrix is no more rank-deficient, and we can use conventional SVD or CCA methods to fill in the active elements of $\boldsymbol{z}_i$, i.e. solve for the leading singular vectors or canonical covariates of this much smaller matrix.

6.2 l_0-Regularized Algorithm

Now, we use Algorithm 1 to optimize Program 30.

Algorithm 3: `MuLe` algorithm for optimizing Program 30

```
Data: Sample Covariance Matrix $\boldsymbol{C}_{12}$
      $l_1$-penalty parameter $\gamma_2$
      Initial value $\boldsymbol{z}_1 \in \mathcal{S}^{p_1}$
Result: $\boldsymbol{\tau}_2$, optimal sparsity pattern for $\boldsymbol{z}_2^*$
initialization;
while convergence criterion is not met do
    $\boldsymbol{z}_1 \leftarrow \sum_{i=1}^{p_2} [(\boldsymbol{c}_i^\top \boldsymbol{z}_1)^2 - \gamma_2]_+ \boldsymbol{c}_i^\top \boldsymbol{z}_1 \boldsymbol{c}_i$
    $\boldsymbol{z}_1 \leftarrow \frac{\boldsymbol{z}_1}{\|\boldsymbol{z}_1\|_2}$
Output $\boldsymbol{\tau}_2 \in \{0,1\}^{p_2}$ where $\tau_{2i} = 0$ if $(\boldsymbol{c}_i^\top \boldsymbol{z}_1^*)^2 \leq \gamma_2$ and 1 otherwise.
```

Similar to 6.1, we perform successive shrinkage and find $\boldsymbol{\tau}_1$ in the nest step by applying Algorithm 3 on the shrunk matrix ${\boldsymbol{C}'_{12}}^\top$.

6.3 Algorithm Complexity

Perhaps the most appealing characteristic of our proposed algorithm is its significantly lower time complexity compared to other state of the art algorithms. Here we analyze the time complexity of `MuLe` and compare it to the most common algorithm for *sCCA* which is the alternating first order optimization, e.g. Waaijenborg et al. (2008), Parkhomenko et al. (2009), Witten and Tibshirani (2009), for which we use the umbrella term `sSVD` here. Following the set-up thus far, assume we have observed $\boldsymbol{X}_1 \in \mathbb{R}^{n\times p_1}$ and $\boldsymbol{X}_2 \in \mathbb{R}^{n\times p_2}$ and we wish to recover sparse canonical loading vectors $\boldsymbol{z}_1 \in \mathbb{R}^{p_1}$ and $\boldsymbol{z}_2 \in \mathbb{R}^{p_2}$. In order to create more intuition about the speed-up consider a hypothetical algorithm which uses power method to solve a SVD problem and finally simply uses hard-thresholding to create sparse loading vectors. We will call this algorithm `pSVDht`. Also consider another hypothetical algorithm called `sSVDht` which performs the alternating maximization and similarly induces sparsity by hard-thresholding.

Proposition 15 *Time complexity of each iteration of* `MuLe` *is smaller than that of pSVDht if* $n < min\{p_1, p_2\}$ *and* $p_1 \sim p_2$.

Proof. The proof of Proposition 15 is presented in Appendix A.1.

Proposition 16 *The time complexity of each* (z_1, z_2) *update of the* `MuLe` *algorithm, i.e. Algorithm 2, is significantly lower than that of the* `sSVD` *algorithm, Witten and Tibshirani (2009) Algorithm 3.*

Proof. A simple proof is provided in Appendix A.2.

6.4 Sparse Multi-View CCA Algorithm

Our sparse multi-view formulation offered in Program 39 scales linearly with the number of views, which along with the immense shrinkage of the search domain as a result of our concave minimization program results in considerable reduction in convergence time. Below is our proposed gradient ascent algorithm for finding $\boldsymbol{\tau}_i \in \{1,2\}^{p_i}$, $i = 1, \dots, m$.

Algorithm 4: `MuLe` algorithm for optimizing Program 39

```
Data: Sample Covariance Matrices $\boldsymbol{C}_{rs}$,   $1 \le r < s \le m$
      Sparsity parameter matrix $\boldsymbol{\Gamma} \in [0,1]^{m\times m}$
      Initial values $\boldsymbol{z}_r \in \mathcal{S}^{p_r}$,   $1 \le r \le m$
Result: $\boldsymbol{\tau}_s$, optimal sparsity pattern for $\boldsymbol{z}_s$
initialization;
while convergence criterion is not met do
    for $r = 1, \dots, m$, $r \neq s$ do
        $\boldsymbol{z}_r \leftarrow \sum_{i=1}^{p_s}[|\sum_{r=1, r\neq s}^{m} \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_r| - \sum_{r=1, r\neq s}^{m} \Gamma_{sr}]_+ sgn(\sum_{r=1, r\neq s}^{m} \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_r)\tilde{\boldsymbol{c}}_{rsi} + \sum_{l=1, l\neq r,s}^{m} \tilde{\boldsymbol{C}}_{rl}\boldsymbol{z}_l$
        $\boldsymbol{z}_r \leftarrow \frac{\boldsymbol{z}_r}{\|\boldsymbol{z}_r\|_2}$
Output $\boldsymbol{\tau}_s \in \{0,1\}^{p_s}$, where $\tau_{si} = 0$ if $|\sum_{r=1, r\neq s}^{m} \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_r| \le \sum_{r=1, r\neq s}^{m} \Gamma_{sr}$ and 1 otherwise.
```

Once $\boldsymbol{\tau}_s$ is computed we can use successive shrinkage to shrink $\tilde{\boldsymbol{C}}_{rs}$, $r = 1, \ldots, m$, $r \neq s$, per instructions provided in Remark 14, to $\tilde{\boldsymbol{C}}'_{rs} \in \mathbb{R}^{p_r \times |\boldsymbol{\tau}_s|}$. We compute the rest of the sparsity patterns by repeating Algorithm 4 together with successive shrinkage.

Finally we shrink all covariance matrices to $\boldsymbol{C}''_{rs} \in \mathbb{R}^{|\boldsymbol{\tau}_r| \times |\boldsymbol{\tau}_s|}$ using computed sparsity patterns. The second stage of our algorithm, as before, involves estimating the active elements of $\boldsymbol{z}_i^*$; for which we propose two algorithms, the `mCCA` algorithm, see Appendix B.2, and the `mSVD` algorithm, see Appendix B.3.

6.5 Single Stage Sparse Directed CCA Algorithm

We proposed three approaches in 5.2 for *Directed sCCA* problem; one two-stage, where we first perform variable selection and then perform sCCA on the covariance matrix of the selected variables, and two single-stage methods, where we direct the canonical covariates to align with certain outcome of subspace. For our proposed two-stage algorithm refer to the Appendix B.4. Here we elaborate on our single-stage algorithms, starting with 5.2.2.a, we apply our gradient ascent algorithm to Program 49. Once again we optimize it with no regards to the regularization term in the first stage.

Algorithm 5: `MuLe` algorithm for optimizing Program 49

```
Data: Sample Covariance Matrix $\boldsymbol{C}_{12}$
      $l_1$ regularization parameter $\gamma_2$
      Alignment hyperparameters $(\epsilon_1, \epsilon_2)$
      Initial value $\boldsymbol{z}_1 \in \mathcal{S}^{p_1}$
Result: $\boldsymbol{\tau}_2$, optimal sparsity pattern for $\boldsymbol{z}_2^*$
initialization;
while convergence criterion is not met do
    $\boldsymbol{z}_1 \leftarrow \sum_{i=1}^{p_2} [|\boldsymbol{c}_i^\top \boldsymbol{z}_1 + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y}| - \gamma_2]_+ sgn(\boldsymbol{c}_i^\top \boldsymbol{z}_1 + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y}) \boldsymbol{c}_i + \epsilon_1 \boldsymbol{X}_1^\top \boldsymbol{y}$
    $\boldsymbol{z}_1 \leftarrow \frac{\boldsymbol{z}_1}{\|\boldsymbol{z}_1\|_2}$
Output $\boldsymbol{\tau}_2 \in \{0,1\}^{p_2}$ where $\tau_{2i} = 0$ if $|\boldsymbol{c}_i^T \boldsymbol{z}_1^* + \epsilon_2 \boldsymbol{x}_{2i}^\top \boldsymbol{y}| \leq \gamma_2$ and 1 otherwise.
```

As before, to compute $\boldsymbol{\tau}_1$, we use successive shrinkage, and in the second stage we use conventional SVD or CCA to estimate the active entries. Regarding 5.2.2.b, rather than an algorithm solving Program 58, we propose a simpler Algorithm which is identical to Algorithm 5, except that we $\boldsymbol{X}_i^\top \boldsymbol{y}$ with $\boldsymbol{\beta}_i$ for $i = 1, 2$, similarly $\boldsymbol{x}_{ij}^\top \boldsymbol{y}$ with β_{ij}, which is the vector of coefficient estimates from regressing $\boldsymbol{y}$ on $\boldsymbol{X}_i$.

6.6 Initialization & Hyperparameter Tuning

6.6.1 Initialization

Concerning the initialization, we follow the suggestion of Journée et al. (2010) and choose an initial value $\boldsymbol{z}_{1,init}$ for which our algorithm is guaranteed to yield a sparsity pattern with at least one non-zero element. This initial value is chosen parallel to the column with the largest L_2 norm.

$$\boldsymbol{z}_{1init} = \frac{\boldsymbol{c}_{i^*}}{\|\boldsymbol{c}_{i^*}\|_2}, \quad i^* = \underset{i \in \{1, \ldots, p_1\}}{\arg\max} \|\boldsymbol{c}_i\|_2 \tag{65}$$

Where $\boldsymbol{c}_i$ is the i-th column of $\boldsymbol{C}_{12}$. Similarly, $\boldsymbol{z}_{2init} = \boldsymbol{c}'_{i*}/\|\boldsymbol{c}'_{i*}\|_2$, where $\boldsymbol{c}'_{i*}$ is the column of the transpose of the shrunk covariance matrix.

6.6.2 Hyperparameter Tuning

Algorithms 2-5 involve choosing hyperparameters $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$. Here we propose two algorithm for choosing the optimal sparsity parameters, γ_i; they are easily extendable to tuning alignment parameters ϵ_i. But we first need to choose a performance criteria in order to compare different choices of parameters. Witten and Tibshirani (2009) choose penalty parameters which best estimate entries that were randomly removed from the covariance matrix, while some choose them by comparing the Frobenius norms of the reconstructed covariance matrices subtracted from the original matrix. These choices are effectively imposed due to solving a penalized SVD instead of the sCCA problem. However, since we solve the CCA problem in the second stage of our algorithm, we use the canonical correlation, $\rho_{\gamma_1,\gamma_2}(\boldsymbol{X}_1^\top \boldsymbol{z}_1, \boldsymbol{X}_2^\top \boldsymbol{z}_2)$, as our measure, which serves our objective more properly.

Algorithm 6 performs hyperparameter tuning using the k-fold cross-validation method, which is widely common in sCCA literature.

```
Algorithm 6: Hyperparameter Tuning via k-Fold Cross-Validation
Data: Sample matrices $\boldsymbol{X}_i \in \mathbb{R}^{n\times p_i}$, $i = 1,2$
      Sparsity parameters $\gamma_i$, $i = 1,2$
      Initial values $\boldsymbol{z}_i \in \mathcal{S}^{p_i}$, $i = 1,2$
      Number of folds $K$
Result: $\rho_{CV}(\gamma_1,\gamma_2)$ the average cross-validated canonical correlation
Let $\boldsymbol{X}_{ik}, \boldsymbol{X}_{i/k}, i = 1,2, j = 1,\dots,K$ be the validation and training sets corresponding to
  the $k$-th fold, respectively.
for $k$ = 1, ..., $K$ do
    Compute $(\boldsymbol{z}_1^{*(k)}, \boldsymbol{z}_2^{*(k)})$ on $\boldsymbol{X}_{1/k}, \boldsymbol{X}_{2/k}$ via proposed methods in 6.1 or 6.2 with sparsity
      hyperparameters $(\gamma_1,\gamma_2)$
    $\rho^{(k)}(\gamma_1,\gamma_2) = corr(\boldsymbol{X}_{1k}\boldsymbol{z}_1^{*(k)}, \boldsymbol{X}_{2k}\boldsymbol{z}_2^{*(k)})$
$\rho_{CV}(\gamma_1,\gamma_2) = 1/K \sum_{k=1}^{K} \rho^{(k)}(\gamma_1,\gamma_2)$
```

This approach has a significant shortcoming, specially in high-dimensional settings, though. The issue is that once the sparsity parameter is small enough, the fitted models return high correlation values, close to one, which makes the choice of best parameters inaccurate. To cope with this problem, we propose a second algorithm which performs a permutation test, where the null hypothesis is that the views $\boldsymbol{X}_i$ are independent. In order to reject the null, the canonical correlation computed from the matched samples must be significantly higher than the average canonical correlation computed from the permuted samples. To this end, we propose Algorithm 7. Given a grid of hyperparameters, the tuple which minimizes the p-value is chosen.

Algorithm 6 performs hyperparameter tuning using the k-fold cross-validation method, which is widely common in sCCA literature.

Algorithm 7: Hyperparameter Tuning via Permutation Test

```
Data: Sample matrices $\boldsymbol{X}_i \in \mathbb{R}^{n \times p_i}$, $i = 1, 2$
      Sparsity parameters $\gamma_i$, $i = 1, 2$
      Initial values $\boldsymbol{z}_i \in \mathcal{S}^{p_i}$, $i = 1, 2$
      Number of permutations $P$
Result: $p_{\gamma_1, \gamma_2}$ the evidence against the null hypothesis that the canonical correlation is
        not lower when $X_i$ are independent.
Compute $(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*)$ on $\boldsymbol{X}_1, \boldsymbol{X}_2$ via proposed methods in 6.1 or 6.2 with sparsity
  hyperparameters $(\gamma_1, \gamma_2)$
$\rho(\gamma_1, \gamma_2) = corr(\boldsymbol{X}_1 \boldsymbol{z}_1^*, \boldsymbol{X}_2 \boldsymbol{z}_2^*)$
for $p = 1, \ldots, P$ do
    Let $\boldsymbol{X}_1^{(p)}$ be a row-wise permutation of $\boldsymbol{X}_1$
    Compute $(\boldsymbol{z}_1^{*(p)}, \boldsymbol{z}_2^{*(p)})$ on $\boldsymbol{X}_1^{(p)}, \boldsymbol{X}_2$ via proposed methods in 6.1 or 6.2 with sparsity
      hyperparameters $(\gamma_1, \gamma_2)$
    $\rho_{perm}^{(p)}(\gamma_1, \gamma_2) = corr(\boldsymbol{X}_1^{(p)} \boldsymbol{z}_1^{*(p)}, \boldsymbol{X}_2 \boldsymbol{z}_2^{*(p)})$
$p_{\gamma_1, \gamma_2} = 1/P \sum_{p=1}^{P} I(\rho_{perm}^{(p)} > \rho)$
```

7. Experiments

In this section we compare and evaluate our proposed algorithm `MuLe` along with few other sparse CCA algorithms. To perform an inclusive comparison, we tried to choose representatives from different approaches. As argued in 3.2.2, optimization problems introduced in Witten and Tibshirani (2009), Parkhomenko et al. (2009), Waaijenborg et al. (2008) are equivalent. The methods used here for comparison are the *Penalized Matrix Decomposition* proposed in Witten and Tibshirani (2009) which is implemented in the `PMA` package, and also a ridge regularized CCA, noted here as `RCCA`. In order to benchmark `MuLe` comprehensively, simple `SVD` and `SVDthr`, which is simply soft-thresholded SVD, are also included. Note that as mentioned before almost all sparse CCA algorithms try to solve a penalized singular value decomposition problem, whereas we solve a CCA problem in the second stage. In 7.1 and 7.2 we first establish the accuracy of our algorithm, then we compare compute and compare few characteristic curves regarding stability of our algorithm. We also compare out Multi-View Sparse CCA algorithm with other popular algorithm, the results of which is included in Appendix C.1.

7.1 A Rank-One Sparse CCA Model

Consider a CCA problem where $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are generated using the following rank-one model,

$$\boldsymbol{X}_1 = (\boldsymbol{z}_1 + \boldsymbol{\epsilon}_1)\boldsymbol{u}^\top, \quad \boldsymbol{X}_2 = (\boldsymbol{z}_2 + \boldsymbol{\epsilon}_2)\boldsymbol{u}^\top \tag{66}$$

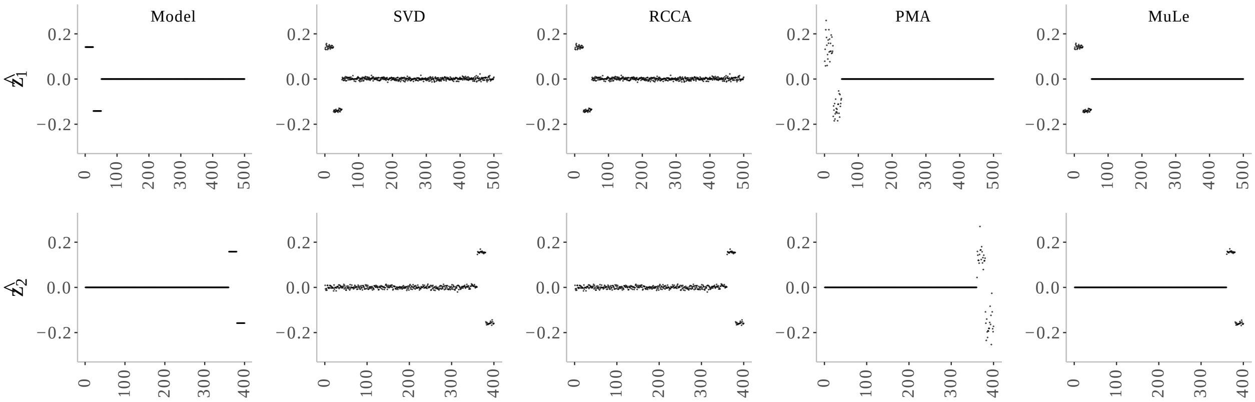

Figure 1: Comparing performance of different sCCA approaches in recovering the sparsity pattern and estimating active elements of the canonical directions. The *Model* or "true" canonical directions are plotted in the leftmost plot.

where $\boldsymbol{z}_1 \in \mathbb{R}^{500}$ and $\boldsymbol{z}_2 \in \mathbb{R}^{400}$ have the following sparsity patterns,

$$\begin{aligned} \boldsymbol{z}_1 &= \left[\underbrace{1,\ldots,1}_{25} \quad \underbrace{-1,\ldots,-1}_{25} \quad \underbrace{0,\ldots,0}_{450} \right] \\ \boldsymbol{z}_2 &= \left[\underbrace{1,\ldots,1}_{25} \quad \underbrace{-1,\ldots,-1}_{25} \quad \underbrace{0,\ldots,0}_{350} \right] \end{aligned} \tag{67}$$

$\boldsymbol{\epsilon}_1 \in \mathbb{R}^{400}$ and $\boldsymbol{\epsilon}_2 \in \mathbb{R}^{500}$ are added Gaussian noise.

$$\begin{aligned} \boldsymbol{\epsilon}_1 &\sim \mathcal{N}(0,\sigma^2), \forall i = 1,\ldots 500, \\ \boldsymbol{\epsilon}_2 &\sim \mathcal{N}(0,\sigma^2), \forall i = 1,\ldots 400, \end{aligned} \tag{68}$$

and

$$\boldsymbol{u}_i \sim \mathcal{N}(0,1), \forall i = 1,\ldots,50. \tag{69}$$

Figure 1 compares MuLe's performance to the methods mentioned above. The noise amplitude, σ was set to 0.2, in order to more significantly differentiate between the methods. It is evident that MuLe successfully identified the underlying sparse model since both the sparsity pattern and the value of the coefficients were estimated quite accurately, while PMA failed to estimate the coefficient sizes accurately. Note here that, our simple cross-validation parameter tuning resulted in accurate identification of the canonical directions while using the same procedure on PMA resulted in cardinalities far from the specified model. Hence, the sparsity parameters for the latter method were chosen by trial-and-error to match model's sparsity pattern.

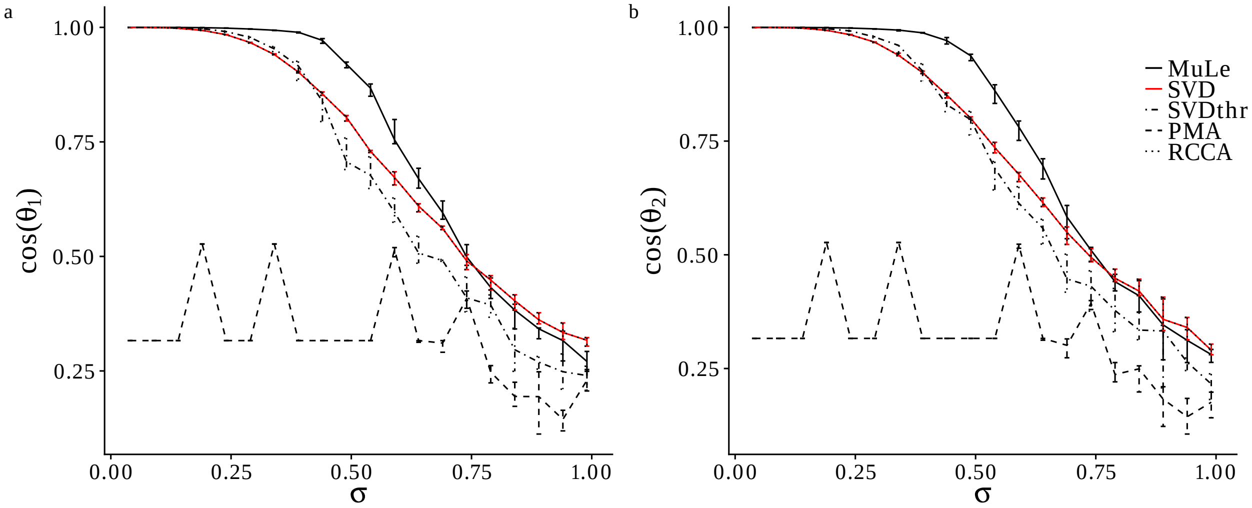

Figure 2: The cosine of the angle between the estimated and true canonical directions, $cos(\theta_i) = |\langle \hat{\boldsymbol{z}}_i, \boldsymbol{z}_i \rangle|$ computed for both datasets.

Under the same setting, but varying level of noise σ, we compute the cosine of the angle between the estimated, $\hat{\boldsymbol{z}}_i$, and true, $\boldsymbol{z}_i$, canonical directions, $cos(\theta_i) = |\langle \boldsymbol{z}_i, \hat{\boldsymbol{z}}_i \rangle|$ for $i = 1, 2$ via the methods utilized in Figure 1. We plotted the results in Figure 2 for both canonical directions; according to which, MuLe outperforms other methods, especially the alternating method of Witten and Tibshirani (2009), throughout the range of noise amplitude. PMA uniquely shows a lot of volatility in its solution. The built-in parameter tuning also misspecified the correct sparsity parameters, but providing correct hyperparameters manually also did not help much. Actually, our test shows that a simple thresholding algorithm like SVDthr outperforms PMA both in terms of support recovery and direction estimation.

But perhaps the most important piece of information one looks for in high-dimensional multi-view studies is the interpretability of the estimated canonical directions. Therefore, ultimately the decisive criteria in choosing the best approach is determined by how well they uncover the "true" underlying sparsity pattern or simply put, how accurately a model performs variable selection. To this end, variable selection accuracy of each method is plotted against the noise amplitude in Fig. 3 as the fraction of the support of $\boldsymbol{z}_i$, $i \in \{1, 2\}$ discovered, here denoted as η_i, vs. the noise amplitude, σ. As before MuLe performs significantly better than other methods throughout the noise amplitude range.

7.2 Solution Stability on Data Without Underlying Sparse CCA Model

In the following simulations, $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are generated by sampling from $\mathcal{N}(\mathbf{0}_{p_i}, \boldsymbol{I}_{p_i}), i \in \{1, 2\}$. The main purpose of this section is to demonstrate the stability of the solution paths while comparing the quality of the solutions of different algorithms as a function of the cardinality of the canonical loadings. The motivation behind this simulation is that the solution of a stable algorithm must grow more similar to the non-sparse CCA solution. Therefore, while setting the sparsity parameter equal

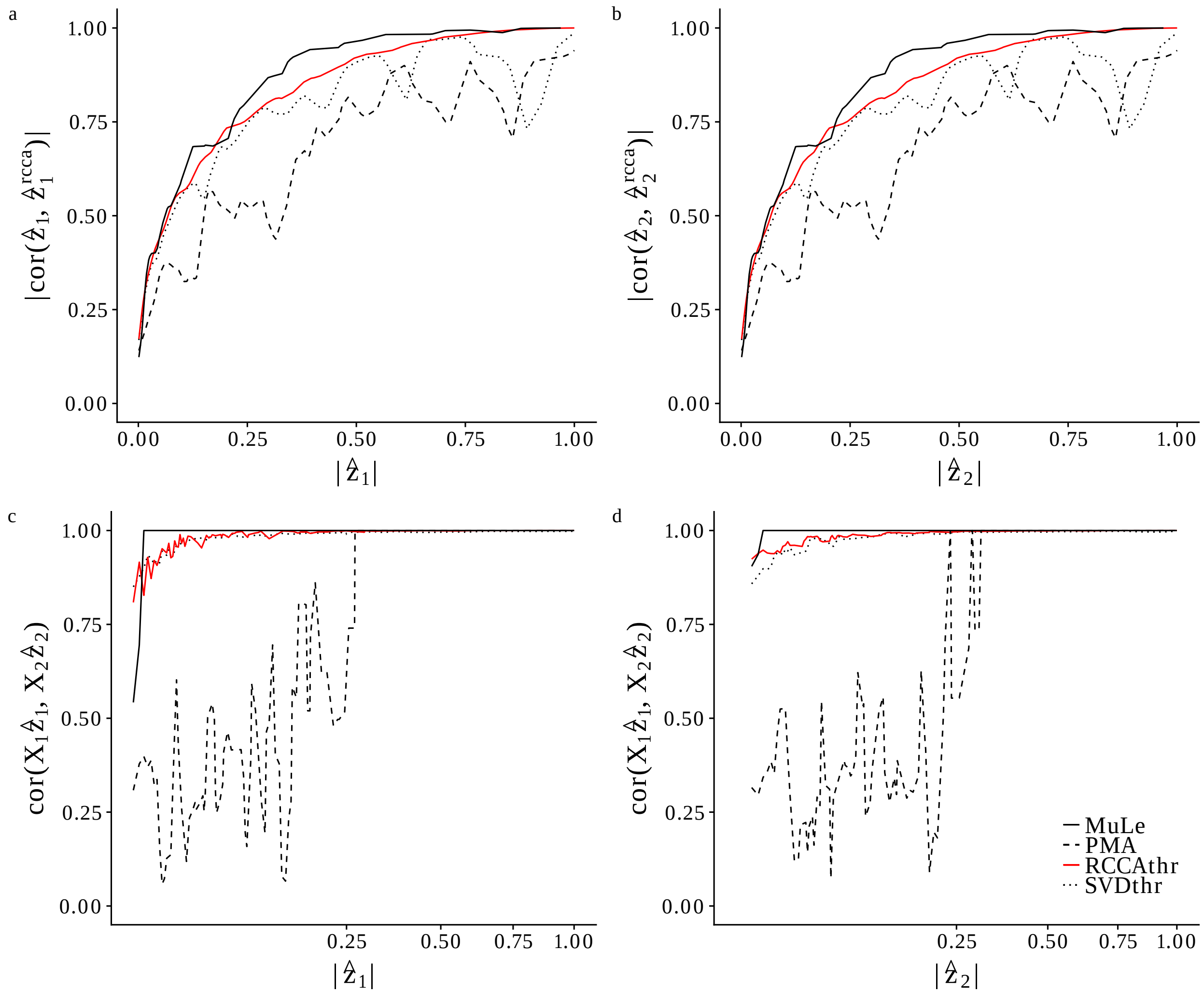

Figure 3: The correlation between the estimated sparse canonical direction and the direction obtained from CCA. (a,b) and the estimated canonical correlation as a function of the cardinality of the estimated direction. (c,d)

to zero for one canonical direction, for an array of sparsity parameters we compute the correlation of the estimated direction with the corresponding direction from the CCA solution, as well as the estimated canonical correlation for the same setting.

The results of the aforementioned simulation is presented in Figure 3. According to our results `MuLe` is consistently more correlated with the CCA solution and for $(\gamma_1, \gamma_2) = (0, 0)$, it solves the CCA problem whereas `PMA` by far does not show the same solution stability. Were columns of $\boldsymbol{X}_i$ more correlated, `PMA` and `SVDThr` would have resulted in even worse solutions.

In the next section we utilize `MuLe` to discover correlation structures in a genomic setting.

8. Fruitfly Pesticide Exposure Multi-Omics

One of the drivers for the development of our method was the rise of multi-omics analysis in functional genomics, pharmacology, toxicology, and a host of related disciplines. Briefly, multiple "omic" modalities, such as transcriptomics, metabolomics, metagenomics, and many other possibilities, are executed on matched (or otherwise related) samples. An increasingly common use in toxicology is the use of transcriptomics and metabolomics to identify, in a single experiment, the genetic and metabolic networks that drive resilience or susceptibility to exposure to a compound[Campos and Colbourne (2018)]. We analyzed recently generated transcriptomics, metabolomics, and 16S DNA metabarcoding data generated on isogenic Drosophila Melanogaster (described in Brown et al. 2019, in preparation). In this experiment, fruit flies are separated into treatment and control groups, where treated animals are exposed to the herbicide *Atrazine*, one of the most common pollutants in US drinking water. Dosage was calculated as 10 times the maximum allowable concentration in US drinking water – a level frequently achieved in surface waters (streams and rivers) and rural wells.

Data was collected after 72 hours, and little to no lethality was observed. Specifically, male and female exposed flies were collected, whereafter mRNA, small molecular metabolites, and 16S rDNA (via fecal collection and PCR amplification of the V3/V4 region) was collected. RNA-seq and 16S libraries were sequenced on an *Illumina MiSeq*, and polar and non-polar metabolites were assayed by direct injection tandem mass spectrometry on a *Thermo Fisher Orbitrap Q Exactive*. Here, we compare 16S, rDNA and metabolites using `MuLe`, to identify small molecules associated with microbial communities in the fly gut microbiome.

This is an intriguing question, as understanding how herbicide exposure remodels the gut microbiome, and, in turn, how this remodeling alters the metabolic landscape to which the host is ultimately exposed is a foundational challenge in toxicology. All dietary co-lateral exposures are "filtered through the lens" of the gut microbiome – compounds that are rapidly metabolized by either the host system or the gut are experienced, effectively, at lower concentrations; the microbiome plays an important role in toxicodynamics.

We utilized the multi-view sparse CCA module of `MuLe` to find three-way associations in our study. Hyper-parameter was performed using our permutation test of Algorithm 7 modified to lean towards more sparse models. Our analysis, see Figures 4 and 5, revealed three principle axes of variation. The first groups host genes for primary and secondary metabolism, cell proliferation, and reproduction along with host metabolites related to antioxidant response. Intriguingly, all metabolites in this axis of variation derive from the linoleic acid pathway, part of the anti-oxidant defense system, which is known to be engaged in response to Atrazine exposure [Sengupta et al. (2015)]. Similarly, *Glutathione S transferase D1 (GstD1)*, a host gene that varies along this axis, is a secondary metabolic enzyme that leverages *glutathione* to neutralize reactive oxygen species (*eletrophilic* substrates). *Linoleic acid* metabolites are known to strongly induce *glutathione* synthesis [Arab et al. (2006)]. The primary metabolism gene, *Cyp6w1* is strongly up-regulated in response to atrazine [Sieber and Thummel (2009)], and here we see it is also tightly correlated with the anti-oxidant defense system. We see broad inclusion of cell proliferation genes *(CG6770, CG16817, betaTub56D)* and genes involved in reproduction (the *Chorion* proteins, major structural components of the eggshell chorion, *Cp15, Cp16, Cp18, Cp19, Cp38*, and *Vitelline* membrane *26Aa (Vm26Aa)*), and it is well known that flies undergo systematic repression of the reproductive system during exposure to environmental stress [Brown et al. (2014)]. Whether this reproductive signal is directly associated with linoleic acid metabolism and glutathione production is an intriguing question for future study.

The second principle axis of variation groups a dominant microbial clade *(Lactobacillales)* along with a collection of host metabolites, and one gene of unknown function. The host metabolites fall principally on the *phosphorylcholine* metabolic pathway, which is known to be induced in a sex-specific fashion in response to atrazine in mammals, but, as far as we know, not previously reported in arthopods [Holásková et al. (2019)] – which may be useful, as it expands the domain of mammalian adverse outcome pathways that can be modeled in Drosophila Melanogaster.

The third and final principle axis includes two host genes – a *cytochrome P450 (Cyp4g1)* known to be involved in atrazine detoxification [Sieber and Thummel (2009)], and a peptidase of unknown function (*CG12374*) – a minority microbial clade (*Rhodospirillales*, [Chandler et al. (2011)]), and another collection of linoleic acid pathway metabolites, along with *1-Oleoylglycerophosphoinositol*, a host metabolite derived from oleic acid. While the ostensible lack of known microbial metabolites is somewhat disappointing, it may also be that these were simply not assigned chemical IDs during the metabolite identification – a common challenge with untargeted chemistry.

In order to verify that the primary effect captured in our canonical directions are co-variations associated with the treatment effect, and not that of sex, exposure length etc., we also projected our samples on to the plane of the first two canonical covariates, see Figure 6. We then color-coded the samples according to the `treatment` vector. We observed that our estimated canonical covariates clearly separate our samples according to the treatment effect.

Overall, we see many of the genes and metabolites involved in response to Atrazine identified in the support of the first and second canonical covariates. The fact that many members of individual pathways were returned together is comforting – genes and metabolites in the same or related pathways should co-vary, and they appear to through the lens of our analysis. The novelty and discovery of the sCCA method lies in identifying potential interactions between these pathways – and the current analysis has yielded a number of hypotheses for follow-up studies, including the coupling of germ cell proliferation repression to Linoleic acid metabolism. The identification of genes of unknown function is also interesting – we posit that *MRE16* along the second principle axis of variation encodes at least one small functional peptide (e.g. a peptidase or an immunopeptide), and this too will be the subject of future study.

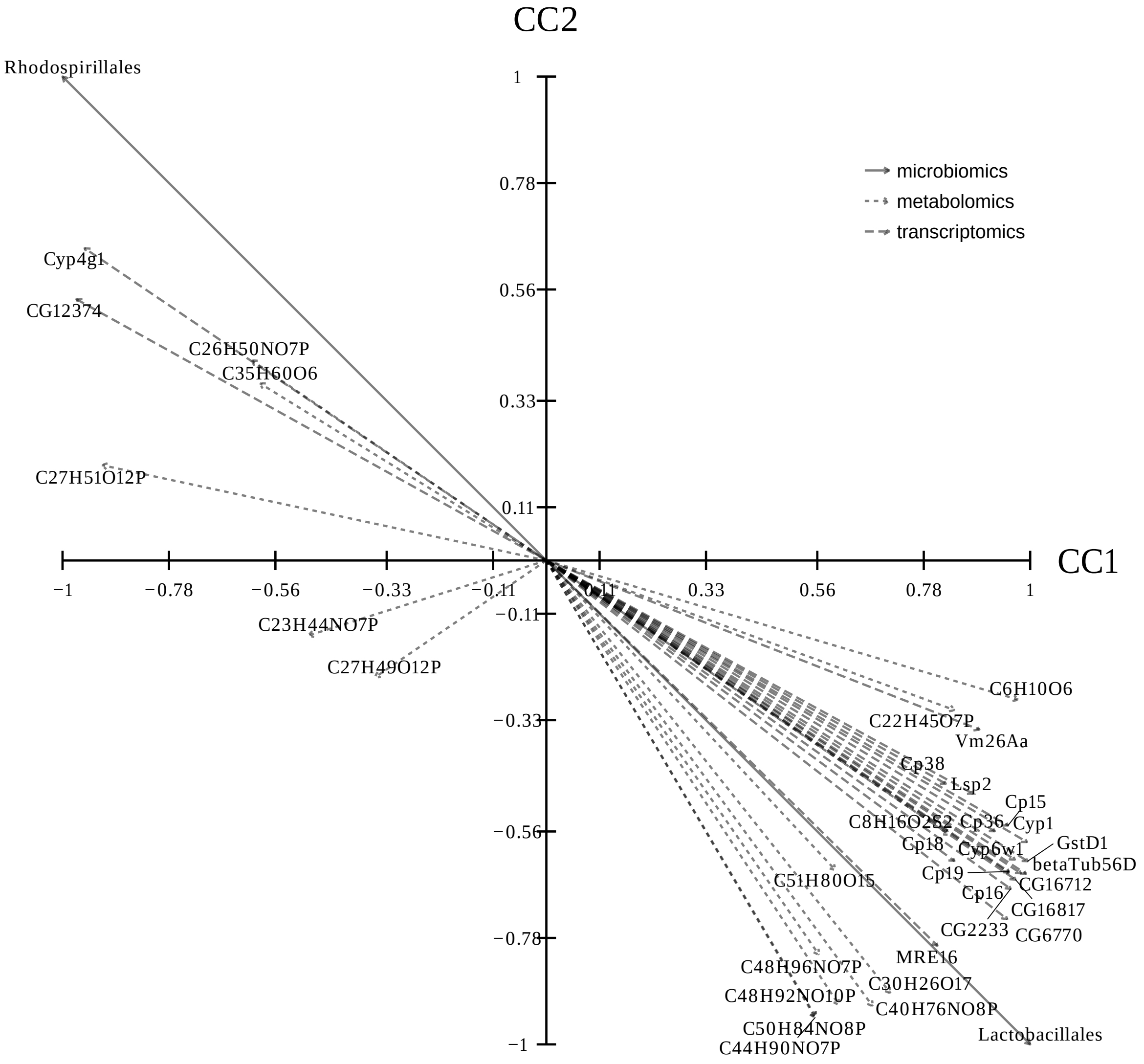

Figure 4: CCA biplot of transcriptomic, microbiomic, and metabolomic datasets in Drosophila Atrazine exposure experiment.

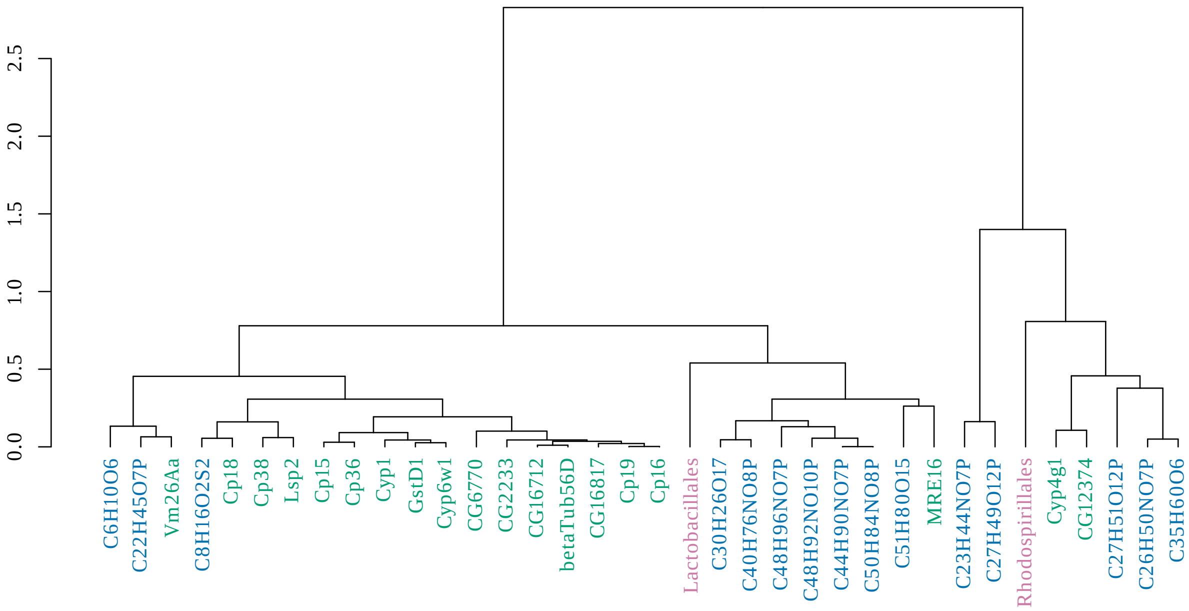

Figure 5: Hierarchical clustering of the first two pairs of canonical directions.

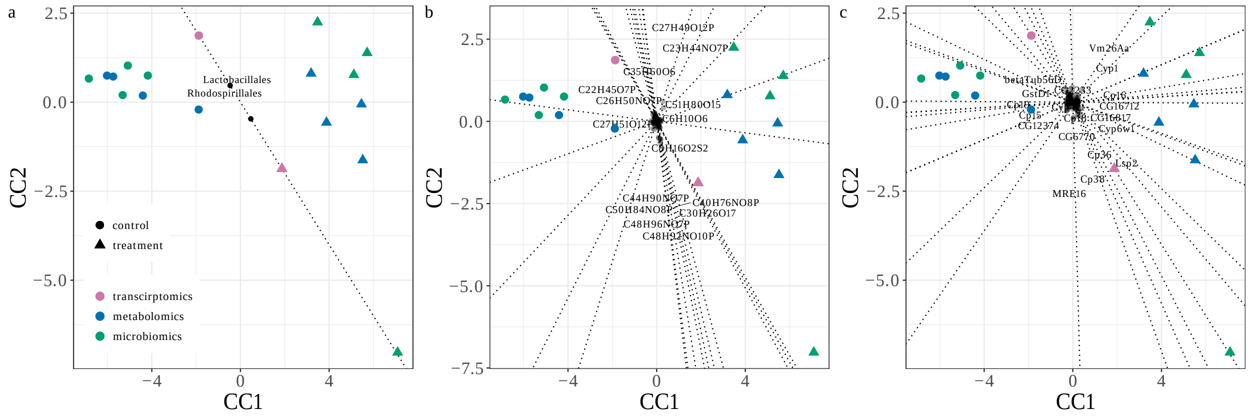

Figure 6: Interpolative plots of microbiomics(a), metabolomics(b), and transcriptomics(c) views. Any given sample is interpolated by either the complete parallelogram or the vector sum method explained in Appendix D.2

9. Conclusion

A two-stage approach to sparse CCA problem was introduced, where in the first stage we computed the sparsity patterns of the canonical directions via a fast, convergent concave minimization program. Then we used these sparsity patterns to shrink our problem to a CCA problem of two drastically smaller matrices, where regular CCA methods may be used. We then extended our methods to multi-view settings, i.e. *Multi-View Sparse CCA*, where we have more than two views and also to scenarios where our objective is to generate targeted hypotheses about associations corresponding to a specific experimental design, i.e. *Directed Sparse CCA*. We benchmarked our algorithm and also compared it to several other popular algorithms. Our simulations clearly demonstrated superior solution stability and convergence properties, as well as higher accuracy both in terms of the correlation of the estimated canonical covariates and also in terms of its ability to recover the underlying sparsity patterns of the canonical directions. We also introduced `MuLe` which is the package implementing our algorithms. We then applied our method to a multi-omic study aiming to understand mechanisms of adaptations of *Drosophila Melanoger (Fruitfly)* to environmental pesticides, here *Atrazine*. Our analysis clearly indicated that the estimated canonical directions, while sparse and interpretable, captures co-variations due to the treatment effect, and also the selected sets of covariates are known, according to the peer-reviewed literature, to be associated with adaptation mechanisms of fruitfly to environmental pesticides and stressors.

Appendix A. Proofs

A.1 Proof of Proposition 15

Let's assume without loss of generality that $p_1 \leq p_2$. In this case we can start `MuLe` to find the sparsity pattern of $\boldsymbol{z}_2 \in \mathbb{R}^{p_2}$ first, shrink $\boldsymbol{X}_2$ to $\boldsymbol{X}_{2red} \in \mathbb{R}^{n \times n_2^{'}}$ where $n_2^{'} \sim n$, then repeat the same for $\boldsymbol{z}_1 \in \mathbb{R}^{p_1}$, shrink $\boldsymbol{X}_1$ to $\boldsymbol{X}_{1red} \in \mathbb{R}^{n \times n_1^{'}}$ where $n_1^{'} \sim n$, and finally compute the first canonical covariates using the shrunken $\boldsymbol{X}_{1red}^T \boldsymbol{X}_{2red} \in \mathbb{R}^{n_1^{'} \times n_2^{'}}$.

According to the setup of algorithm 2, each iteration to find $\boldsymbol{\tau}_2$ is $O(2p_1p_2 + 4p_1 + p_1)$, using $\boldsymbol{\tau}_2$ and shrinking $\boldsymbol{X}_2$, each iteration for finding $\boldsymbol{\tau}_1$ is $O(2p_1n_2^{'} + 4p_1 + n_2^{'})$ which makes the time complexity of both $O(2p_1p_2 + 4p_1 + p_1 + 2p_1n_2^{'} + 4p_1 + n_2^{'})$. With `pSVDht`, the time complexity of both passes together is $O(4p_1p_2 + 2(p_1 + p_2))$. Assuming $p_2/p_1 = k = o(1)$ and $n \sim n_2^{'}$, if

$$n < \frac{2kp_1^2 - 2(k+1)p_1 - p_1}{2p_1 + 1}$$

The time complexity of `MuLe` is less than `pSVDht`. If $p_1 >> 1$,

$$\frac{2kp_1^2 - 2(k+1)p_1 - p_1}{2p_1 + 1} \approx \frac{2kp_1^2 - 2(k+1)p_1 - p_1}{2p_1} = kp_1 - (k + 0.5) > p_1$$

So as long as $n < min\{p_1, p_2\}$, our claim stands.

A.2 Proof of Proposition 16

Here, just to provide more clarity, Algorithm 3 of Witten and Tibshirani (2009) is provided as a representative for the bigger family of `sSVD` algorithms.

Algorithm 8: *$PMD(L_1, L_1)$* as proposed in Witten and Tibshirani (2009)

```
Data: Sample Covariance Matrices $\Sigma_{12} = X_1^T X_2$
      $l_1$-penalty parameters $c_1, c_2$
Result: $z_1 \in \mathbb{R}^{p_1}$, $z_2 \in \mathbb{R}^{p_2}$, and $d = z_1^T \Sigma_{12} z_2$
1 Initialize $z_2$ to have $l2 - norm$ 1;
2 while convergence criterion is not met do
3     $z_1 \leftarrow \frac{S(\Sigma_{12} z_2, \Delta_1)}{\|S(\Sigma_{12} z_2, \Delta_1)\|_2}$ where $\Delta_1 = 0$ if this results in $\|z_1\|_1 \leq c_1$; otherwise, $\Delta_1$ is chosen
        to be a positive constant such that $\|z_1\|_1 = c$
4     $z_2 \leftarrow \frac{S(\Sigma_{12}^T z_1, \Delta_2)}{\|S(\Sigma_{12}^T z_1, \Delta_2)\|_2}$ where $\Delta_2 = 0$ if this results in $\|z_2\|_1 \leq c_2$; otherwise, $\Delta_2$ is chosen
        to be a positive constant such that $\|z_2\|_1 = c$
5     $d \leftarrow z_1^T \Sigma_{12} z_2$
```

There is no need for a detailed time complexity analysis, as it is evident that although `MuLe` has order two polynomial time complexity, refer to Appendix A.1, the optimization problems in stages 3 and 4 of *PMD*, i.e. finding Δ_1 and Δ_2 that results in $\|z_1\|_1 = c_1$ and $\|z_2\|_1 = c_2$, are of exponential time complexity $O(2_1^p)$ and $O(2_2^p)$. They propose a binary search algorithm for this problem which has less time complexity but doesn't have guaranteed convergence, neither heuristically nor theoretically. In the implementation of the algorithm in the `PMA` package, the maximum number of iterations is set to a very small number, replacing which with a convergence criteria did not prove to be successful.

Appendix B. Complementary Methods and Algorithms

B.1 Multi-Factor MuLe

Algorithm 9: Multi-Factor MuLe

```
Data: Sample Covariance Matrix $\boldsymbol{C}_{12}$
      Regularization parameter vectors $\boldsymbol{\gamma}_i \in \mathbb{R}^m, i \in \{1,2\}$
      Initial value vectors $\boldsymbol{z}_i \in \mathcal{S}^{p_i}, i \in \{1,2\}$
Result: $\boldsymbol{Z}_i \in \mathbb{R}^{p_i \times m}, i \in \{1,2\}$
Let $\boldsymbol{C}_{12}^{(0)} \leftarrow \boldsymbol{C}_{12}$
for $i = 1, \dots, m$ do
    $(\boldsymbol{z}_1^{*(i)}, \boldsymbol{Z}_2^{*(i)} \leftarrow sCCA_{MuLe}(\boldsymbol{C}_{12}^{(i-1)}, \gamma_{1i}, \gamma_{2i})$
    $\boldsymbol{C}_{12}^{(i)} = \boldsymbol{C}_{12} - \sum_{k=1}^{i}(\boldsymbol{z}_1^{(k)*\top}\boldsymbol{C}_{12}^{(k-1)}\boldsymbol{z}_2^{(k)*})\boldsymbol{z}_1^{(k)*}\boldsymbol{z}_2^{(k)*\top}$
    $(\boldsymbol{Z}_1[,i], \boldsymbol{Z}_2[,i]) \leftarrow (\boldsymbol{z}_1^{*(i)}, \boldsymbol{z}_2^{*(i)})$
```

B.2 Multi-View CCA as Generalized Eigenvalue Problem

Here, we frame the CCA problem applied to multiple datasets, $\boldsymbol{X}_i$, $i = 1, \dots, m$, analyzed in Kettenring (1971) as the following *Generalized Eigenvalue Problem*,

$$\begin{bmatrix} \mathbf{0} & \boldsymbol{C}'_{12} & \dots & \boldsymbol{C}'_{1m} \\ \boldsymbol{C}'_{21} & \mathbf{0} & & \vdots \\ \vdots & & \ddots & \boldsymbol{C}'_{(m-1)m} \\ \boldsymbol{C}'_{m1} & & \boldsymbol{C}'_{m(m-1)} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}'_1 \\ \boldsymbol{z}'_2 \\ \vdots \\ \boldsymbol{z}'_m \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{C}'_{11} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{C}'_{22} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{C}'_{mm} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}'_1 \\ \boldsymbol{z}'_2 \\ \vdots \\ \boldsymbol{z}'_m \end{bmatrix} \tag{70}$$

where $\boldsymbol{C}'_{ij}$ is the shrunken $\boldsymbol{C}_{ij}$, or the sample covariance matrix of the active entries of $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$, denoted here as $\boldsymbol{z}'_i$ and $\boldsymbol{z}'_j$. Equation 70 can be solved using a wide variety of solvers. We used the `geigen`[4] function which is implemented in an r-package of the same name, which uses the routines implemented in `LAPACK`[5]. Given that m is usually less than 10, and $\boldsymbol{z}'_i = O(n)$, where n is not very large given we're assuming high-dimensional settings, problem 70 does not involve very large matrices.

B.3 Multi-View SVD via Power Iteration

We proposed a Multi-View CCA in Appendix B.2 which served as the second stage of out two-stage sCCA approach which was to estimate active elements of the canonical directions. Although 70 is of reasonable size, it still requires inversions which might be deemed as a disadvantage. Although it's very trivial to use ridge regularization to alleviate this issue, here we propose an algorithm which uses power iterations to perform multi-View SVD.

[4] https://CRAN.R-project.org/package=geigen

[5] http://github.com/Reference-LAPACK

Algorithm 10: `MuLe` algorithm for optimizing Program 39

```
Data: Shrunk Sample Covariance Matrices $\boldsymbol{C}'_{rs}$,   $1 \le r < s \le m$
      Initial values $\boldsymbol{z}'_r \in \mathcal{S}^{|\boldsymbol{\tau}_r|}$,   $1 \le r \le m$
Result: $\boldsymbol{z}'_r$, $r = 1, \dots, m$, estimated active elements of $\boldsymbol{z}_r$
initialization;
for $r = m, \dots, 1$ do
    while convergence criterion is not met do
        $\boldsymbol{z}'_r \leftarrow \sum_{s=1}^{r} \boldsymbol{C}_{sr}(\boldsymbol{C}_{sr}^\top \boldsymbol{z}'_r) + \sum_{s=r+1}^{m} \boldsymbol{C}'_{rs}\boldsymbol{z}'_s$
        $\boldsymbol{z}_r \leftarrow \frac{\boldsymbol{z}_r}{\|\boldsymbol{z}_r\|_2}$
```

B.4 Two-Stage Directed CCA

Though simple and obvious, we include this approach in this appendix for the sake of clarity and completeness. Here are the steps for this algorithm.

1. Perform variable selection via univariate regression or classification of $\boldsymbol{y}$ on each $\boldsymbol{X}_i$ resulting in a set of variables, Q_i, which are highly associated with the accessory variable.

2. Subset every datasets such that only the columns selected in the previous steps are kept, resulting in $\boldsymbol{X}'_i \in \mathbb{R}^{n \times |Q_i|}$.

3. Perform sCCA between the datasets using any of the algorithms implemented in `MuLe`.

Appendix C. Further Experimmentations

C.1 Rank-One Sparse Multi-View CCA Model

To assess the validity of the formulation presented in Program 32 and accuracy of our solution and algorithm presented in Section 6.4, for the cases involving more than two, the rank-one model introduced in Section 7.1 is extended to three datasets by generating $\mathbf{X}_3$ as follows,

$$\begin{aligned} \mathbf{X}_3 = (\mathbf{z}_3 + \epsilon_3)u^T, \quad \mathbf{z}_3 \in \mathcal{R}^{600}, \quad \epsilon_3 \sim \mathcal{N}(0, 0.1^2), \forall i = 1, \dots, 600, \\ \mathbf{z}_1 = \begin{bmatrix} \underbrace{1, \dots, 1}_{25} & \underbrace{0, \dots, 0}_{550} & \underbrace{-1, \dots, -1}_{25} \end{bmatrix} \end{aligned} \tag{71}$$

where $\mathbf{u}_i \sim \mathcal{N}(0, 1), \forall i = 1, \dots, 50$.

The coefficient estimates are presented in Figure 7. Here, we also included the `RGCCA` package. Although their conventional sCCA algorithm results were identical to `PMA`, their generalization to more than two datasets resulted in different and better results. Hence, its inclusion in this simulation. We used each package's own built-in hyper-parameter tuning procedure to find the best parameters. As evident from the results, `MuLe` identifies the underlying model quite accurately, but `RGCCA` although does a good job on parameter estimation, it does a very poor job on recovering the sparsity patterns of the canonical directions. `PMA` misses both critera quite significantly.

In the next section we utilize `MuLe` to discover correlation structures in a genomic setting.

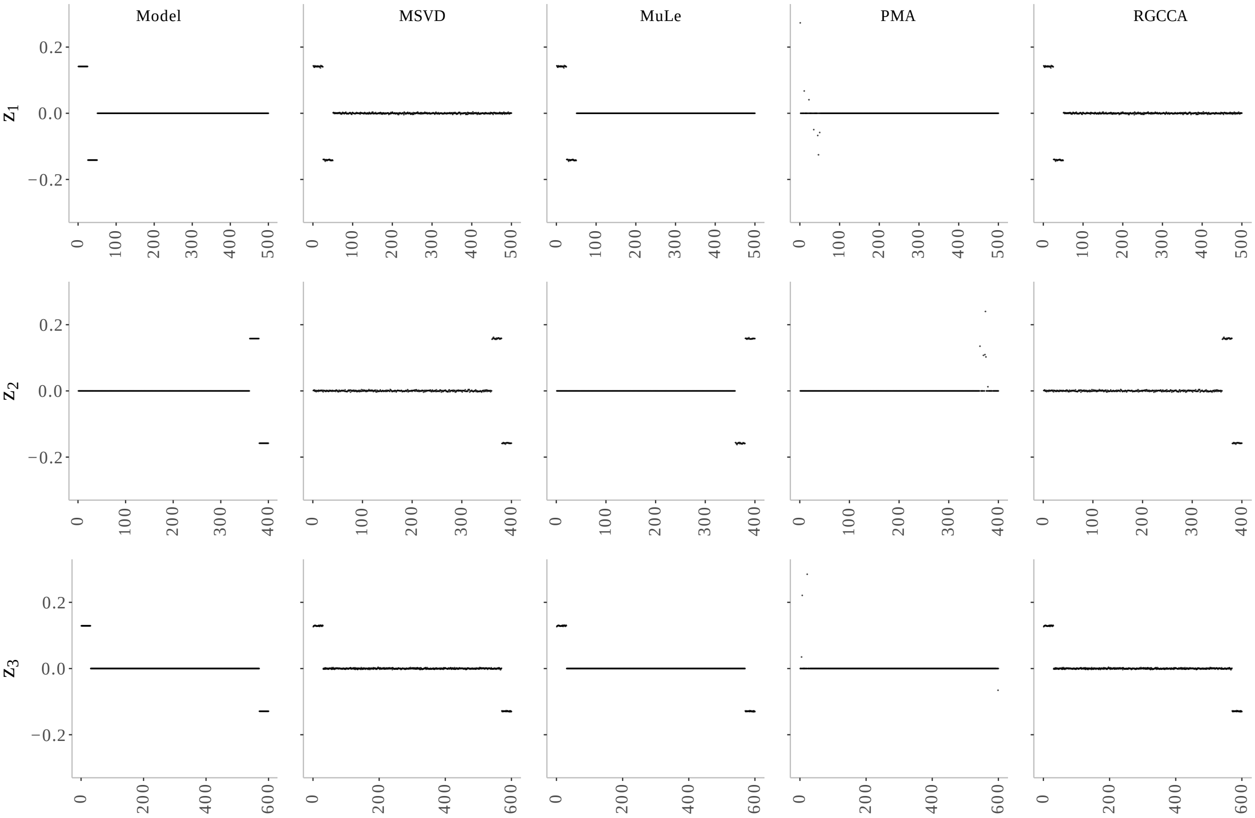

Figure 7: Comparing performance of some of the most common multi-view sCCA approaches to that of *MuLe* in recovering the sparsity pattern and estimating active elements of the canonical directions. The *Model* or "true" canonical directions are plotted in the leftmost plot.

Appendix D. Visualization Methods

In a general subspace learning problem involving datasets, we're seeking to replace each dataset with three low-dimensional pieces of information, a rule for projecting the original covariates to the learned subspace for the respective subspace, a low-dimensional projection of samples from the original sample-space to the learned sub-space, and a measure of similarity or alignment between the learned subspace. In our linear sCCA context, we replace the dataset $\boldsymbol{X}_i$ with $\boldsymbol{Z}_i$ whose rows contain the correlation of the covariate $\boldsymbol{x}_i$ with the canonical covariates, $\boldsymbol{CC}_i$ the projection of samples onto the canonical directions and the canonical correlations $\boldsymbol{\rho}_i^{(j)} \in \mathbb{R}^m$ containing the correlation between the j-th canonical covariate of the i-th dataset and the j-th canonical covariates obtained from other datasets. Now we explain the procedures used to create the figures in Section 8 which facilitate the interpretation of sCCA results. Inspired by the methods proposed in Alves and Oliveira (2003), we adapt their CCA biplot and interpolative plot to our sCCA settings. In the following brief tutorial, we focus on the first two canonical covariates, thereby keeping only the first two columns of $\boldsymbol{Z}_i$ and $\boldsymbol{CC}_i$, denoted by $\boldsymbol{Z}_i^{(2)}$ and $\boldsymbol{CC}_i^{(2)}$, and only $\boldsymbol{\rho}_i^{(j)}$ for $j \in \{1, 2\}$ and $i = \{1, \dots, m\}$.

D.1 CCA Biplot

In order to create the CCA biplot, e.g. Figure 4, we simply plot the first two columns of $\boldsymbol{Z}_i^{(2)}$ in the same plot. A key complementary piece of information facilitating interpretation are the first two canonical correlations. Utilizing at this plot, we can form hypotheses about how and to what extend groups of variables from different datasets are associated with each other. The length of the vectors indicate the variable's share in each canonical direction, while the angle between them indicate their degree of association.

D.2 CCA Interpolative Plots

Another informative visualization we exploit to interpret sCCA results are *Interpolative CCA Plots*, e.g. Figure 6. In order to create such figure for each dataset, we first plot $\boldsymbol{CC}_i$ from all datasets in the same plot, which by itself provides enlightening insights into how strongly the samples from different datasets align with each other. Next we need to add lines corresponding to the variables from the respective dataset. In order to make interpolation easier and the plots more clear, we first choose a set of marker points $\boldsymbol{\mu}_{ij}$ corresponding to the j-th variable from the i-th dataset, consisting of values within the range of observed values of the variable $\boldsymbol{x}_{ij}$, i.e. $\mu_{ijk} \in [min(\boldsymbol{x}_{ij}), max(\boldsymbol{x}_{ij})]$. We project these points using the following projection $\boldsymbol{\mu}_{ij}\boldsymbol{e}_{ij}\boldsymbol{V}_i^{(2)}$, where $\boldsymbol{e}_{ij}$ is a vector whose elements except the j-th is zeroed out. Finally, we pass a line through the projected points. Marking the values of each variable corresponding to a sample as a vector along each variable we can find the interpolated position of the said sample. This is a powerful tool as we can find how accurately we can interpolate a samples position using the values of a different dataset. This is specially important in cases where sample matching from different datasets are not exact and samples are matched based on some other metadata, e.g. gender, age etc.

Supplemental Materials: Sparse Canonical Correlation Analysis via Concave Minimization

S 1. MuLe Package

An R-implementation of our package `MuLe`, named `MuLe-R`, along with the scripts used to perform the simulations and create the visualizations, and the data used in Section 8 is available online at `https://github.com/osolari/MuleR`.

References

S. Akaho. A kernel method for canonical correlation analysis. *In Proceedings of the International Meeting of the Psychometric Society*, 2001.

Md. A. Alam, M. Nasser, and K. Fukumizu. Sensitivity analysis in robust and kernel canonical correlation analysis. *11th International Conference on Computer and Information Technology*, 0:399–404, 2008.

M Rui Alves and M Beatriz Oliveira. Interpolative biplots applied to principal component analysis and canonical correlation analysis. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 17(11):594–602, 2003.

G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. *International Conference on Machine Learning*, pages 1247–1255, 2013.

Khelifa Arab, Adrien Rossary, Laurent Soulere, and Jean-Paul Steghens. Conjugated linoleic acid, unlike other unsaturated fatty acids, strongly induces glutathione synthesis without any lipoperoxidation. *British Journal of Nutrition*, 96(5):811–819, 2006.

F.R. Bach and M.I. Jordan. Kernel independent component analysis. *Journal of machine learning research*, pages 1–48, 2002.

F.R. Bach and M.I. Jordan. A probabilistic interpretation of canonical correlation analysis. *Technical Report*, 2005.

B. Baur and S. Bozdag. A canonical correlation analysis-based dynamic bayesian network prior to infer gene regulatory networks from multiple types of biological data. *Journal of Computational Biology*, 22(4):289–299, 2015.

Harold P Benson. Concave minimization: theory, applications and algorithms. In *Handbook of global optimization*, pages 43–148. Springer, 1995.

M.B. Blaschko, C.H. Lampert, and A. Gretton. Semi-supervised laplacian regularization of kernel canonical correlation analysis. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 0:133–145, 2008.

James B Brown, Nathan Boley, Robert Eisman, Gemma E May, Marcus H Stoiber, Michael O Duff, Ben W Booth, Jiayu Wen, Soo Park, Ana Maria Suzuki, et al. Diversity and dynamics of the drosophila transcriptome. *Nature*, 512(7515):393, 2014.

J. Cai. The distance between feature subspaces of kernel canonical correlation analysis. *Mathematical and Computer Modelling*, 3:970–975, 2013.

Bruno Campos and John K Colbourne. How omics technologies can enhance chemical safety regulation: perspectives from academia, government, and industry: The perspectives column is a regular series designed to discuss and evaluate potentially competing viewpoints and research findings on current environmental issues. *Environmental toxicology and chemistry*, 37(5):1252, 2018.

L. Cao, Z. Ju, J. Li, R. Jian, and C. Jiang. Sequence detection analysis based on canonical correlation for steady-state visual evoked potential brain computer interfaces. *Journal of neuroscience methods*, 0(253):10–17, 2015.

James Angus Chandler, Jenna Morgan Lang, Srijak Bhatnagar, Jonathan A Eisen, and Artyom Kopp. Bacterial communities of diverse drosophila species: ecological context of a host–microbe model system. *PLoS genetics*, 7(9):e1002272, 2011.

A. Cichonska, J. Rousu, P. Marttinen, A.J. Kangas, P. Soininen, T. Lehtimäki, O.T. Raitakari, M.R. Järvelin, V. Salomaa, M Ala-Korpela, and others. metacca: Summary statistics-based multivariate meta-analysis of genome-wide association studies using canonical correlation analysis. *Bioinformatics*, 32:1981–9, 2016.

R.B. Dunham and D.J. Kravetz. Canonical correlation analysis in a predictive system. *The Journal of Experimental Education*, 43(4):35–42, 1975.

Alexandre d'Aspremont, Francis Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9(Jul):1269–1294, 2008.

L.M. Ewerbring and F.T. Luk. Canonical correlations and generalized svd: applications and new algorithms. *In 32nd Annual Technical Symposium, International Society for Optics and Photonics*, page 206–222, 1989.

J. Fang, D. Lin, Z. Xu S.C. Schulz, V.D. Calhoun, , and Y.P. Wang. Joint sparse canonical correlation analysis for detecting differential imaging genetics modules. *Bioinformatics*, 32(22): 3480–3488, 2016.

O. Friman, J. Cedefamn, P. Lundberg, M. Borga, and H. Knutsson. Detection of neural activity in functional mri using canonical correlation analysis. *Magnetic Resonance in Medicine*, 45:323–330, 2001.

T. Van Gestel, J.A.K. Suykens, J. De Brabanter, B. De Moor, and J. Vandewalle. Kernel canonical correlation analysis and least squares support vector machines. *International Conference on Artificial Neural Networks.*, pages 384–389, 2001.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

D.R. Hardoon and J. Shawe-Taylor. Convergence analysis of kernel canonical correlation analysis: theory and practice. *Machine learning.*, 1:23–38, 2009.

D.R. Hardoon and J. Shawe-Taylor. Sparse canonical correlation analysis. *Machine Learning*, 3: 331–353, 2011.

M.J.R. Healy. A rotation method for computing canonical correlations. *Math. Comp.*, 58:83–86, 1957.

C. Heij and B. Roorda. A modified canonical correlation approach to approximate state space modeling. *Proceedings of the 30th IEEE Conference on Decision and Control*, pages 1343–1348, 1991.

Ida Holásková, Meenal Elliott, Kathleen Brundage, Ewa Lukomska, Rosana Schafer, and John B Barnett. Long-term immunotoxic effects of oral prenatal and neonatal atrazine exposure. *Toxicological Sciences*, 168(2):497–507, 2019.

C.E. Hopkins. Statistical analysis by canonical correlation: a computer application. *Health services research*, 4(4):304, 1969.

H. Hotelling. The most predictable criterion. *Journal of Educational Psychology*, 26:139–142, 1935.

E. Hyman, P. Kauraniemi, S. Hautaniemi, M. Wolf, S. Mousses, E. Rozen-blum, M. Ringner, G. Sauter, O. Monni, A. Elkahloun, O.-P. Kallioniemi, and A. Kallioniemi. Impact of dna amplication on gene expression patterns in breast cancer. *Cancer Research*, 0(62):6240–6245, 2002.

M. Journée, Y. Nesterov, P. Richtrárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11:517–553, 2010.

Jon R Kettenring. Canonical analysis of several sets of variables. *Biometrika*, 58(3):433–451, 1971.

A. Klami, S. Virtanen, and S. Kaski. Bayesian exponential family projections for coupled data sources. *arXiv:1203.3489*, 2012.

P.L. Lai and C. Fyfe. A neural implementation of canonical correlation analysis. *Neural Networks*, 10:1391–1397, 1999.

P.L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10:365–377, 2000.

N.B. Larson, G.D. Jenkins, M.C. Larson, R.A. Vierkant, T.A. Sellers, C.M. Phelan, J.M. Schildkraut, R. Sutphen, P.P.D. Pharoah, S. A. Gayther, et al. Kernel canonical correlation analysis for assessing gene–gene interactions and application to ovarian cancer. *European Journal of Human Genetics*, 1:126–131, 2014.

H. Lindsey, J.T. Webster, , and S. Halper. Canonical correlation as a discriminant tool in a periodontal problem. *Biometrical journal*, 3(27):257–264, 1985.

OL Mangasarian. Machine learning via polyhedral concave minimization. In *Applied Mathematics and Parallel Computing*, pages 175–188. Springer, 1996.

T. Melzer, M. Reiter, and H. Bischof. Nonlinear feature extraction using generalized canonical correlation analysis. *International Conference on Artificial Neural Networks.*, 0:353–360, 2001.

James Mercer. Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209(441-458):415–446, 1909.

M.S. Monmonier and F.E. Finn. Improving the interpretation of geographical canonical correlation models. *The Professional Geographer*, 25:140–142, 1973.

M. Morley, C. Molony, T. Weber, J. Devlin, K. Ewens, R. Spielman, and V. Cheung. Genetic analysis of genome-wide variation in human gene expression. *Nature*, 0(430):743–747, 2004.

M. Nakanishi, Y. Wang, Y.T Wang, and T.P. Jung. A comparison study of canonical correlation analysis based methods for detecting steady-state visual evoked potentials. *PloS one*, 10(10): 10–17, 2015.

T. Ogura, Y. Fujikoshi, and T. Sugiyama. A variable selection criterion for two sets of principal component scores in principal canonical correlation analysis. *Communications in Statistics-Theory and Methods*, 42(12):2118–2135, 2013.

Luisa Orsini, James B Brown, Omid Shams Solari, Dong Li, Shan He, Ram Podicheti, Marcus H Stoiber, Katina I Spanier, Donald Gilbert, Mieke Jansen, et al. Early transcriptional response pathways in daphnia magna are coordinated in networks of crustacean-specific genes. *Molecular ecology*, 27(4):886–897, 2018.

T. B. M. J. Ouarda, C. Girard, G. S. Cavadias, and B. Bobée. Regional flood frequency estimation with canonical correlation analysis. *Journal of Hydrology*, 254:157–173, December 2001. doi: 10.1016/S0022-1694(01)00488-7.

E. Parkhomenko, D. Tritchler, and J. Beyene. Genome-wide sparse canonical correlation of gene expression with genotypes. *BMC Proceedings*, 1:s119, 2007.

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

J. Pollack, T. Sorlie, C. Perou, C. Rees, S. Jerey, P. Lonning, R. Tibshi-rani, D. Botstein, A. Borresen-Dale, and P. Brown. Microarray analysis reveals a major direct role of dna copy number alteration in the transcriptional program of human breast tumors. *Proceedings of the National Academy of Sciences*, 0(99):12963–12968, 2002.

J. Rousu, D.D. Agranoff, O. Sodeinde, J. Shawe-Taylor, and D. Fernandez-Reyes. Biomarker discovery by sparse canonical correlation analysis of complex clinical phenotypes of tuberculosis and malaria. *PLoS Comput Biol*, 9(4), 2013.

B.K. Sarkar and C. Chakraborty. Dna pattern recognition using canonical correlation algorithm. *Journal of biosciences*, 40(4):709–719, 2015.

S.V. Schell and W.A. Gardner. Programmable canonical correlation analysis: A flexible framework for blind adaptive spatial filtering. *IEEE transactions on signal processing*, 43(12):2898–2908, 1995.

Namrata Sengupta, Elizabeth J Litoff, and William S Baldwin. The hr96 activator, atrazine, reduces sensitivity of d. magna to triclosan and dha. *Chemosphere*, 128:299–306, 2015.

J.A. Seoane, C. Campbell, I.N.M. Day, J.P. Casas, and T.R. Gaunt. Canonical correlation analysis for genebased pleiotropy discovery. *PLoS Comput Biol*, 10(10), 2014.

X-B Shen, Q-S Sun, and Y-H Yuan. Orthogonal canonical correlation analysis and its application in feature fusion. *16th International Conference on Information Fusion*, pages 151–157, 2013.

Matthew H Sieber and Carl S Thummel. The dhr96 nuclear receptor controls triacylglycerol homeostasis in drosophila. *Cell metabolism*, 10(6):481–490, 2009.

D. Simonson, J. Stowe, and C. Watson. A canonical correlation analysis of commercial bank asset/liability structures. *Journal of Financial and Quantitative Analysis*, 10:125–140, 1983.

Antoine M Snijders, Sasha A Langley, Young-Mo Kim, Colin J Brislawn, Cecilia Noecker, Erika M Zink, Sarah J Fansler, Cameron P Casey, Darla R Miller, Yurong Huang, et al. Influence of early life exposure, host genetics and diet on the mouse gut microbiome and metabolome. *Nature microbiology*, 2(2):16221, 2017.

X.M. Tu, D.S. Burdick, D.W. Millican, and L.B. McGown. Canonical correlation technique for rank estimation of excitation-emission matrices. *Analytical Chemistry*, 19(61):2219–2224, 1989.

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

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

G.C. Wang, N. Lin, and B. Zhang. Dimension reduction in functional regression using mixed data canonical correlation analysis. *Stat Interface*, 6:187–196, 2013.

Weiran Wang, Xinchen Yan, Honglak Lee, and Karen Livescu. Deep variational canonical correlation analysis. *arXiv preprint arXiv:1610.03454*, 2016.

F.V. Waugh. Regressions between sets of variables. *Econometrica, Journal of the Econometric Society*, page 290–310, 1942.

Ami Wiesel, Mark Kliger, and Alfred O Hero III. A greedy approach to sparse canonical correlation analysis. *arXiv preprint arXiv:0801.2748*, 2008.

D. Witten and R. Tibshirani. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical Applications in Genomics and Molecular Biology*, 8, 2009.

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

K.W. Wong, P.C.W. Fung, and C.C. Lau. Study of the mathematical approximations made in the basis correlation method and those made in the canonical-transformation method for an interacting bose gas. *Physical Review*, 3(22):1272, 1980.

Y. Yamanishi, J.P. Vert, A. Nakaya, and M. Kanehisa. Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis. *Bioinformatics*, 19: i323–i330, 2003.

Chapter 2

BLOCCS: Block Sparse Canonical Correlation Analysis WithApplication To Interpretable Omics Integration

Abstract

We introduce the first *Sparse Canonical Correlation Analysis (sCCA)* approach which is able to estimate the leading d pairs of canonical directions of a pair of datasets (together a "block") at once, rather than the common deflation scheme, resulting in significantly improved orthogonality of the sparse directions – which translates to more interpretable solutions. We term our approach *block sCCA*. Our approach builds on the sparse CCA method of Solari et al. (2019) in that we also express the bi-convex objective of our block formulation as a concave minimization problem, whose search domain is shrunk significantly to its boundaries, which is then optimized via gradient descent algorithm. Our simulations show that our method significantly outperforms existing sCCA algorithms and implementations in terms of computational cost and stability, mainly due to the drastic shrinkage of our search space, and the correlation within and orthogonality between pairs of estimated canonical covariates. Finally, we apply our method, available as an R-package called BLOCCS, to multi-omic data on *Lung Squamous Cell Carcinoma(LUSC)* obtained via *The Cancer Genome Atlas*, and demonstrate its capability in capturing meaningful biological associations relevant to the hypothesis under study rather than spurious dominant variations.

1. Introduction

Multi-view[1] observations, i.e. observations of multiple random vectors or feature sets on matching subjects– i.e., heterogeneous datasets, are increasingly ubiquitous in data science. Particularly, in molecular biology, multiple ”omics” layers are regularly collected – measurements that sample comprehensively from an underlying pool of molecules, such as a genome, or the set of all RNA transcripts, known as the transcriptome. For example, The Cancer Genome Atlas (TCGA) is a multi-omics molecular characterization of tumors across thousands of patients. In such studies, we are often interested in understanding how two or more omics layers, or views, are related to one another – e.g., how genotype relates to gene expression, revealing transcriptional regulatory relationships – for a review see Li et al. (2016). This is very different from classical regression settings, where we have a one-dimensional response that we aim to model as a function of a vector of explanatory variables. As a result, new models are needed to enable the discovery of interpretable hypotheses regarding the association structures in multi-view settings, including multi-omics.

Canonical Correlation Analysis(CCA), Hotelling (1935), is one set of such models whose objective is to find linear combinations of two sets of random variables such that they are maximally correlated. CCA is the most popular approach up to date in such settings which has been applied in almost all areas of science including: medicine Monmonier and Finn (1973), policy Hopkins (1969), physics Wong et al. (1980), chemistry Tu et al. (1989), and finance Simonson et al. (1983). Several variants of CCA to incorporate non-linear combinations of covariates, e.g. Kernel CCA of Lai and Fyfe (2000) and Deep CCA of Andrew et al. (2013), have also been widely particularly popular in neuro-imaging Blaschko et al. (2011), computer vision Huang et al. (2010), and genetics Chaudhary et al. (2018).

Despite various improvements in multi-view models, inference, interpretability and model selection is still a challenge, which is mainly owed to very high-dimensional multi-view observations that become increasingly common as high-throughput measurement systems advance. Variable selection via sparsity inducing norms is a popular approach to identifying interpretable association structures in such high-dimensional settings, which are particularly important since, from a biological perspective, it is likely that responses of interest arise from the action of genes functioning in pathways. In other words, for a particular outcome, such as disease-free survival in particular cancer, not all genes are relevant, or, to use the multi-view learning parlance, ”active”. Hence, the derivation of sparse models from the analysis of multi-omics data is of intrinsic interest to biological data scientists.

While several sparse CCA methods are available, Witten and Tibshirani (2009), Parkhomenko et al. (2009), Waaijenborg et al. (2008), Chu D. (2013), their lack of stability and empirical consistency, and additionally their high computational cost, makes them unsuitable non-parametric hypothesis testing or hyperparameter tuning. Solari et al. (2019) introduce `MuLe` which is a set of approaches to solving sparse CCA problems using power iterations. They demonstrate superior stability and empirical consistency compared to other popular algorithms as well as significantly lower computational cost. One shortcoming however, which is common among all sparse CCA and sparse PCA approaches, is that none guarantee, or even heuristically enforce, orthogonality between estimated canonical directions. Here, our approach also relies on power iterations; however,

[1] Each dataset, denoted by $\boldsymbol{X}_i \in \mathbb{R}^{n \times p_i}$ in this paper, containing observations on random vectors is termed a *view* in this article.

we address the lack of orthogonality by estimating multiple canonical directions at once – adapting a block formulation for novel use in sparse CCA Journée et al. (2010).

2. Notation

We term the observed random vector $X_i(\omega) : \Omega \to \mathbb{R}^{p_i}$, denoted by $\boldsymbol{X}_i \in \mathbb{R}^{n \times p_i}$, $i = 1, \ldots, m$, a *view*. We denote scalar, vector, and matrix parameters by lower-case normal, lower-case bold, and upper-case bold letters, respectively, and random variables by upper-case normal letters. n is used to indicate the sample size and p_i the dimensionality of the covariate space of each of m views. Canonical directions are denoted by $\boldsymbol{z}_i \in \mathcal{B}^{p_i}$, or $\boldsymbol{z}_i \in \mathcal{S}^{p_i}$, and $\boldsymbol{Z}_i \in \mathcal{S}_d^{p_i}$, where $\mathcal{B} = \{\boldsymbol{x} \in \mathbb{R} | \|\boldsymbol{x}\|_2 \leq 1\}$ and $\mathcal{S} = \{\boldsymbol{x} \in \mathbb{R} | \|\boldsymbol{x}\|_2 = 1\}$. $\mathcal{S}_d^p = \{\boldsymbol{Z} \in \mathbb{R}^{p \times d} | \boldsymbol{Z}^\top \boldsymbol{Z} = \boldsymbol{I}_d\}$ denotes a *Stiefel* manifold which is the set of all d-frames, i.e. the space of ordered sets of d linearly independent vectors, in $\mathbb{R}^p$. $l_x(\boldsymbol{z}) : \mathbb{R}^p \to \mathbb{R}$ denotes any norm function, more specifically $l_{0/1}(\boldsymbol{z}) = \|\boldsymbol{z}\|_{0/1}$, and $\boldsymbol{\tau}^{(i)}$ refers to the $i - th$ non-zero element of the vector which is specifically used for the sparsity pattern vector. We also introduce *accessory variables* in Section 4.3 to term variables towards which we direct estimated canonical directions, neglecting their inferential role as covariates or dependent variables. We also use "program" to refer to "optimization programs".

3. Background

Sub-space learning is perhaps the most popular concept in multi-view learning, and implies a *Latent Space* generative model, where each view, $X_i(\omega) : \mathcal{U} \to \mathcal{X}_i, i = 1, \ldots, m$, is assumed to be a function of a common unobservable random vector, $U : \Omega \to \mathcal{U}$ in the latent space. The main objective in subspace learning is to estimate the inverse of these mappings within a functional family, $\mathcal{F}_i = \{F_i : \mathcal{X}_i \to \mathcal{U}\}$ assuming invertibility. At the sample level, this is interpreted as estimating $F_i(X_i)$ by $\boldsymbol{F}_i : \mathbb{R}^{n \times p_i} \to \mathcal{U}^n$ such that $\mathcal{S} : \mathcal{U}^{n \times m} \to \mathbb{R}^d$, $\mathcal{S} = (s_1, \ldots, s_d)$, where $s(F_1(X_1), \ldots, F_m(X_m)) : \mathcal{U}^{n \times m} \to \mathbb{R}$ is some similarity measure between these transformed observed views is maximized,

$$\boldsymbol{F}^* = \underset{\substack{F_i \in \mathcal{F}_i \\ i \in \{1,\ldots,m\}}}{\arg\max} \, \mathcal{S}(F_1(X_1), \ldots, F_m(X_m)) \tag{1}$$

Where $\boldsymbol{F} = (F_1, \ldots, F_m)$. d is the number of dimensions in which similarity is maximized, which is of importance since here we are concerned with block algorithms where $d > 1$, i.e. we estimate d distinct mappings for each view at the same time such that these mappings maximize $\mathcal{S}$. In the rest of this section and most of Section 4 we assume that we observe only a pair of views, i.e. $m = 2$. Throughout this paper we also assume that $U : \Omega \to \mathbb{R}^k$, $X_i : \mathbb{R}^k \to \mathbb{R}^{p_i}$.

3.1 Canonical Correlation Analysis

If we assert the functional families $\mathcal{F}_i$ to be a subset of the parametric family of linear functions $\mathcal{L} = \{l_i : \mathbb{R}^{p_i} \to \mathbb{R}^k, l_i(X_i) = \boldsymbol{z}_i X_i\}$, and the similarity criterion to be the Pearson correlation, we end up with the *Canonical Correlation Analysis* criterion. Assuming $E[X_1] = \boldsymbol{0}^{p_1}$ and $E[X_2] = \boldsymbol{0}^{p_2}$,

$$\begin{aligned}(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*) &= \underset{\boldsymbol{z}_1 \in \mathbb{R}^{p_1}, \boldsymbol{z}_2 \in \mathbb{R}^{p_2}}{\arg\max} \; \rho(X_1\boldsymbol{z}_1, X_2\boldsymbol{z}_2) \\ &= \underset{\boldsymbol{z}_1 \in \mathbb{R}^{p_1}, \boldsymbol{z}_2 \in \mathbb{R}^{p_2}}{\arg\max} \; \frac{E[(X_1\boldsymbol{z}_1)^\top (X_2\boldsymbol{z_2})]}{E[(X_1\boldsymbol{z}_1)^2]^{1/2} E[(X_2\boldsymbol{z_2})^2]^{1/2}}\end{aligned} \tag{2}$$

Since we almost always have access only to samples from X_1 and X_2, we estimate Program 2 using plug-in sample estimators for population terms.

$$(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*) = \underset{\boldsymbol{z}_1 \in \mathbb{R}^{p_1}, \boldsymbol{z}_2 \in \mathbb{R}^{p_2}}{\arg\max} \; \frac{\boldsymbol{z}_1^\top \boldsymbol{X}_1^\top \boldsymbol{X}_2 \boldsymbol{z}_2}{\sqrt{\boldsymbol{z}_1^\top \boldsymbol{X}_1^\top \boldsymbol{X}_1 \boldsymbol{z}_1}\sqrt{\boldsymbol{z}_2^\top \boldsymbol{X}_2^\top \boldsymbol{X}_2 \boldsymbol{z}_2}} \tag{3}$$

$\boldsymbol{z}_i$ are termed *Canonical Loading Vectors* and $\boldsymbol{X}_i\boldsymbol{z}_i$ are called the *Canonical Covariates.*

4. Block Reformulations of CCA Models

Generalizing Program 3 to $\boldsymbol{Z}_i \in \mathbb{R}^{p_i \times d}$,

$$(\boldsymbol{Z}_1^*, \boldsymbol{Z}_2^*) = \underset{\substack{\boldsymbol{Z}_1 \in \mathbb{R}^{p_1 \times d}, \boldsymbol{Z}_2 \in \mathbb{R}^{p_2 \times d} \\ \boldsymbol{Z}_1^\top \boldsymbol{X}_1^\top \boldsymbol{X}_1 \boldsymbol{Z}_1 = \boldsymbol{Z}_2^\top \boldsymbol{X}_2^\top \boldsymbol{X}_2 \boldsymbol{Z}_2 = \boldsymbol{I}^d}}{\arg\max} \; tr(\boldsymbol{Z}_1^\top \boldsymbol{X}_1^\top \boldsymbol{X}_2 \boldsymbol{Z}_2) \tag{4}$$

Here we reserve the term *"block formulation"* to discuss settings in which $d > 1$, i.e. we estimate multiple pairs of canonical directions at once, $\boldsymbol{Z}_i \in \mathbb{R}^{p_i \times d}$ $i = 1, 2$ rather than a single pair of canonical directions $\boldsymbol{Z}_i \in \mathbb{R}^{p_i}$, $i = 1, 2$. As is customary in the sparse CCA literature, here we also assume that the covariance matrix of each random vector is diagonal, i.e. $\boldsymbol{X}_i^\top \boldsymbol{X}_i = \boldsymbol{I}^{p_i}, i = 1, 2$, which is justified in Dudoit et al. (2002). This enables us to rewrite Program 4 as,

$$(\boldsymbol{Z}_1^*, \boldsymbol{Z}_2^*) = \underset{\substack{\boldsymbol{Z}_1 \in \mathcal{S}_d^{p_1} \\ \boldsymbol{Z}_2 \in \mathcal{S}_d^{p_2}}}{\arg\max} tr(\boldsymbol{Z}_1^\top \boldsymbol{X}_1^\top \boldsymbol{X}_2 \boldsymbol{Z}_2) \tag{5}$$

Where $\mathcal{S}_d^{p_1}$ is a *Stiefel Manifold*[2]

4.1 Regularized Block CCA

We analyze the following generalized formulation of the sparse block CCA problem in this section,

$$\begin{aligned}\phi_{l,d}(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2) := & \max_{\substack{\boldsymbol{Z}_1 \in \mathcal{S}_d^{p_1} \\ \boldsymbol{Z}_2 \in \mathcal{S}_d^{p_2}}} tr(\boldsymbol{Z}_1^\top \boldsymbol{C}_{12} \boldsymbol{Z}_2 \boldsymbol{N}) \\ & - \sum_{j=1}^{d} \gamma_{1j} l(\boldsymbol{z}_{1j}) - \sum_{j=1}^{d} \gamma_{2j} l(\boldsymbol{z}_{2j})\end{aligned} \tag{6}$$

[2] $\mathcal{S}_m^p = \{\boldsymbol{M} \in \mathbb{R}^{p \times d} | \boldsymbol{M}^\top \boldsymbol{M} = \boldsymbol{I}\}$

$\boldsymbol{\gamma}_i \in \mathbb{R}^d, \boldsymbol{\gamma}_i \geq 0$ is the sparsity parameter vector for each view, and $\boldsymbol{N} = diag(\boldsymbol{\mu}), \boldsymbol{\mu} \in \mathbb{R}^+$, where d is the number of canonical covariates. $l(\boldsymbol{z}_{ij})$ is some norm of the $j-th$ column of the $i-th$ view, and $\boldsymbol{C}_{12}$ is the sample covariance matrix.

Remark 1 *In practice, distinct μ_i enforces the objective in Program 6 to have distinct maximizers Journée et al. (2010).*

4.1.1 L_1 Regularization

Here we consider Program 6 with L_1 regularization, and decouple the problem along multiple canonical directions resulting in the following program,

$$\begin{aligned}\phi_{l_1,d}(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2) = \max_{\boldsymbol{Z}_1 \in \mathcal{S}_d^{p_1}} \sum_{j=1}^{d} \max_{\boldsymbol{z}_{2j} \in \mathcal{S}^{p_2}} [\mu_j \boldsymbol{z}_{1j}^\top \boldsymbol{C}_{12} \boldsymbol{z}_{2j} - \gamma_{2j} \|\boldsymbol{z}_{2j}\|_1] \\ - \sum_{j=1}^{d} \gamma_{1j} \|\boldsymbol{z}_{1j}\|_1\end{aligned} \tag{7}$$

where $\boldsymbol{z}_{ij}$ is the j-th column of the i-th dataset.

Theorem 2 *Maximizers $\boldsymbol{Z}_1^*$ and $\boldsymbol{Z}_2^*$ of Program 7 are,*

$$\boldsymbol{Z}_1^* = \arg\max_{\boldsymbol{Z}_1 \in \mathcal{S}_d^{p_1}} \sum_{j=1}^{d} \sum_{i=1}^{p_2} [\mu_j |\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}| - \gamma_{2j}]_+^2 - \sum_{j=1}^{d} \gamma_{1j} \|\boldsymbol{z}_{1j}\|_1 \tag{8}$$

and,

$$[\boldsymbol{Z}_2]_{ij}^* = \frac{sgn(\boldsymbol{c}_i^\top \boldsymbol{z}_{1j})[\mu_j |\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}| - \gamma_{2j}]_+}{\sqrt{\sum_{k=1}^{p_2} [\mu_j |\boldsymbol{c}_k^\top \boldsymbol{z}_{1j}| - \gamma_{2j}]_+^2}} \tag{9}$$

Equation 9 is utilized to derive the necessary and sufficient conditions under which z_{2ji}^* is active, i.e. inferring the sparsity pattern matrix, $supp(\boldsymbol{Z})$, which is denoted her by $\boldsymbol{T}_2 \in \{0,1\}^{p_2 \times d}$.

Corollary 3 $[\boldsymbol{T}_2]_{ij} = 0$, *i.e.* $z_{2ji}^* \in supp(\boldsymbol{Z}_2^*)$, *iff* $|\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}^*| \leq \gamma_{2j}/\mu_j$.

Theorem 2 enables us to infer the the sparsity pattern of either of the canonical directions due to the symmetry of the problem. Assuming we estimate $\boldsymbol{T}_2$ first, we shrink the sample covariance matrix to $[\boldsymbol{C}'_{12}]_{kl} = [\boldsymbol{C}_{12}]_{k\boldsymbol{\tau}_{2j}^{(l)}}$ where $\boldsymbol{\tau}_{2j}^{(l)}$ is the l-th non-zero element of the j-th column of $\boldsymbol{T}_2$. We then use this reduced covariance matrix to estimate $\boldsymbol{T}_1$. Having estimated the sparsity pattern matrices in the first stage, we estimate the active elements of the canonical direction matrices in the second stage by first shrinking the covariance matrix on both sides, resulting in $[\boldsymbol{C}_{12}^{(j)}]_{kl} = [\boldsymbol{C}_{12}]_{\boldsymbol{\tau}_{1j}^{(k)},\boldsymbol{\tau}_{2j}^{(l)}}$, then estimating its active elements via an alternating algorithm introduced in 5.2.

Remark 4 *According to Theorem 2, in order to infer the sparsity pattern matrices, we need to maximize Program 9. This program is non-convex; however we approximate it by ignoring the penalty term which turns it into the following concave minimization over the unit sphere,*

$$\phi_{l_1,d}(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2) = \max_{\boldsymbol{Z}_1 \in \mathcal{S}_d^{p_1}} \sum_{j=1}^{d} \{ \sum_{i=1}^{p_2} [\mu_j |\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}| - \gamma_{2j}]_+^2 \} \tag{10}$$

which is solved using a simple gradient ascent algorithm. It is important to note that this approximation is justifiable. Our simulations demonstrate that this approximation does not affect the capability of our approach to uncover the support of our underlying generative model. Secondly, as we have mentioned in Corollary 3, we use the optima of this program in the first stage to infer the sparsity patterns of canonical directions. Also we can show that for every $(\gamma_{1j}, \gamma_{2j})$ that results in $\boldsymbol{z}_{1j}^* = 0$ according to the Corollary 3, there is a $\gamma_{2j}' \geq \gamma_{2j}$ in Program 10 for which $z_{2ji}^* = 0$.

In the rest of this section we introduce *Block Sparse Multi-View CCA* and *Block Sparse Directed CCA.*

4.2 L_1 Regularized Block Multi-View CCA

Now we extend our approach from 4.1.1 to identify correlation structures between more than two views, $\boldsymbol{X}_i \in \mathbb{R}^{n \times p_i}, i = 1, \dots, m$. The application of such methods are ever-increasing, e.g. understanding the enriched genetic pathways in a population of patients with a specific type of cancer. We extend the approach introduced in Solari et al. (2019) to our block setting, which results in the following optimization program,

$$\begin{aligned} \phi_{l_1,d}^m(\boldsymbol{\Gamma}_1, \dots, \boldsymbol{\Gamma}_d) = & \max_{\substack{\boldsymbol{Z}_i \in \mathcal{S}_d^{p_i} \\ \forall i=1,\dots,m}} \sum_{r<s=2}^{m} tr(\boldsymbol{Z}_r^\top \boldsymbol{C}_{rs} \boldsymbol{Z}_s \boldsymbol{N}) \\ & - \sum_{j=1}^{d} \sum_{s=2}^{m} \sum_{\substack{r=1 \\ r \neq s}}^{s-1} \gamma_{srj} \|\boldsymbol{z}_{sj}\|_1 \end{aligned} \tag{11}$$

where $\boldsymbol{\Gamma}_j \in [0,1]^{p_j \times M}$ are the sparsity parameter matrices whose elements γ_{srj} regulate the sparsity of canonical direction $\boldsymbol{z}_{sj}$ in relation to $\boldsymbol{z}_{rj}$, where $\boldsymbol{z}_{sj}$ is the j-th column of $\boldsymbol{Z}_s$. As before $\boldsymbol{C}_{rs} = 1/n \boldsymbol{X}_r^\top \boldsymbol{X}_s$ is a sample covariance matrix.

Theorem 5 *Maximizers $\boldsymbol{Z}_i^*, i = 1, \dots, m$ of Program 11 are,*

$$\begin{aligned} & z_{sij}^*(\gamma_{sr1}, \dots, \gamma_{srd}) = \\ & \frac{sgn(\sum_{\substack{r=1 \\ r\neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_{rj})[\mu_j | \sum_{\substack{r=1 \\ r\neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_{rj}| - \sum_{\substack{r=1 \\ r\neq s}}^{m} \gamma_{srj}]_+}{\sqrt{\sum_{k=1}^{p_2} [\mu_j | \sum_{\substack{r=1 \\ r\neq s}}^{m} \tilde{\boldsymbol{c}}_{rsk}^\top \boldsymbol{z}_{rj}| - \sum_{\substack{r=1 \\ r\neq s}}^{m} \gamma_{srj}]_+^2}} \end{aligned} \tag{12}$$

and for $r = 1, \dots, m$ *and* $r \neq s$,

$$\boldsymbol{Z}_r^*(\boldsymbol{\Gamma}_1, \dots, \boldsymbol{\Gamma}_d) = \\ \underset{\substack{\boldsymbol{Z}_r \in \mathcal{S}_d^{p_r} \\ r \neq s, r=1,\dots,m}}{\arg\max} \sum_{j=1}^{d} \sum_{i=1}^{p_s} [\mu_j | \sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_{rj}| - \sum_{\substack{r=1 \\ r \neq s}}^{m} \gamma_{srj}]_+^2 + \\ \sum_{\substack{i<r=2 \\ i,r \neq s}}^{m} tr(\boldsymbol{Z}_i^\top \boldsymbol{C}_{ir} \boldsymbol{Z}_r \boldsymbol{N}) - \sum_{j=1}^{d} \sum_{\substack{i=1 \\ i \neq s}}^{m} \sum_{\substack{r=1 \\ i \neq j}}^{s-1} \gamma_{irj} \|\boldsymbol{z}_{ij}\|_1 \tag{13}$$

Similar to the previous section, we drop the last term in Program 13 following the same justifications offered in Remark 4. This approximation leaves us with a concave minimization program which can be solved in a significantly faster and more stable way.

Corollary 6 *Given the sparsity parameter matrices* $\boldsymbol{\Gamma}_i, i = 1, \dots, d$ *and the solution,* $\boldsymbol{Z}_r^*$ *for* $r = 1, \dots, m$ *and* $r \neq s$, *to the Program 13,*

$$[\boldsymbol{T}_s]_{ij} = \begin{cases} 0 & |\sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_{rj}| \leq 1/\mu_j \sum_{\substack{r=1 \\ r \neq s}}^{m} \gamma_{srj} \\ 1 & otherwise \end{cases} \tag{14}$$

4.3 L_1 Regularized Directed CCA

Often samples involved in a multi-view learning problem are part of a designed experiment which differ along the direction of some treatment vector, or an observational study where we have information about the samples in addition to the observed views, e.g. socioeconomic status, sex, education level, etc. Solari et al. (2019) coined the term *Accessory Variable* to avoid confusions with the rich lexicon of statistical inference, to point out that this extra piece of information will be solely used to direct canonical directions such that they capture correlation structures which also align with these accessory variables, denoted here by $\boldsymbol{Y} \in \mathbb{R}^{n \times d}$, towards each column of which we direct the canonical directions. To this end, we form the following optimization problem,

$$\phi_{l,d}(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2) = \max_{\substack{\boldsymbol{Z}_1 \in \mathcal{S}_d^{p_1} \\ \boldsymbol{Z}_2 \in \mathcal{S}_d^{p_2}}} tr(\boldsymbol{Z}_1^\top \boldsymbol{C}_{12} \boldsymbol{Z}_2 \boldsymbol{N}) \\ - \sum_{i=1}^{2} [\mathcal{L}(\boldsymbol{X}_i \boldsymbol{Z}_i \boldsymbol{N} \boldsymbol{E}_i, \boldsymbol{Y}) + \boldsymbol{\gamma}_i^\top \boldsymbol{l}(\boldsymbol{Z}_i)] \tag{15}$$

where $\boldsymbol{E}_i = diag(\boldsymbol{\epsilon}_i)$ are diagonal hyper-parameter matrices controlling the effect of the accessory variables on the canonical directions. $\mathcal{L}(\boldsymbol{A}, \boldsymbol{B}) : \mathcal{X}_A \times \mathcal{X}_B \to \mathbb{R}$ is a measure of column-wise misalignment of $\boldsymbol{A}$ and $\boldsymbol{B}$. Here, we choose the Euclidean inner-product as our alignment measure, i.e. $\mathcal{L}(\boldsymbol{X}_i \boldsymbol{Z}_i \boldsymbol{N} \boldsymbol{E}_i, \boldsymbol{y}) = -\langle \boldsymbol{X}_i \boldsymbol{Z}_i \boldsymbol{N} \boldsymbol{E}_i, \boldsymbol{Y} \rangle = -tr(\boldsymbol{Y}^\top \boldsymbol{X}_i \boldsymbol{Z}_i \boldsymbol{N} \boldsymbol{E}_i)$. Plugging in 15 and decoupling,

$$\begin{aligned}\phi_{l_1,d}(\boldsymbol{\gamma}_1,\boldsymbol{\gamma}_2) = \max_{\boldsymbol{Z}_1\in\mathcal{S}_d^{p_1}} \sum_{j=1}^{d} \max_{\boldsymbol{z}_{2j}\in\mathcal{S}^{p_2}} [\mu_j \boldsymbol{z}_{1j}^\top \boldsymbol{C}_{12}\boldsymbol{z}_{2j} \\ +(\mu_j\epsilon_{1j}\boldsymbol{y}_j^\top \boldsymbol{X}_2\boldsymbol{z}_{2j} - \gamma_{2j}\|\boldsymbol{z}_{2j}\|_1)] \\ +\sum_{j=1}^{d}(\mu_j\epsilon_{2j}\boldsymbol{y}_j^\top \boldsymbol{X}_1\boldsymbol{z}_{1j} - \gamma_{1j}\|\boldsymbol{z}_{1j}\|_1) \qquad (16)\end{aligned}$$

where $\boldsymbol{z}_{ij}$ is the j-th column of the i-th dataset.

Theorem 7 *Maximizers of Program 16 are,*

$$\boldsymbol{Z}_1^* = \arg\max_{\boldsymbol{Z}_1\in\mathcal{S}_d^{p_1}} \sum_{j=1}^{d}\sum_{i=1}^{p_2}[\mu_j|\boldsymbol{c}_i^\top \boldsymbol{z}_{1j} + \epsilon_{2j}\boldsymbol{x}_{2i}^\top \boldsymbol{y}_j| - \gamma_{2j}]_+^2 \qquad (17)$$

$$+\sum_{j=1}^{d}(\mu_j\epsilon_{1j}\boldsymbol{y}_j^\top \boldsymbol{X}_1\boldsymbol{z}_{1j} - \gamma_{1j}\|\boldsymbol{z}_{1j}\|_1) \qquad (18)$$

and,

$$[\boldsymbol{Z}_2]_{ij}^* = \qquad (19)$$

$$\frac{sgn(\boldsymbol{c}_i^\top \boldsymbol{z}_{1j} + \epsilon_{2j}\boldsymbol{x}_{2i}^\top \boldsymbol{y}_j)[\mu_j|\boldsymbol{c}_i^\top \boldsymbol{z}_{1j} + \epsilon_{2j}\boldsymbol{x}_{2i}^\top \boldsymbol{y}_j| - \gamma_{2j}]_+}{\sqrt{\sum_{k=1}^{p_2}[\mu_j|\boldsymbol{c}_k^\top \boldsymbol{z}_{1j} + \epsilon_{2j}\boldsymbol{x}_{2k}^\top \boldsymbol{y}_j| - \gamma_{2j}]_+^2}} \qquad (20)$$

In the following corollary we formalize the necessary and sufficient conditions under which z_{2ij}^* is active using Equation 19.

Corollary 8 $[\boldsymbol{T}_2]_{ij} = 0$, *iff* $|\boldsymbol{c}_k^\top \boldsymbol{z}_{1j}^* + \epsilon_{2j}\boldsymbol{x}_{2k}^\top \boldsymbol{y}_j| \leq \gamma_{2j}/\mu_j$.

In the following section we propose algorithms to solve the optimization programs discussed so far.

Please refer to the *Supplementals* for detailed proofs of the theorems and corollaries presented above as well as a discussion of l_0*-regularized* Canonical Correlation Analysis.

5. BLOCCS: Gradient Ascent Algorithms for Regularized Block Models

As discussed so far, we reformulated each of the four cases studied into a concave minimization program over a Stiefel manifold. Our proposed algorithms involve a simple first-order optimization method at their cores, see *Supplementals*. In 5.1 we apply this first-order method to the scenarios discussed so far, which constitutes the first stage of our two-stage approach. In the first stage, we estimate the sparsity patterns of our canonical directions. In the second stage we estimate the "active" entries (non-zero loadings) of the canonical directions using an alternating optimization algorithm discussed in 5.2.

5.1 Sparsity Pattern Estimation

In the first stage we estimate the sparsity patterns of the canonical directions, $\boldsymbol{T}_i$, by applying each of the following algorithms once for each dataset. As we move from estimating $\boldsymbol{T}_1$ to $\boldsymbol{T}_m$, we use a technique which we term *Successive Shrinking*, that is having estimated $\boldsymbol{T}_i$, we shrink every sample covariance matrix $\boldsymbol{C}_{ij}, j \neq i$ to $[\boldsymbol{C}'_{ij}]_{rs} = [\boldsymbol{C}_{ij}]_{\boldsymbol{\tau}_{ik}^{(r)} s}$, where $\boldsymbol{\tau}_{ik}^{(r)}$ is the r-th non-zero element of the k-th column of the i-th sparsity pattern matrix. As a result, in each successive shrinkage the covariance matrices are shrunk drastically, which in turn results in significant speed-up of our algorithm.

5.1.1 L_1 Regularized Algorithm

Now we apply our first-order maximization algorithm to Program 10,

Algorithm 1: `BLOCCS` algorithm for solving Program 10

```
Data: Sample Covariance Matrix $\boldsymbol{C}_{12}$
      Regularization parameter vector $\boldsymbol{\gamma_2} \in [0,1]^d$
      Initialization $\boldsymbol{Z}_1 \in \mathcal{S}_d^{p_1}$
      $\boldsymbol{N} = diag(\mu_1, \dots, \mu_d) \succ 0$
      (optional) $\boldsymbol{T}_1 \in \{0,1\}^{p_1 \times d}$
Result: $\boldsymbol{T}_2$, optimal sparsity pattern of $\boldsymbol{Z}_2^*$
initialization;
while convergence criterion is not met do
    for $j = 1, \dots, d$ do
        $\boldsymbol{z}_{1j} \leftarrow \sum_{i=1}^{p_2} \mu_j [\mu_j |\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}| - \gamma_2]_+ sgn(\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}) \boldsymbol{c}_i$
    $\boldsymbol{Z}_1 \leftarrow polar(\boldsymbol{Z}_1)$
    if $\boldsymbol{T}_1$ is given then
        $\boldsymbol{Z}_1 \leftarrow \boldsymbol{Z}_1 \circ \boldsymbol{T}_1$
Output $\boldsymbol{T}_2 \in \{0,1\}^{p_2 \times d}$ where $[\boldsymbol{T}_2]_{ij} = 0$ if $|\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}^*| \leq \gamma_{2j}/\mu_j$ and 1 otherwise.
```

As we pointed out above, we then compute $\boldsymbol{T}_1$ using successive shrinkage.

Remark 9 *One of the appealing qualities of our algorithm is that it is solely dependent on a function which can evaluate power iterations, which can be implemented very efficiently by exploiting sparse structures in the data matrix and canonical directions. This quality is significantly rewarded by successive shrinkage. It can also very easily be deployed on a distributed computing infrastructure. S. Solari et al. (2019) utilize this quality to offer a Spark-based distributed regularized multi-view learning package.*

5.1.2 Multi-View Block Sparse Algorithm

We now propose an algorithm to solve Program 13, leaving out the regularization term in the first stage.

Algorithm 2: BLOCCS algorithm for solving Program 13

```
Data: Sample Covariance Matrices $\boldsymbol{C}_{rs}$,   $1 \leq r < s \leq m$
      Sparsity parameter matrices $\boldsymbol{\Gamma}_j \in [0,1]^{m \times m}$ for $j = 1, \ldots, d$
      Initial values $\boldsymbol{Z}_r \in \mathcal{S}_d^{p_r}$,   $1 \leq r \leq m$
      $\boldsymbol{N} = diag(\mu_1, \ldots, \mu_d) \succ 0$
      (optional) $\boldsymbol{T}_r \in \{0,1\}^{p_r \times d}, r \neq s$
Result: $\boldsymbol{T}_s$, optimal sparsity pattern for $\boldsymbol{Z}_s$
initialization;
while convergence criterion is not met do
    for $r = 1, \ldots, m,\ r \neq s$ do
        for $j = 1, \ldots, d$ do
            $\boldsymbol{z}_{rj} \leftarrow \sum_{i=1}^{p_s} \mu_j [\mu_j | \sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_{rj}| - \sum_{\substack{r=1 \\ r \neq s}}^{m} \gamma_{srj}]_+ sgn(\sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_{rj}) \tilde{\boldsymbol{c}}_{rsi} + \mu_j \sum_{\substack{l=1 \\ l \neq r,s}}^{m} \tilde{\boldsymbol{C}}_{rl} \boldsymbol{z}_{lj}$
        $\boldsymbol{Z}_r \leftarrow polar(\boldsymbol{Z}_r)$
        if $\boldsymbol{T}_r$ is given then
            $\boldsymbol{Z}_r \leftarrow \boldsymbol{Z}_r \circ \boldsymbol{T}_r$
Output $\boldsymbol{T}_s \in \{0,1\}^{p_s \times d}$, $[T_s]_{ij} = 0$ if $| \sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_{rj}| \leq 1/\mu_j \sum_{\substack{r=1 \\ r \neq s}}^{m} \gamma_{srj}$ and 1 otherwise.
```

5.1.3 Directed Block Regularized Algorithm

Before we present our algorithm, it is helpful to realize that the directed regularized case in Program 16 is equivalent to the multi-modal case in Program 11 with $m = 3$ and $\boldsymbol{\epsilon}_i = \mathbf{1}_d$. As though we regard the accessory variable $\boldsymbol{y}$ as a third view. But many times the researcher wants to have a direct control on how much effect the accessory variable will have on the canonical directions. Basically the larger ϵ_{ij}, the smaller the aperture of the convex cone that contains both $\boldsymbol{y}$ and the canonical covariate $\boldsymbol{X}_i \boldsymbol{z}_i$. Below is the algorithm we devised for this problem,

In Section 6.2, we demonstrate the capabilities of this approach in exploratory data analysis and hypothesis development.

5.2 Active Entry Estimation

In the second stage of the algorithm, we estimate the active elements of the canonical directions for which, following Journée et al. (2010), we also propose alternating algorithm to solve the following optimization program,

$$\phi_{d,0} = \max_{\substack{\boldsymbol{Z}_1 \in \mathcal{S}_d^{p1}, \boldsymbol{Z}_1|_{\neq 0} = \boldsymbol{T}_1 \\ \boldsymbol{Z}_2 \in \mathcal{S}_d^{p2}, \boldsymbol{Z}_2|_{\neq 0} = \boldsymbol{T}_2}} tr(\boldsymbol{Z}_1^\top \boldsymbol{C}_{12} \boldsymbol{Z}_2 \boldsymbol{N}) \tag{21}$$

Our simulations show that for small enough $\boldsymbol{\gamma}_i, i = 1, 2$ such local maximizers exist.

The same algorithm is used in the multi-modal case by maximizing over a single $\boldsymbol{Z}_i$ while keeping others constant and looping over all canonical directions. In the directed case, we use the same $\boldsymbol{\epsilon}_i$ we used in the first stage and it's again very similar to the multi-modal case. Although

Algorithm 3: `BLOCSS` algorithm for solving Program 16

```
Data: Sample Covariance Matrix $\boldsymbol{C}_{12}$
      Regularization parameter vector $\boldsymbol{\gamma_2} \in [0,1]^d$
      Hyper-parameter vectors $\boldsymbol{\epsilon}_i \in \mathbb{R}^d, i = 1,2$
      Initialization $\boldsymbol{Z}_1 \in \mathcal{S}_d^{p_1}$
      $\boldsymbol{N} = diag(\mu_1, \dots, \mu_d) \succ 0$
      (optional) $\boldsymbol{T}_1 \in \{0,1\}^{p_1 \times d}$
Result: $\boldsymbol{T}_2$, optimal sparsity pattern of $\boldsymbol{Z}_2^*$
initialization;
while convergence criterion is not met do
    for $j = 1, \dots, d$ do
        $\boldsymbol{z}_{1j} \leftarrow \sum_{i=1}^{p_2} \mu_j [\mu_j |\boldsymbol{c}_i^\top \boldsymbol{z}_{1j} + \epsilon_{2j} \boldsymbol{x}_{2i}^\top \boldsymbol{y}| - \gamma_2]_+ sgn(\boldsymbol{c}_i^\top \boldsymbol{z}_{1j} + \epsilon_{2j} \boldsymbol{x}_{2i}^\top \boldsymbol{y}) \boldsymbol{c}_i + \epsilon_{1j} \boldsymbol{X}_1^\top \boldsymbol{y}$
    $\boldsymbol{Z}_1 \leftarrow polar(\boldsymbol{Z}_1)$
    if $\boldsymbol{T}_1$ is given then
        $\boldsymbol{Z}_1 \leftarrow \boldsymbol{Z}_1 \circ \boldsymbol{T}_1$
Output $\boldsymbol{T}_2 \in \{0,1\}^{p_2 \times d}$ where $[\boldsymbol{T}_2]_{ij} = 0$ if $|\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}^* + \epsilon_{2j} \boldsymbol{x}_{2i}^\top \boldsymbol{y}| \leq \gamma_{2j}/\mu_j$ and 1 otherwise.
```

Algorithm 4: `BLOCCS` algorithm for solving Program 21

```
Data: Sample Covariance Matrix $\boldsymbol{C}_{12}$
      Initialization $\boldsymbol{Z}_i \in \mathcal{S}_d^{p_i}$ for $i = 1,2$
      $\boldsymbol{N} = diag(\mu_1, \dots, \mu_d) \succ 0$
      $\boldsymbol{T}_i \in \{0,1\}^{p_i \times d}$ for $i = 1,2$
Result: $\boldsymbol{Z}_i^*, i = 1,2$, local maximizers of 21
initialization;
while convergence criterion is not met do
    $\boldsymbol{Z}_2 \rightarrow polar(\boldsymbol{C}_{12}^\top \boldsymbol{Z}_1 \boldsymbol{N}) \circ \boldsymbol{T}_2$
    $\boldsymbol{Z}_1 \rightarrow polar(\boldsymbol{C}_{12} \boldsymbol{Z}_2 \boldsymbol{N}) \circ \boldsymbol{T}_1$
```

simple, we've included the corresponding algorithms for the two cases as well as algorithm for the l_0-regularized CCA in the *Supplementals*.

6. Experiments

In this section we first demonstrate performance characteristics of BLOCCS on simulated data; then we apply our approach to *Lung Squamous Cell Carcinoma(LUSC)* multi-omics from *The Cancer Genome Atlas*Weinstein et al. (2013).

6.1 Simulated Data

Here we compare bloccs to PMA Witten and Tibshirani (2009), which is a commonly used package and is a good representative of the approaches based on alternating optimization scheme which is the dominant school of approaches to the sCCA problem. We applied both methods to the pairs of views $\boldsymbol{X}_i, i = 1, 2$ estimate the first two pairs of canonical directions $\boldsymbol{Z}_i, i = 1, 2$, where $\boldsymbol{X}_i \sim \mathcal{N}(\mathbf{0}_{p_i}, \boldsymbol{C}_{ii}), i = 1, 2$, and $\boldsymbol{C}_{ii} = \boldsymbol{V}_i \boldsymbol{D} \boldsymbol{V}_i^\top$. We chose $p_1 = p_2, p_i/n = 10$, and constructed $\boldsymbol{V}_1 \in \mathbb{R}^{p_1 \times p_1}$ by seting up the first two columns as

$$\boldsymbol{v}_{11} = [\underbrace{1, \ldots, 1}_{p_1/10}, 0, \ldots, 0], \boldsymbol{v}_{12} = [\underbrace{0, \ldots, 0}_{p_1/10}, \underbrace{1, \ldots, 1}_{p_1/10}, 0, \ldots, 0],$$

and the rest of the columns by sampling according to

$$[\boldsymbol{V}_{1j}]_{j=2}^{p_1} \sim \mathcal{N}(\mathbf{0}_{p_1-2}, \boldsymbol{I}_{p_1-2}).$$

Similarly, $\boldsymbol{V}_2 \in \mathbb{R}^{p_2 \times p_2}$,

$$\boldsymbol{v}_{21} = [0, \ldots, 0, \underbrace{1, \ldots, 1}_{p_2/10}], \boldsymbol{v}_{22} = [0, \ldots, 0, \underbrace{1, \ldots, 1}_{p_2/10} \underbrace{0, \ldots, 0}_{p_2/10}]$$

$$[\boldsymbol{V}_{2j}]_{j=2}^{p_2} \sim \mathcal{N}(\mathbf{0}_{p_2-2}, \boldsymbol{I}_{p_2-2})$$

We also set $\boldsymbol{D} = diag(\sigma_1, \sigma_2, \underbrace{\sigma, \ldots, \sigma}_{p_1-2})$, where $\sigma_1/\sigma_2 = 2$, and $\sigma_3 = \ldots = \sigma_{p_i} = \sigma$. We sampled $\boldsymbol{X}_i$ for 100 different values of σ, repeated 10 times, each time computing the average correlation of estimated canonical direction, $\boldsymbol{z}_{ij}$ and the underlying model, $\boldsymbol{z}_{ij} = \boldsymbol{v}_{ij}$ for $j = 1, 2$, see Figure 1.a and 1.b, and also the average correlation of the first and second estimated directions, see Figure 1.c, vs. the λ_3/λ_2, where λ_i is the i-th eigenvalue of the sample covariance matrix, $\boldsymbol{C}_{12}$. It is clear from Figure 1 that our approach learn the underlying model with superior accuracy while summarizing independent pieces of information in different canonical covariates. We guess that the apparent orthogonality of PMA estimates are mainly due to the fact that they contain minimal information about the underlying model.

6.2 TCGA: Lung Squamous Cell Carcinoma(LUSC)

We first performed sCCA between methylation and RNA-expression datasets obtained via TCGA2STAT [Wan et al. (2015)]. We used a permutation test, see Supplementals, for hyper-parameter tuning.

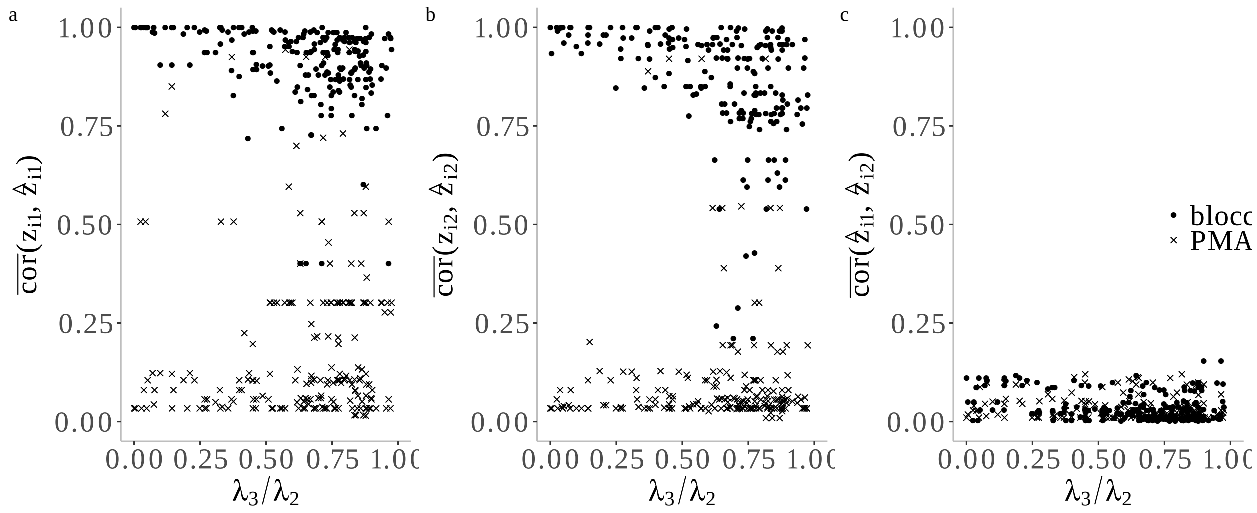

Figure 1: **a**,**b**. The average correlation of the "true", underlying model, and estimated first, and second respectively, pair of canonical directions. **c**. Average within pair correlation of the estimated directions. (plotted points are running medians).

While the analysis provided in Wan et al. (2015) filters out transcripts/CpG sites with expression/methylation level falling into the 99th percentile, we didn't filter out any covariates to simulate an fully automated pipeline. Despite this disadvantage, `bloccs` also identified two distinct clusters, with (between cluster distance)/(within cluster radius) = 9.79 compared to their 2.66, as plotted in Figure 2.a. However, contrary to their interpretation that these two groups indicate two different survival groups, as they point out the evidence against H_0*: two survival distributions are the same* is weak; A *Mantel-Cox* test returns $p - value = 0.062$, $\chi^2_1 = 3.5$. We found out that the clusters precisely capture the `sex` effect rather than survival. We repeated the analysis, but this time we used our novel *Directed sCCA* method of Algorithm 3 with $\hat{S}(t)$ as the accessory variable. As a result we identified 25 genes and 44 CpG sites which are associated with each other and also associated with survival. Projecting the individuals onto the canonical directions, we identified two distinct clusters using `kmeans` clustering, see Figure 2.c. We then computed the Kaplan-Meier curves for these two groups separately in Figure 2.d. These two distributions are significantly different with $p - value = 0.0058$, $\chi^2_1 = 7.6$.

7. Conclusion

We presented a block sparse CCA algorithm suitable for very high-dimensional settings. The method we propose and the software we provide are more stable than previous implementations of sparse CCA. Of particular interest to us is the felicity of this method to incorporate a "guide vector" – or an experimental design, termed *accessory variables* in this article. In our lung cancer example, we included empirical survival distribution as an accessory variable, and explored genes and CpG sites that are associated with each other and patient survival probability. Indeed, we find

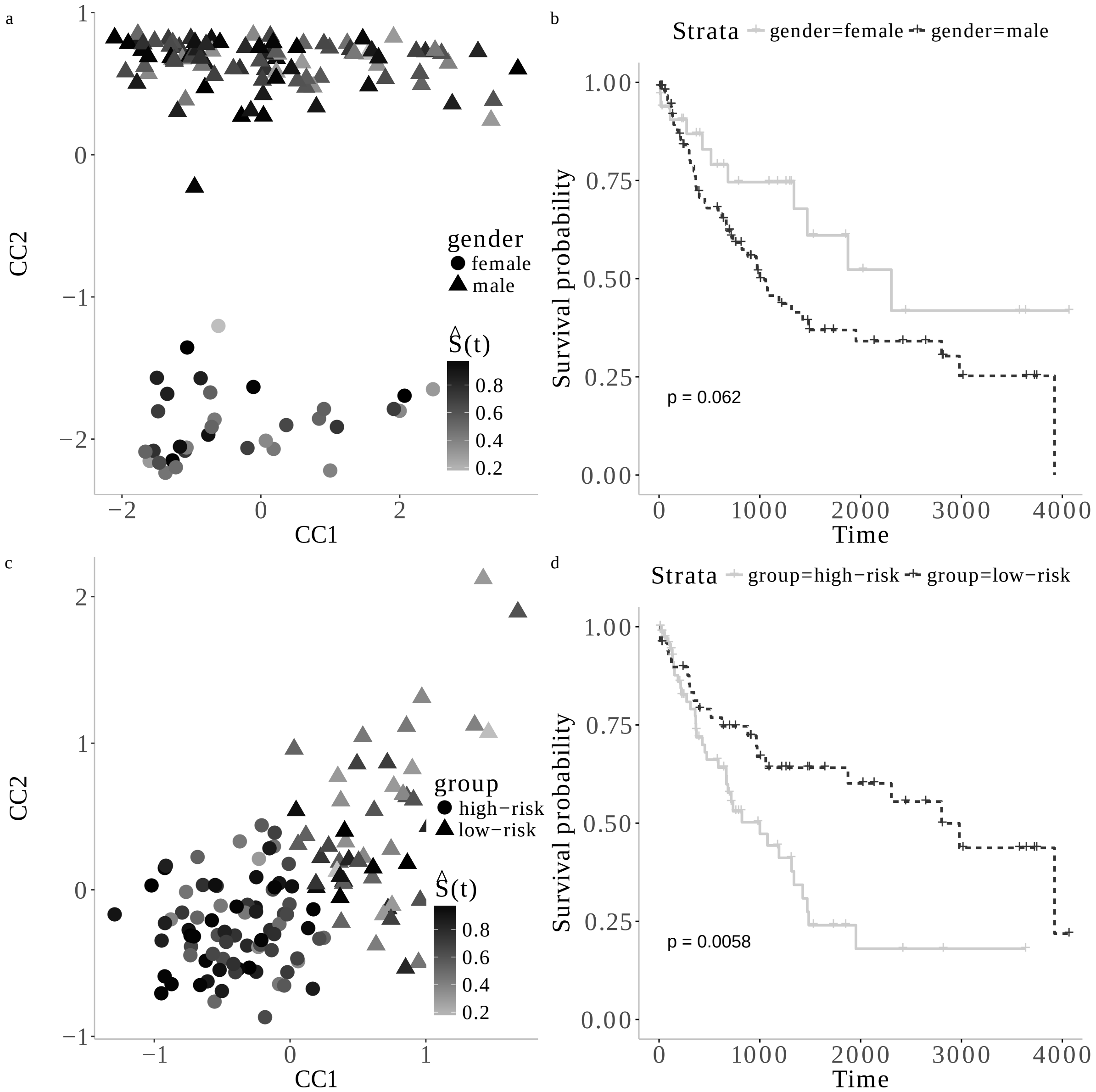

Figure 2: **a**. `kmeans` clustering of the samples projected onto the canonical directions estimated by applying *sCCA* to methylation and RNA-Seq datasets for LUSC patients, shape-coded by `gender`, and color-coded by $\hat{S}(t)$, i.e. the empirical survival distribution. **b**. $\hat{S}(t)$ for the two identified groups which precisely corresponded to `gender` rather than survival propability. **c**. `kmeans` clustering of the samples projected onto the canonical directions estimated by applying *Directed sCCA* to the same views and using $\hat{S}(t)$ as an accessory variable, color-coded by $\hat{S}(t)$. **d**. $\hat{S}(t)$ of the two identified groups by the Directed sCCA correspond to two significantly different, $p - value = 0.0058$, high-risk and low-risk survival groups.

the tuning parameters of our algorithm useful tools for data exploration, enabling the user to view a variety of relationships between views correlated more or less with an accessory variable. While multi-omics studies in biology were the motivation behind creating `bloccs`, we anticipate utility in a number of domains within and beyond the biomedical sciences.

Appendix A. Proofs of Theorems

A.1 Proof of Theorem 2

Proof

$$\begin{aligned}\phi_{l_1,d}(\boldsymbol{\gamma}_1,\boldsymbol{\gamma}_2) &= \max_{\boldsymbol{Z}_1\in\mathcal{S}_d^{p_1}} \sum_{j=1}^{d} \max_{\boldsymbol{z}_{2j}\in\mathcal{S}^{p_2}} [\mu_j \boldsymbol{z}_{1j}^\top \boldsymbol{C}_{12}\boldsymbol{z}_{2j} - \gamma_{2j}\|\boldsymbol{z}_{2j}\|_1] - \sum_{j=1}^{d}\gamma_{1j}\|\boldsymbol{z}_{1j}\|_1 \\ &= \max_{\boldsymbol{Z}_1\in\mathcal{S}_d^{p_1}} \sum_{j=1}^{d} \max_{\boldsymbol{z}_{2j}\in\mathcal{S}^{p_2}} [\sum_{i=1}^{p_2} z_{2ji}(\mu_j \boldsymbol{c}_i^\top \boldsymbol{z}_{1j}) - \gamma_{2j}\|\boldsymbol{z}_{2j}\|_1] - \sum_{j=1}^{d}\gamma_{1j}\|\boldsymbol{z}_{1j}\|_1 \qquad (22)\\ &= \max_{\boldsymbol{Z}_1\in\mathcal{S}_d^{p_1}} \sum_{j=1}^{d} \max_{\boldsymbol{z}_{2j}\in\mathcal{S}^{p_2}} [\sum_{i=1}^{p_2} |z'_{2ji}|(\mu_j|\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}| - \gamma_{2j})] - \sum_{j=1}^{d}\gamma_{1j}\|\boldsymbol{z}_{1j}\|_1\end{aligned}$$

where $z_{2ji} = sgn(\boldsymbol{c}_i^\top \boldsymbol{z}_{1j})\boldsymbol{z}'_{2ji}$. Maximizing over z'_{2ji} while keeping $\boldsymbol{z}_{1j}$ constant and transforming back to z_{2ji}, we obtain Equation 9. Substituting the result back in 22 we obtain the following optimization program,

$$\phi_{l_1,d}(\boldsymbol{\gamma}_1,\boldsymbol{\gamma}_2) = \max_{\boldsymbol{Z}_1\in\mathcal{S}_d^{p_1}} \sum_{j=1}^{d}\{\sum_{i=1}^{p_2}[\mu_j|\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}| - \gamma_{2j}]_+^2 - \gamma_{1j}\|\boldsymbol{z}_{1j}\|_1\} \qquad (23)$$

■

A.2 Proof of Corollary 3

Proof In light of Theorem 2,

$$z_{2ji} = 0 \Leftrightarrow [\mu_j|\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}^*| - \gamma_{2j}]_+ = 0 \Leftrightarrow |\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}^*| \leq \gamma_{2j}/\mu_j \qquad (24)$$

We can derive a sufficient condition even without solving for $\boldsymbol{Z}_1^*$ if we realize that $|\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}^*| \leq \|\boldsymbol{c}_i\|_2\|\boldsymbol{z}_{1j}^*\|_2 = \|\boldsymbol{c}_i\|_2$. So, $\|\boldsymbol{c}_i\|_2 \leq \gamma_{2j}/\mu_j$ is sufficient for $[\boldsymbol{T}_2]_{ij} = 0$.

■

A.3 Theorem 10

$$\phi_{l_0,d}(\boldsymbol{\gamma}_1,\boldsymbol{\gamma}_2) := \max_{\substack{\boldsymbol{Z}_1\in\mathcal{S}_d^{p_1}\\ \boldsymbol{Z}_2\in\mathcal{S}_d^{p_2}}} tr(diag(\boldsymbol{Z}_1^\top \boldsymbol{C}_{12}\boldsymbol{Z}_2\boldsymbol{N})^2) - \sum_{j=1}^{d}\gamma_{1j}\|\boldsymbol{z}_{1j}\|_0 - \sum_{j=1}^{d}\gamma_{2j}\|\boldsymbol{z}_{2j}\|_0 \qquad (25)$$

where as before $\boldsymbol{N} = diag(\mu_1,\ldots,\mu_d) \succ 0$, and $\gamma_{ij} \geq 0$.

Theorem 10 *The solutions $\boldsymbol{Z}_1^*$ and $\boldsymbol{Z}_2^*$ of the optimization program 25 is given by,*

$$\boldsymbol{Z}_1^* = \arg\max_{\boldsymbol{Z}_1\in\mathcal{S}_d^{p_1}} \sum_{j=1}^{d}\sum_{i=1}^{p_2}[(\mu_j \boldsymbol{c}_i^\top \boldsymbol{z}_{1j})^2 - \gamma_{2j}]_+ - \sum_{j=1}^{d}\gamma_{1j}\|\boldsymbol{z}_{1j}\|_0 \qquad (26)$$

and,

$$[\boldsymbol{Z}_2]^*_{ij} = \frac{[sgn((\mu_j \boldsymbol{c}_i^\top \boldsymbol{z}_{1j})^2 - \gamma_{2j})]_+ \mu_j \boldsymbol{c}_i^\top \boldsymbol{z}_{1j}}{\sqrt{\sum_{k=1}^{p_2} [sgn((\mu_j \boldsymbol{c}_k^\top \boldsymbol{z}_{1j})^2 - \gamma_{2j})]_+ (\mu_j \boldsymbol{c}_k^\top \boldsymbol{z}_{1j})^2}} \tag{27}$$

Proof Maximization problem 25 can be decoupled along different canonical directions as the following optimization problem over $\boldsymbol{Z}_1$,

$$\phi_{l_0,d}(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2) = \max_{\boldsymbol{Z}_1 \in \mathcal{S}_d^{p_1}} \sum_{j=1}^{d} \max_{\boldsymbol{z}_{2j} \in \mathcal{S}_d^{p_2}} [(\mu_j \boldsymbol{z}_{1j}^\top \boldsymbol{C}_{12} \boldsymbol{z}_{2j})^2 - \gamma_{2j} \|\boldsymbol{z}_{2j}\|_0] - \sum_{j=1}^{d} \gamma_{1j} \|\boldsymbol{z}_{1j}\|_0 \tag{28}$$

As in Theorem 2, we first solve for $\boldsymbol{z}_{2j}$ while keeping $\boldsymbol{Z}_1$ constant, resulting in Equation 27. The reason is that $z_{2ji} \neq 0$ only if the maximum objective value $(\mu_j \boldsymbol{c}_i^\top \boldsymbol{z}_{1j})^2 - \gamma_{2j}$ is positive. Now replacing back in 28 we obtain,

$$\phi_{l_0,d}(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2) = \max_{\boldsymbol{Z}_1 \in \mathcal{S}_d^{p_1}} \sum_{j=1}^{d} \{ \sum_{i=1}^{p_2} [(\mu_j \boldsymbol{c}_i^\top \boldsymbol{z}_{1j})^2 - \gamma_{2j}]_+ - \gamma_{1j} \|\boldsymbol{z}_{1j}\|_0 \} \tag{29}$$

■

Corollary 11 *$[\boldsymbol{T}_2]_{ij} = 0$, i.e. $z^*_{2ji} \in supp(\boldsymbol{Z}_2^*)$, iff $(\boldsymbol{c}_i^\top \boldsymbol{z}^*_{1j})^2 \leq \gamma_{2j}/\mu_j^2$.*

Proof According to Theorem 10,

$$z^*_{2ji} = 0 \Leftrightarrow [(\mu_j \boldsymbol{c}_i^\top \boldsymbol{z}^*_{1j})^2 - \gamma_{2j}]_+ = 0 \Leftrightarrow (\boldsymbol{c}_i^\top \boldsymbol{z}^*_{1j})^2 \leq \gamma_{2j}/\mu_j^2 \tag{30}$$

We can again derive a sufficient condition by just realizing that $(\boldsymbol{c}_i^\top \boldsymbol{z}^*_{1j})^2 \leq \|\boldsymbol{c}_i\|_2^2 \|\boldsymbol{z}^*_{1j}\|_2^2 = \|\boldsymbol{c}_i\|_2^2$. So, $\|\boldsymbol{c}_i\|_2^2 \leq \gamma_{2j}/\mu_j^2$ is sufficient for $[\boldsymbol{T}_2]_{ij} = 0$.

■

Remark 12 *According to Theorem 10, in order to infer the sparsity pattern matrices, we need to optimize Program 29 depending on the regularization of choice. This program is non-convex; however we approximate it by ignoring the penalty term which turns it into the following concave minimization programs over the unit sphere,*

$$\phi_{l_0,d}(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2) = \max_{\boldsymbol{Z}_1 \in \mathcal{S}_d^{p_1}} \sum_{j=1}^{d} \{ \sum_{i=1}^{p_2} [(\mu_j \boldsymbol{c}_i^\top \boldsymbol{z}_{1j})^2 - \gamma_{2j}]_+ \} \tag{31}$$

which is solved using a simple gradient ascent algorithm. It is important to note that this approximation is very reasonable and justifiable. Our simulations demonstrate that this approximation does not affect the capability of our approach to precisely uncover the support of our underlying generative model. Secondly, as we have mentioned in Corollary 11, we use the optima of this program in the first stage to infer the sparsity pattern of the canonical direction on the other side. Also we can show that for every $(\gamma_{1j}, \gamma_{2j})$ that results in $\boldsymbol{z}^*_{1j}$ for which 30 holds and as a result $z^*_{2ji} = 0$ in Program 29, one can find a $\gamma'_{2j} \geq \gamma_{2j}$ in Program 31 for which $z^*_{2ji} = 0$.

A.4 Proof of Theorem 5

Proof

$$\phi^{m}_{l_1,d}(\boldsymbol{\Gamma}_1,\ldots,\boldsymbol{\Gamma}_d) = \max_{\substack{\boldsymbol{Z}_r\in\mathcal{S}^{p_r}_d \\ r\neq s, r=1,\ldots,m}} \max_{\boldsymbol{Z}_s\in\mathcal{S}^{p_s}_d} \sum_{r<s=2}^{m} tr(\boldsymbol{Z}_r^\top \boldsymbol{C}_{rs}\boldsymbol{Z}_s\boldsymbol{N}) - \sum_{j=1}^{d}\sum_{s=2}^{m}\sum_{\substack{r=1\\ r\neq s}}^{s-1}\gamma_{srj}\|\boldsymbol{z}_{sj}\|_1 \tag{32}$$

$$= \max_{\substack{\boldsymbol{Z}_r\in\mathcal{S}^{p_r}_d \\ r\neq s, r=1,\ldots,m}} \sum_{j=1}^{d}[\max_{\boldsymbol{z}_{sj}\in\mathcal{S}^{p_s}} \sum_{r<s=2}^{m-1} \mu_j \boldsymbol{z}_{rj}^\top \boldsymbol{C}_{rs}\boldsymbol{z}_{sj} - \sum_{s=1}^{m}\sum_{\substack{r=1\\ r\neq s}}^{m-1}\gamma_{srj}\|\boldsymbol{z}_{sj}\|_1] \tag{33}$$

$$\begin{aligned} = \max_{\substack{\boldsymbol{Z}_r\in\mathcal{S}^{p_r}_d \\ r\neq s, r=1,\ldots,m}} \sum_{j=1}^{d}[\max_{\boldsymbol{z}_{sj}\in\mathcal{S}^{p_s}} \sum_{i=1}^{p_s} z_{sij}(\sum_{\substack{r=1\\ r\neq s}}^{m}\tilde{\boldsymbol{c}}_{rsi}^\top\boldsymbol{z}_{rj}) - \sum_{\substack{r=1\\ r\neq s}}^{m}\gamma_{srj}\|\boldsymbol{z}_{sj}\|_1]+ \\ \overbrace{\sum_{\substack{i<j=2\\ i,j\neq s}}^{m} tr(\boldsymbol{Z}_r^\top \boldsymbol{C}_{rs}\boldsymbol{Z}_s\boldsymbol{N}) - \sum_{j=1}^{d}\sum_{\substack{i=1\\ i\neq s}}^{m}\sum_{\substack{r=1\\ i\neq r}}^{i-1}\gamma_{irj}\|\boldsymbol{z}_{ij}\|_1}^{I} \end{aligned} \tag{34}$$

$$= \max_{\substack{\boldsymbol{Z}_r\in\mathcal{S}^{p_r}_d \\ r\neq s, r=1,\ldots,m}} \sum_{j=1}^{d}[\max_{\boldsymbol{z}_{sj}\in\mathcal{S}^{p_s}} \sum_{i=1}^{p_s} |z'_{sij}|(|\sum_{\substack{r=1\\ r\neq s}}^{m}\tilde{\boldsymbol{c}}_{rsi}^\top\boldsymbol{z}_{rj}| - \sum_{\substack{r=1\\ r\neq s}}^{m}\gamma_{srj})] + I \tag{35}$$

where the last line follows from $z_{sij} = sgn(\sum_{\substack{r=1\\ r\neq s}}^{m}\tilde{\boldsymbol{c}}_{rsi}^\top\boldsymbol{z}_r)z'_{sij}$. $\tilde{\boldsymbol{c}}_{rsi} = \boldsymbol{c}_{rsi}$ if $r < s$, and $\tilde{\boldsymbol{c}}_{rsi} = \boldsymbol{c}_{rsi}^\top$ if $r > s$ where $\boldsymbol{c}_{rsi}$ is the ith row of $\boldsymbol{C}_{rs} = 1/n\boldsymbol{X}_r^T\boldsymbol{X}_s$. Now solving for $\boldsymbol{z}'_{sj}$ and translating back to $\boldsymbol{z}_{sj}$ and normalizing, we get the solution in 12. Substituting this solution back to 35,

$$\begin{aligned} \phi^{2m}_{l_1,d}(\boldsymbol{\Gamma}_1,\ldots,\boldsymbol{\Gamma}_d) = \max_{\substack{\boldsymbol{Z}_r\in\mathcal{S}^{p_r}_d \\ r\neq s, r=1,\ldots,m}} \sum_{j=1}^{d}\sum_{i=1}^{p_s}[\mu_j|\sum_{\substack{r=1\\ r\neq s}}^{m}\tilde{\boldsymbol{c}}_{rsi}^\top\boldsymbol{z}_{rj}| - \sum_{\substack{r=1\\ r\neq s}}^{m}\gamma_{srj}]_+^2 + \\ \sum_{\substack{i<r=2\\ i,r\neq s}}^{m} tr(\boldsymbol{Z}_i^\top \boldsymbol{C}_{ir}\boldsymbol{Z}_r\boldsymbol{N}) - \sum_{j=1}^{d}\sum_{\substack{i=1\\ i\neq s}}^{m}\sum_{\substack{r=1\\ i\neq j}}^{s-1}\gamma_{irj}\|\boldsymbol{z}_{ij}\|_1 \end{aligned} \tag{36}$$

■

A.5 Proof of Corollary 6

Proof Utilizing the results in Equation 12,

$$[\boldsymbol{Z}_s]^*_{ij} = 0 \Leftrightarrow [\mu_j|\sum_{\substack{r=1\\ r\neq s}}^{m}\tilde{\boldsymbol{c}}_{rsi}^\top\boldsymbol{z}_{rj}| - \sum_{\substack{r=1\\ r\neq s}}^{m}\gamma_{srj}]_+ = 0 \Leftrightarrow \mu_j|\sum_{\substack{r=1\\ r\neq s}}^{m}\tilde{\boldsymbol{c}}_{rsi}^\top\boldsymbol{z}_{rj}| \leq \sum_{\substack{r=1\\ r\neq s}}^{m}\gamma_{srj} \tag{37}$$

and as before we can identify a more general sufficient condition regardless of $\boldsymbol{Z}_r^*$,

$$\mu_j|\sum_{\substack{r=1\\r\neq s}}^{m}\tilde{\boldsymbol{c}}_{rsi}^\top \boldsymbol{z}_{rj}| \leq \mu_j \sum_{\substack{r=1\\r\neq s}}^{m} \|\tilde{\boldsymbol{c}}_{rsi}\|_2\|\boldsymbol{z}_{rj}\|_2 = \sum_{\substack{r=1\\r\neq s}}^{m}\|\tilde{\boldsymbol{c}}_{rsi}\|_2 \tag{38}$$

Hence, $[\boldsymbol{T}_s]_{ij} = 0$ if $\sum_{\substack{r=1\\r\neq s}}^{m}\|\tilde{\boldsymbol{c}}_{rsi}\|_2 \leq \sum_{\substack{r=1\\r\neq s}}^{m}\gamma_{srj}$ regardless of $\boldsymbol{Z}_r^*$. ■

A.6 Proof of Theorem 7

Proof

$$\begin{aligned}\phi_{l_1,d}(\boldsymbol{\gamma}_1,\boldsymbol{\gamma}_2) &= \max_{\boldsymbol{Z}_1\in\mathcal{S}_d^{p_1}}\sum_{j=1}^{d}\max_{\boldsymbol{z}_{2j}\in\mathcal{S}^{p_2}}[\sum_{i=1}^{p_2} z_{2ji}\mu_j(\boldsymbol{c}_i^\top \boldsymbol{z}_{1j} + \epsilon_{2j}\boldsymbol{x}_{2i}^\top \boldsymbol{y}_j) - \gamma_{2j}\|\boldsymbol{z}_{2j}\|_1] \\ &+ \sum_{j=1}^{d}(\mu_j\epsilon_{1j}\boldsymbol{y}_j^\top \boldsymbol{X}_1\boldsymbol{z}_{1j} - \gamma_{1j}\|\boldsymbol{z}_{1j}\|_1) \\ &= \max_{\boldsymbol{Z}_1\in\mathcal{S}_d^{p_1}}\sum_{j=1}^{d}\max_{\boldsymbol{z}_{2j}\in\mathcal{S}^{p_2}}[\sum_{i=1}^{p_2}|z'_{2ji}|(\mu_j|\boldsymbol{c}_i^\top \boldsymbol{z}_{1j} + \epsilon_{2j}\boldsymbol{x}_{2i}^\top \boldsymbol{y}_j| - \gamma_{2j})] \\ &+ \sum_{j=1}^{d}(\mu_j\epsilon_{1j}\boldsymbol{y}_j^\top \boldsymbol{X}_1\boldsymbol{z}_{1j} - \gamma_{1j}\|\boldsymbol{z}_{1j}\|_1)\end{aligned} \tag{39}$$

Similar to Theorem 2, $z_{2ji} = sgn(\boldsymbol{c}_i^\top \boldsymbol{z}_{1j} + \epsilon_{2j}\boldsymbol{x}_{2i}^\top \boldsymbol{y}_j)z'_{2ji}$. Maximizing over z'_{2ji} while keeping $\boldsymbol{z}_{1j}$ constant and transforming back to z_{2ji}, we obtain Equation 19. Substituting the result back in Program 22 we obtain the following optimization program,

$$\phi_{l_1,d}(\boldsymbol{\gamma}_1,\boldsymbol{\gamma}_2) = \max_{\boldsymbol{Z}_1\in\mathcal{S}_d^{p_1}}\sum_{j=1}^{d}\{\sum_{i=1}^{p_2}[\mu_j|\boldsymbol{c}_i^\top \boldsymbol{z}_{1j} + \epsilon_{2j}\boldsymbol{x}_{2i}^\top \boldsymbol{y}_j| - \gamma_{2j}]_+^2 + \mu_j\epsilon_{1j}\boldsymbol{y}_j^\top \boldsymbol{X}_1\boldsymbol{z}_{1j} - \gamma_{1j}\|\boldsymbol{z}_{1j}\|_1\} \tag{40}$$

■

A.7 Proof of Corollary 8

Proof Per Equation 19,

$$z_{2ij} = 0 \Leftrightarrow [\mu_j|\boldsymbol{c}_k^\top \boldsymbol{z}_{1j} + \epsilon_{2j}\boldsymbol{x}_{2k}^\top \boldsymbol{y}_j| - \gamma_{2j}]_+ = 0 \Leftrightarrow |\boldsymbol{c}_k^\top \boldsymbol{z}_{1j}^* + \epsilon_{2j}\boldsymbol{x}_{2k}^\top \boldsymbol{y}_j| \leq \gamma_{2j}/\mu_j \tag{41}$$

More generally in order for $[\boldsymbol{T}_2]_{ij} = 0$, it is sufficient to have $\|\boldsymbol{c}_i^\top\|_2 \leq \gamma_{2j}/\mu_j$ since $|\boldsymbol{c}_k^\top \boldsymbol{z}_{1j}^* + \epsilon_{2j}\boldsymbol{x}_{2k}^\top \boldsymbol{y}_j| \leq \|\boldsymbol{c}_i\|_2\|\boldsymbol{z}_{1j}^*\|_2 + \epsilon_{2j}\|\boldsymbol{x}_{2k}\|_2\|\boldsymbol{y}_j\|_2 = \|\boldsymbol{c}_i\|_2 + \epsilon_{2j}\|\boldsymbol{x}_{2k}\|_2$ assuming $\boldsymbol{y}_j$ is normalized. ■

Appendix B. Algorithms

B.1 First Order Optimization Method

Algorithm 5: A first-order optimization method.

```
Data: $z_0 \in \mathcal{Q}$
Result: $z_k^* = \arg\max_{z \in \mathcal{Q}} f(z)$
$k \leftarrow 0$
while convergence criterion is not met do
    $z_{k+1} \leftarrow \arg\max_{x \in \mathcal{Q}}(f(z_k) + (x - z_k)^T f'(z_k))$
    $k \leftarrow k + 1$
```

B.2 L_0 Regularized Algorithm

Now we apply our first-order maximization algorithm to Program 31,

Algorithm 6: `BLOCCS` algorithm for solving Program 31

```
Data: Sample Covariance Matrix $\boldsymbol{C}_{12}$
      Regularization parameter vector $\boldsymbol{\gamma_2} \in [0,1]^d$
      Initialization $\boldsymbol{Z}_1 \in \mathcal{S}_d^{p_1}$
      $\boldsymbol{N} = diag(\mu_1, \dots, \mu_d) \succ 0$
      (optional) $\boldsymbol{T}_1 \in \{0,1\}^{p_1 \times d}$
Result: $\boldsymbol{T}_2$, optimal sparsity pattern of $\boldsymbol{Z}_2^*$
initialization;
while convergence criterion is not met do
    for $j = 1, \dots, d$ do
        $\boldsymbol{z}_{1j} \leftarrow \sum_{i=1}^{p_2} \mu_j^2 [(\mu_j \boldsymbol{c}_i^\top \boldsymbol{z}_1)^2 - \gamma_2]_+ \boldsymbol{c}_i^\top \boldsymbol{z}_1 \boldsymbol{c}_i$
    $\boldsymbol{Z}_1 \leftarrow polar(\boldsymbol{Z}_1)$
    if $\boldsymbol{T}_1$ is given then
        $\boldsymbol{Z}_1 \leftarrow \boldsymbol{Z}_1 \circ \boldsymbol{T}_1$
Output $\boldsymbol{T}_2 \in \{0,1\}^{p_2 \times d}$ where $[\boldsymbol{T}_2]_{ij} = 0$ if $(\boldsymbol{c}_i^\top \boldsymbol{z}_{1j}^*)^2 \leq \gamma_{2j}/\mu_j^2$ and 1 otherwise.
```

There won't be a second stage here, since finding $\boldsymbol{T}_i$ is the final goal.

B.3 Active Entry Estimation For Multi-Modal sCCA

$$\phi_{d,0}^m = \max_{\substack{\boldsymbol{Z}_r \in \mathcal{S}_d^{p_r}, r=1,\dots,m \\ \boldsymbol{Z}_r|_{\neq 0} = \boldsymbol{T}_r}} \sum_{r<s=2}^{m} tr(\boldsymbol{Z}_r^\top \boldsymbol{C}_{rs} \boldsymbol{Z}_s \boldsymbol{N}) \tag{42}$$

Algorithm 7: BLOCCS algorithm for solving Program 42

```
Data: Sample Covariance Matrices $\boldsymbol{C}_{rs}, \quad 1 \leq r < s \leq m$
      Initial values $\boldsymbol{Z}_r \in \mathcal{S}_d^{p_r}, \quad 1 \leq r \leq m$
      $\boldsymbol{N} = diag(\mu_1, \ldots, \mu_d) \succ 0$
      $\boldsymbol{T}_r \in \{0,1\}^{p_r \times d}, r \neq s$
Result: $\boldsymbol{Z}_i^*, i = 1, \ldots, m$, local maximizers of 42
initialization;
while convergence criterion is not met do
    for $s = 1, \ldots, m$ do
        $\boldsymbol{Z}_s \to polar(\sum_{r=1, r \neq s}^{m} \tilde{\boldsymbol{C}}_{rs}^\top \boldsymbol{Z}_r \boldsymbol{N})$
        $\boldsymbol{Z}_s \to \boldsymbol{Z}_s \circ \boldsymbol{T}_s$
```

B.4 Active Entry Estimation For Directed sCCA

We estimate active entries of the canonical directions in the second stage via the following maximization program,

$$\phi_{d,0} = \max_{\substack{\boldsymbol{Z}_1 \in \mathcal{S}_d^{p1}, \boldsymbol{Z}_1|_{\neq 0} = \boldsymbol{T}_1 \\ \boldsymbol{Z}_2 \in \mathcal{S}_d^{p2}, \boldsymbol{Z}_2|_{\neq 0} = \boldsymbol{T}_2}} tr(\boldsymbol{Z}_1^\top \boldsymbol{C}_{12} \boldsymbol{Z}_2 \boldsymbol{N}) + \sum_{i=1}^{2} tr(\boldsymbol{Y}^\top \boldsymbol{X}_i \boldsymbol{Z}_i \boldsymbol{N} \boldsymbol{E}_i) \tag{43}$$

Algorithm 8: BLOCCS algorithm for solving Program 43

```
Data: Sample Covariance Matrix $\boldsymbol{C}_{12}$
      Initialization $\boldsymbol{Z}_i \in \mathcal{S}_d^{p_i}$ ,$i = 1, 2$
      $\boldsymbol{N} = diag(\mu_1, \ldots, \mu_d) \succ 0$
      $\boldsymbol{T}_i \in \{0,1\}^{p_i \times d}$, $i = 1, 2$
      $\boldsymbol{E}_i = diag(\boldsymbol{\epsilon_i}), i = 1, 2$
Result: $\boldsymbol{Z}_i^*, i = 1, 2$, local maximizers of 43
initialization;
while convergence criterion is not met do
    $\boldsymbol{Z}_2 \to polar(\boldsymbol{C}_{12}^\top \boldsymbol{Z}_1 \boldsymbol{N} + \boldsymbol{X}_2^\top \boldsymbol{Y} \boldsymbol{N} \boldsymbol{E}_2) \circ \boldsymbol{T}_2$
    $\boldsymbol{Z}_1 \to polar(\boldsymbol{C}_{12} \boldsymbol{Z}_2 \boldsymbol{N} + \boldsymbol{X}_1^\top \boldsymbol{Y} \boldsymbol{N} \boldsymbol{E}_1) \circ \boldsymbol{T}_1$
```

B.5 Hyper-parameter Tuning Using Permutation Test

```
Algorithm 9: Hyperparameter Tuning via Permutation Test
  Data: Sample matrices $\boldsymbol{X}_i \in \mathbb{R}^{n \times p_i}$, $i = 1, 2$
        Sparsity parameters $\gamma_i$, $i = 1, 2$
        Initial values $\boldsymbol{z}_i \in \mathcal{S}^{p_i}$, $i = 1, 2$
        Number of permutations $P$
  Result: $p_{\gamma_1,\gamma_2}$ the evidence against the null hypothesis that the canonical correlation is
          not lower when $X_i$ are independent.
  Compute $(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*)$ on $\boldsymbol{X}_1, \boldsymbol{X}_2$ via any of the proposed algorithms with sparsity
    hyperparameters $(\gamma_1, \gamma_2)$
  $\rho(\gamma_1, \gamma_2) = corr(\boldsymbol{X}_1\boldsymbol{z}_1^*, \boldsymbol{X}_2\boldsymbol{z}_2^*)$
  for $p$ = 1, ..., $P$ do
      Let $\boldsymbol{X}_1^{(p)}$ be a row-wise permutation of $\boldsymbol{X}_1$
      Compute $(\boldsymbol{z}_1^{*(p)}, \boldsymbol{z}_2^{*(p)})$ on $(\boldsymbol{X}_1^{(p)}, \boldsymbol{X}_2)$ via any of the proposed algorithms with sparsity
        hyperparameters $(\gamma_1, \gamma_2)$
      $\rho_{perm}^{(p)}(\gamma_1, \gamma_2) = corr(\boldsymbol{X}_1^{(p)}\boldsymbol{z}_1^{*(p)}, \boldsymbol{X}_2\boldsymbol{z}_2^{*(p)})$
  $p_{\gamma_1,\gamma_2} = 1/P \sum_{p=1}^{P} I(\rho_{perm}^{(p)} > \rho)$
```

References

G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. *International Conference on Machine Learning*, pages 1247–1255, 2013.

Matthew B Blaschko, Jacquelyn A Shelton, Andreas Bartels, Christoph H Lampert, and Arthur Gretton. Semi-supervised kernel canonical correlation analysis with application to human fmri. *Pattern Recognition Letters*, 32(11):1572–1583, 2011.

Kumardeep Chaudhary, Olivier B Poirion, Liangqun Lu, and Lana X Garmire. Deep learning–based multi-omics integration robustly predicts survival in liver cancer. *Clinical Cancer Research*, 24 (6):1248–1259, 2018.

Ng M. K. Zhang X. Chu D., Liao L. Sparse canonical correlation analysis: New formulation and algorithm. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 35, 2013.

Sandrine Dudoit, Jane Fridlyand, and Terence P Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, 97(457):77–87, 2002.

C.E. Hopkins. Statistical analysis by canonical correlation: a computer application. *Health services research*, 4(4):304, 1969.

H. Hotelling. The most predictable criterion. *Journal of Educational Psychology*, 26:139–142, 1935.

Hua Huang, Huiting He, Xin Fan, and Junping Zhang. Super-resolution of human face image using canonical correlation analysis. *Pattern Recognition*, 43(7):2532–2543, 2010.

M. Journée, Y. Nesterov, P. Richtrárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11:517–553, 2010.

P.L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10:365–377, 2000.

Yifeng Li, Fang-Xiang Wu, and Alioune Ngom. A review on machine learning principles for multi-view biological data integration. *Briefings in bioinformatics*, 19(2):325–340, 2016.

M.S. Monmonier and F.E. Finn. Improving the interpretation of geographical canonical correlation models. *The Professional Geographer*, 25:140–142, 1973.

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

Omid S. Solari, James P. Duncan, Rojin Safavi, James B. Brown, and Peter J. Bickel. Sparkle: A generalized spark-based sparse kernel multi-view learning framework. *arXiv preprint arXiv:1206.3242*, 2019.

D. Simonson, J. Stowe, and C. Watson. A canonical correlation analysis of commercial bank asset/liability structures. *Journal of Financial and Quantitative Analysis*, 10:125–140, 1983.

Omid S Solari, James B Brown, and Peter J Bickel. Sparse canonical correlation analysis via concave minimization. *arXiv preprint arXiv*, 2019.

X.M. Tu, D.S. Burdick, D.W. Millican, and L.B. McGown. Canonical correlation technique for rank estimation of excitation-emission matrices. *Analytical Chemistry*, 19(61):2219–2224, 1989.

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

Ying-Wooi Wan, Genevera I Allen, and Zhandong Liu. Tcga2stat: simple tcga data access for integrated statistical analysis in r. *Bioinformatics*, 32(6):952–954, 2015.

John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M Stuart, Cancer Genome Atlas Research Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10): 1113, 2013.

D. Witten and R. Tibshirani. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical Applications in Genomics and Molecular Biology*, 8, 2009.

K.W. Wong, P.C.W. Fung, and C.C. Lau. Study of the mathematical approximations made in the basis correlation method and those made in the canonical-transformation method for an interacting bose gas. *Physical Review*, 3(22):1272, 1980.

Chapter 3

Sparse Multi-View Multiple Kernel Learning with Application to Large-Scale Interpretable Kernel Canonical Correlation Analysis

Abstract

A large-scale sparse kernel canonical correlation analysis approach is introduced, with the aim of inferring non-linear, yet interpretable, associations between multiple sets of high-dimensional covariates from observations on matching subjects, termed *view* in this article. Our model learns the association structures in two stages; in the first stage, we solve a regularized *Multiple Kernel Learning (MKL)* problem where we learn a convex sparse combination of kernel basis functions, whether feature-wise kernels or pair-wise feature interaction kernels, via a regularized *Hilbert-Schmidt Independence Criterion (HSIC)* maximization, and in the second stage, we solve a regular KCCA problem between the learned kernels from the first stage. Our main contribution is our sparse MKL approach where, inspired by the approach of Solari et al. (2019), we cast the HSIC maximization program as a concave minimization program which we solve via a simple first order method; this results in a significantly faster algorithm with remarkable convergence properties mainly due to the drastic shrinkage of our search space from a euclidean ball to a sphere. We establish superior empirical consistency and accuracy of our approach compared to the more common alternating maximization based MKL approaches, e.g. Yoshida et al. (2017), via extensive high-dimensional simulations. Our approach is specifically appealing where sub-kernels are very large, e.g. if pair-wise kernel is used in feature space. A comprehensive `spark` implementation of our method, called SparKLe, is provided making it possible to handle large-scale problems which results in even further speed-up. Finally we reaffirm the superior speed of our package compared to its rivals empirically.

1. Introduction

Advances in data analytic and information retrieval methods has made it possible for scientists to collect different types of information in a single research problem, making it possible to observe the same phenomena from multiple views, which enables them to use multiple approaches to make different inferences in a single experiment.

While conventional, i.e. single view, learning methods concatenate all different views together in one bigger feature set, optimizing over one set of parameters, multi-view learning algorithms aim to optimize over one set of parameters per view at the same time. In this modeling paradigm we are aware of the heterogeneity and possible redundancy of different views, which can prevent overfitting in low sample size settings provide robustness to corrupt views. Multi-view models also provide valuable information regarding the inter-dependence of features from multiple views which can be used to design more specific and targeted follow-up experiments. In essence, considering different models for different views with different physical meaning and statistical interpretation, and fitting them jointly provides us with more interpretable models which are more robust to view redundancy[Cui et al. (2007)], low-sample size[Xu et al. (2013)] and corrupt input[Christoudias et al. (2012)].

Perhaps the most celebrated modeling paradigm within multi-view learning is *Sub-space Learning*. In this approach, each observed view is assumed to be sampled from a posterior distribution where parameters contain latent variables shared between different views and that these multiple views are generated via a transformation applied to the observed latent variables. Having observed multiple transformed latent variables, the objective is to simultaneously estimate the parameters of the inverse transforms. Understandably, according to the assumed generative model, the notion of ”optimality” applies here to sets of parameters that form inverse transforms whose samples, i.e. latent variable samples, are as similar as possible using some measure of similarity, e.g. correlation.

A classic descendant of this generative framework and also the first multi-view learning algorithm is *Canonical Correlation Analysis*, due Hotelling (1935), which formed by limiting the inverse transforms to be from the family of linear functions and using linear correlation as the measure of similarity. Due to the simple form of linear transforms and their geometric interpretation, these models are more interpretable than other non-linear transforms. These transforms try to map two observed views to lower-dimensional convex polytopes while maximizing their Euclidean dot product. In high-dimensional settings, to avoid overfitting and improve interpretability, regularization terms are added which results in sparse additive linear transforms. These models are usually trained using alternating optimization of a bi-convex objective, e.g. Parkhomenko et al. (2009), Witten and Tibshirani (2009), Waaijenborg et al. (2008), however Solari et al. (2019) recently presented a power method solution with better convergence characteristics as an alternative.

The desire to discover more complex dependence relationships between multiple random vectors has compelled researchers to explore beyond the class of linear functions. Hsieh (2000) pioneered using neural networks in multi-view learning and Bach and Jordan (2002) used the *kernel trick* to extend the multi-view learning paradigm to non-linear transforms. Ever since, there has been substantial development in

After its proposition by Hotelling (1935), CCA was first applied in Waugh (1942) where he studied the relationship between the characteristics of wheat and the resulting flour. *CCA* and its variants, especially non-linear extensions, have since been used in various fields of data science and machine learning with successful application in finance[Simonson et al. (1983)], signal processing[Schell and Gardner (1995)], neuro-science esp. neuro-imaging[Friman et al. (2001) and Kay et al. (2008)], image processing and object recognition[Covell and Slaney (2002)], NLP[Faridani (2011)], social sciences[Hopkins (1969)], urban development and city planning[Monmonier and Finn (1973)], astronomy[Dainotti et al. (2010)], chemistry[Tu et al. (1989)], physics[Wong et al. (1980)], dentistry[Lindsey et al. (1985)] and recently popular multi-omics population studies[Tini et al. (2017)], where they are utilized with the aim of discovering complex yet meaningful dependence structures between two sets of variables. What follows is organized as below,

In Section 2 a new sparse MKL approach is presented where *Kernel Selection* is performed via constrained *Hillbert-Schmidt Independence Criterion* maximization which is cast as a penalized matrix decomposition for which we advise a new scalable solution. These sparse convex combinations provide variable selection based on non-linear mappings of features or groups of features. We then propose a first-order algorithm to solve the advised optimization program. Section 5 offers a glimpse into `SparKLe`, the Spark implementation of algorithms offered in this paper and the sparse CCA algorithm of Solari et al. (2019). In Section 6, we compare `SparKLe`'s performance to few other CCA and sCCA algorithms and also to few KCCA and sKCCA algorithms, e.g. `pyrcca`[Bilenko and Gallant (2016)] and `TSKCCA`[Yoshida et al. (2017)]. More discussion material are provided in the appendices and supplement sections and are referenced in the text wherever applicable, But first we lay the foundations for our method in the remainder of this section.

1.1 The "Sub-Space Learning" Paradigm

In this paradigm, given the views $X_i \in \mathcal{X}_i$, $i = 1, \dots, m$, and functional families $\mathcal{F}_i = \{f_i : \mathcal{X}_i \to \mathcal{E}\}$, the main goal is to estimate functions $f_i \in \mathcal{F}_i$, $i = 1, \dots, m$ such that transformed views $f_i(X_i)$ minimize some distance criterion $\mathcal{D}(f_1(X_1), \dots, f_m(X_m))$ or maximize some similarity criterion $\mathcal{S}(f_1(X_1), \dots, f_m(X_m))$ respectively.

$$\boldsymbol{f}^* = \underset{\substack{f_i \in \mathcal{F}_i \\ i \in \{1,\dots,m\}}}{\arg\max}\ \mathcal{S}(f_1(X_1), \dots, f_m(X_m)) \tag{1}$$

Where $\boldsymbol{f} = (f_1, \dots, f_m)$.

1.1.1 CCA

In a setting with only a pair of views, $m = 2$, if we assert the functional families $\mathcal{F}_i$ to be a subset of the parametric family of linear functions $\mathcal{L} = \{l_i : \mathbb{R}^{p_i} \to \mathbb{R}^k, l_i(X_i) = \boldsymbol{z}_i X_i\}$, and the similarity criterion to be the Pearson correlation, we end up with the *Canonical Correlation Analysis* criterion. Assuming $E[X_1] = \mathbf{0}^{p_1}$ and $E[X_2] = \mathbf{0}^{p_2}$,

$$\begin{aligned}(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*) &= \underset{\boldsymbol{z}_1 \in \mathbb{R}^{p_1}, \boldsymbol{z}_2 \in \mathbb{R}^{p_2}}{\arg\max}\ \rho(X_1\boldsymbol{z}_1, X_2\boldsymbol{z}_2) \\ &= \underset{\boldsymbol{z}_1 \in \mathbb{R}^{p_1}, \boldsymbol{z}_2 \in \mathbb{R}^{p_2}}{\arg\max}\ \frac{E[(X_1\boldsymbol{z}_1)^\top (X_2\boldsymbol{z}_2)]}{E[(X_1\boldsymbol{z}_1)^2]^{1/2} E[(X_2\boldsymbol{z}_2)^2]^{1/2}}\end{aligned} \tag{2}$$

Since almost always we just have access to samples from X_1 and X_2, we estimate Program 2 using plug-in sample estimators for population terms.

$$\begin{aligned}(\boldsymbol{z}_1^*, \boldsymbol{z}_2^*) &= \underset{\boldsymbol{z}_1 \in \mathbb{R}^{p_1}, \boldsymbol{z}_2 \in \mathbb{R}^{p_2}}{\arg\max} \; \hat{\rho}(\boldsymbol{X}_1 \boldsymbol{z}_1, \boldsymbol{X}_2 \boldsymbol{z}_2) \\ &= \underset{\boldsymbol{z}_1 \in \mathbb{R}^{p_1}, \boldsymbol{z}_2 \in \mathbb{R}^{p_2}}{\arg\max} \; \frac{\boldsymbol{z}_1^\top \boldsymbol{X}_1^\top \boldsymbol{X}_2 \boldsymbol{z}_2}{\sqrt{\boldsymbol{z}_1^\top \boldsymbol{X}_1^\top \boldsymbol{X}_1 \boldsymbol{z}_1} \sqrt{\boldsymbol{z}_2^\top \boldsymbol{X}_2^\top \boldsymbol{X}_2 \boldsymbol{z}_2}}\end{aligned} \tag{3}$$

1.1.2 KCCA & The Kernel Trick

Now let's choose $\mathcal{F}_i$ to be the family of linear combinations of feature maps $\phi_i(\boldsymbol{x}_i) : \mathcal{X}_i \to \mathcal{E}$, and $k_i(\boldsymbol{x}_i, \boldsymbol{x}_i') =< \boldsymbol{\phi}_i(\boldsymbol{x}_i), \boldsymbol{\phi}_i(\boldsymbol{x}_i') >_{\mathcal{V}}$: $\mathcal{X}_i \times \mathcal{X}_i \to \mathbb{R}$ to be the *inner product kernel* associated with an inner product space $\mathcal{V}$ and $[\boldsymbol{K}_i]_{rs} = k_i(\boldsymbol{x}_{ir}, \boldsymbol{x}_{is})$ be the associated kernel matrix. Then, a reformulation of Program 3 is,

$$(\boldsymbol{\alpha}_1^*, \boldsymbol{\alpha}_2^*) = \underset{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \in \mathbb{R}^n}{\arg\max} \; \hat{\rho}(\boldsymbol{K}_1 \boldsymbol{\alpha}_1, \boldsymbol{K}_2 \boldsymbol{\alpha}_2) = \underset{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \in \mathbb{R}^n}{\arg\max} \; \frac{\boldsymbol{\alpha}_1^\top \boldsymbol{K}_1^\top \boldsymbol{K}_2 \boldsymbol{\alpha}_2}{\sqrt{\boldsymbol{\alpha}_1^\top \boldsymbol{K}_1^\top \boldsymbol{K}_1 \boldsymbol{\alpha}_1} \sqrt{\boldsymbol{\alpha}_2^\top \boldsymbol{K}_2^\top \boldsymbol{K}_2 \boldsymbol{\alpha}_2}} \tag{4}$$

Due to rank-deficiency, $\boldsymbol{K}_i' = \boldsymbol{K}_i + \kappa \boldsymbol{I}$ is usually used in the denominator instead of $\boldsymbol{K}_i$.

In practice, we don't need to explicitly define mappings $\boldsymbol{\phi}_i$ in order to form kernel matrices by computing their inner products. We can bypass this stage as long as we can choose an inner-product kernel $k_i : \mathbb{R}^{p_i} \times \mathbb{R}^{p_{i'}} \to \mathbb{R}$ which satisfies the *Mercer's Condition* as prescribed in *Mercer's Theorem* in Mercer (1909). This idea is what is commonly known as the *Kernel Trick*. This theorem guarantees the existence of a mapping $\boldsymbol{\phi}_i$ corresponding to the kernel k_i iff,

$$\int k(x, x') g(x) g(y) dx dy \geq 0 \quad \forall g$$

That is, it is a positive semi-definite kernel.

For a comprehensive review of different methods of solving CCA and KCCA problem, we refer you to Hardoon et al. (2004).

1.2 Kernel Learning

Kernel methods traditionally require kernels to be specified. This choice, which affects the success of learning, used to be performed manually through trial and error. In order to avoid excruciating kernel engineering and to create kernels which serve the learning task at hand, *Kernel Learning* methods were developed for classification and regression problems[Lanckriet et al. (2004), Bach et al. (2004)], where feature space is represented via sets of base kernels and functions of these base kernels are learned rather than deterministically choosing the kernels used in classification/regression from the beginning. Specifically, rather than specifying a kernel, a kernel family, here $\mathcal{K} = \{k : \mathbb{R}^{p_i} \times \mathbb{R}^{p_i'} \to \mathbb{R}\}$, is specified. By restricting to specific families of kernels, the optimization becomes tractable, and we can simultaneously learn useful kernels along with their support in terms of features in each view. Our goal in subsequent sections will be to learn linear combinations of base kernels, each of which is sparse in each view, that explain and make evident correlation between views. Inner product kernels, $k_i(\boldsymbol{x}_i, \boldsymbol{x}_i')$, are modeled as a function of a finite number of base kernels $\{k_i^{(q)}\}_{q=1}^{Q_i} \in \mathcal{K}$, that is $k_i = h(k_i^{(1)}, \dots, k_i^{(Q_i)})$. As just two examples of possible families of base kernels $k_i^{(q)}$ may be feature-wise mappings, consequently $Q_i = p_i$, pair-wise interaction kernels, consequently $Q_i = p_i(p_i + 1)/2$, etc. Our objective in kernel learning is to learn h_i. Here we choose this function to be a convex linear combination.

$$k_i(\boldsymbol{x}_i, \boldsymbol{x}_i') = \sum_{q=1}^{Q_i} \zeta_{iq} k_i^{(q)}(\boldsymbol{x}_i, \boldsymbol{x}_i') \qquad \boldsymbol{\zeta}_i \geq \boldsymbol{0} \tag{5}$$

where $\boldsymbol{\zeta}_i \geq 0$ to ensure that $k_i(\boldsymbol{x}_i, \boldsymbol{x}_i')$ is a positive semi-definite kernel function. Cristianini et al. (2002), Cortes et al. (2010), and Yoshida et al. (2017) utilize this paradigm in *Two-Stage Kernel Learning*. Since base kernel functions are almost always a function of a small selection of the covariates, often one or two covariates, their learned sparse combinations, $\boldsymbol{\zeta}_i$, provide interpretable non-linear association structures of the observed covariates.

1.3 Nonlinear Kernel Alignment Criteria

Parameters $\boldsymbol{\zeta}_i \in \mathbb{R}^{Q_i}, i = 1, \dots, m$, are estimated such that they maximize some similarity measure between kernels $\{k_i\}_{i=1}^m$,

$$\{\boldsymbol{\zeta}_1^*, \dots, \boldsymbol{\zeta}_m^*\} = \underset{\substack{\boldsymbol{\zeta}_i \in \mathbb{R}^{Q_i}, i=1,\dots,m \\ \boldsymbol{\zeta}_i \geq \boldsymbol{0}}}{\arg\max} \; \mathcal{S}(\boldsymbol{\zeta}_1^\top \boldsymbol{k}_1, \dots, \boldsymbol{\zeta}_m^\top \boldsymbol{k}_m) \qquad \boldsymbol{k}_i \in \mathbb{R}^{Q_i} \tag{6}$$

where $k_i(\boldsymbol{x}_i, \boldsymbol{x}_i') = \boldsymbol{\zeta}_i^\top \boldsymbol{k}_i$. In the following we review two commonly used non-linear kernel similarity criteria applied to only two views.

1.3.1 Hilbert-Schmidt Independence Criterion(HSIC)

Gretton et al. (2005) proposed an independence criterion based on the eigenspectrum of covariance operators in reproducing kernel Hilbert spaces $\mathbb{H}_1$ and $\mathbb{H}_2$ with an empirical estimate of the Hilbert-Schmidt norm of the cross-covariance operator,

$$H\hat{S}IC(\boldsymbol{X}_1, \boldsymbol{X}_2, \mathbb{H}_1, \mathbb{H}_2) = \frac{1}{(n-1)^2} tr(\boldsymbol{K}_1 \boldsymbol{H} \boldsymbol{K}_2 \boldsymbol{H}) \tag{7}$$

Where $\boldsymbol{K}_1, \boldsymbol{K}_2, \boldsymbol{H} \in \mathbb{R}^{n \times n}$, $[\boldsymbol{H}]_{ij} = \delta_{ij} - 1/n$, and $\boldsymbol{K}_i$ are the inner-product kernel matrices for each view.

1.3.2 Kernel Target Alignment(KTA)

Given kernel functions $k_i, i = 1, 2$ where the first two moments are available, Cristianini et al. (2002) formalizes the notion of alignment between the two kernels by defining *Kernel Target Alignment* as,

$$\rho_a(k_1, k_2) = \frac{E[k_1 k_2]}{E[k_1^2]^{1/2} E[k_2^2]^{1/2}} \tag{8}$$

an empirical estimator of which using kernel matrices is given by,

$$\hat{\rho}_a(\boldsymbol{K}_1, \boldsymbol{K}_2) = \frac{tr(\boldsymbol{K}_1\boldsymbol{K}_2)}{tr(\boldsymbol{K}_1^2)^{1/2} tr(\boldsymbol{K}_2^2)^{1/2}} \tag{9}$$

Cortes et al. (2010) offers a comprehensive discussion on KTA maximization. We use HSIC as the similarity measure in `SparKLe`.

2. Two-Stage Sparse Multiple Kernel Learning via Power Iterations

Here we introduce a two-stage sparse MKL methods. Similar to Yoshida et al. (2017), we first learn a convex sparse combination of base kernels, then apply the classical KCCA to the learned kernels. Our contribution is to introduce a fast and more stable power method for learning sparse combinations of base kernels which is also extended to multiple views. We learn positive-definite symmetric kernel matrices, $\boldsymbol{K}_i, i = 1, 2$, in two stages. We first learn the sparsity patterns of $\boldsymbol{\zeta}_i, i = 1, 2$, denoted by $\boldsymbol{\tau}_i$, and in the second stage we estimate the non-zero elements of $\boldsymbol{\zeta}_i$. Throughout this paper we utilize the notion of *Centered Kernel Functions and Matrices.* Following Cortes et al. (2010), we center a kernel function $k_i(\boldsymbol{x}_i, \boldsymbol{x}_i')$ by centering it's individual mappings $\boldsymbol{\phi}_i(\boldsymbol{x}_i)$,

$$\begin{aligned} \tilde{k}_i(\boldsymbol{x}_i, \boldsymbol{x}_i') &= (\boldsymbol{\phi}_i(\boldsymbol{x}_i) - E[\boldsymbol{\phi}_i(\boldsymbol{x}_i)])^\top (\boldsymbol{\phi}_i(\boldsymbol{x}_i') - E[\boldsymbol{\phi}_i(\boldsymbol{x}_i')]) \\ &= k_i(\boldsymbol{x}_i, \boldsymbol{x}_i') - E_{\boldsymbol{x}_i}[k_i(\boldsymbol{x}_i, \boldsymbol{x}_i')] - E_{\boldsymbol{x}_i'}[k_i(\boldsymbol{x}_i, \boldsymbol{x}_i')] + E_{\boldsymbol{x}_i, \boldsymbol{x}_i'}[k_i(\boldsymbol{x}_i, \boldsymbol{x}_i')] \end{aligned} \tag{10}$$

where we denote the centered kernel function by $\tilde{k}_i$. Plugging in the sample estimators in Equation 10, the following is a sample estimator for centered kernel matrices.

$$\tilde{\boldsymbol{K}} = \boldsymbol{K} - 1/m \sum_{i=1}^{n} \boldsymbol{K}_{ij} - 1/m \sum_{j=1}^{n} \boldsymbol{K}_{ij} + 1/m^2 \sum_{i,j=1}^{n} \boldsymbol{K}_{ij} \tag{11}$$

2.1 First Stage: Sparsity Pattern Estimation

Here we propose a method for finding sparse $\boldsymbol{\zeta}_i^* \geq \boldsymbol{0}, i = 1, 2$ in Equation 5 based on *HSIC* maximization,

$$\phi_{l_1}(\boldsymbol{\gamma}) = \max_{\substack{\boldsymbol{\zeta}_1 \in \mathbb{R}^{Q_1}, \boldsymbol{\zeta}_2 \in \mathbb{R}^{Q_2} \\ \boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2 \geq \boldsymbol{0} \\ \|\boldsymbol{\zeta}_1\|_2 = \|\boldsymbol{\zeta}_2\|_2 = 1 \\ \gamma_1, \gamma_2 \geq 0}} \frac{1}{(n-1)^2} tr(\boldsymbol{K}_1 \boldsymbol{H} \boldsymbol{K}_2 \boldsymbol{H}) - \gamma_1 \boldsymbol{1} \cdot \boldsymbol{\zeta}_1 - \gamma_2 \boldsymbol{1} \cdot \boldsymbol{\zeta}_2 \tag{12}$$

where $\boldsymbol{K}_i = \sum_{q=1}^{Q_i} \zeta_{iq} \boldsymbol{K}_i^{(q)}$.

Lemma 1 *HSIC criterion, Eq. 7, is equivalent to the un-normalized kernel alignment of centered kernels proposed in Cortes et al. (2010), i.e.* $\rho_u(\boldsymbol{K}, \boldsymbol{K}') = 1/n^2 \langle \tilde{\boldsymbol{K}}, \tilde{\boldsymbol{K}}' \rangle_F$*.*

Proof We can rewrite Equation 11 as $\tilde{\boldsymbol{K}} = (\boldsymbol{I} - \boldsymbol{1}/m)\boldsymbol{K}(\boldsymbol{I} - \boldsymbol{1}/m) = \boldsymbol{HKH}$. It is straightforward to show that $\boldsymbol{H}$ is idempotent. Hence,

$$\langle \tilde{\boldsymbol{K}}, \tilde{\boldsymbol{K}}' \rangle_F = tr(\tilde{\boldsymbol{K}}\tilde{\boldsymbol{K}}') = tr(\boldsymbol{HKHK}'\boldsymbol{H}) = tr(\boldsymbol{KHK}'\boldsymbol{H}) \tag{13}$$

where we utilized $tr(\tilde{\boldsymbol{K}}\tilde{\boldsymbol{K}}') = tr(\tilde{\boldsymbol{K}}'\tilde{\boldsymbol{K}})$.

■

Proposition 2 *Optimization problem in 12 is equivalent to the following optimization problem with a strongly biconvex objective,*

$$\phi_{l_1}(\boldsymbol{\gamma}) = \max_{\substack{\boldsymbol{\zeta}_1 \in \mathbb{R}^{Q_1}, \boldsymbol{\zeta}_2 \in \mathbb{R}^{Q_2} \\ \boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2 \geq \boldsymbol{0} \\ \|\boldsymbol{\zeta}_1\|_2 = \|\boldsymbol{\zeta}_2\|_2 = 1 \\ \gamma_1, \gamma_2 \geq 0}} \boldsymbol{\zeta}_1^\top \boldsymbol{Q} \boldsymbol{\zeta}_2 - \gamma_1 \boldsymbol{1} \cdot \boldsymbol{\zeta}_1 - \gamma_2 \boldsymbol{1} \cdot \boldsymbol{\zeta}_2 \tag{14}$$

Where $\boldsymbol{Q} \in \mathbb{R}^{Q_1} \times \mathbb{R}^{Q_2}$ and,

$$[\boldsymbol{Q}]_{ij} = \frac{1}{(n-1)^2} tr(\boldsymbol{K}_1^{(i)} \boldsymbol{H} \boldsymbol{K}_2^{(j)} \boldsymbol{H}) \tag{15}$$

Proof *Proof of this proposition is provided in Appendix A.1* ■

Proposition 3 *All elements of the multiple alignment matrix are non-negative, i.e. $\forall 1 \leq i \leq Q_1, 1 \leq j \leq Q_2$, $[\boldsymbol{Q}]_{ij} \geq 0$.*

Proof Since kernel functions are assumed to satisfy *Mercer's Condition*, their sample matrices are positive semi-definite and $\tilde{\boldsymbol{K}}_1^{(i)} = \boldsymbol{\phi}_1(\boldsymbol{x_i})^\top \boldsymbol{\phi}_1(\boldsymbol{x_i})$, and $\tilde{\boldsymbol{K}}_2^{(j)} = \boldsymbol{\phi}_2(\boldsymbol{x_j})^\top \boldsymbol{\phi}_2(\boldsymbol{x}_j)$. So,

$$\begin{aligned} [\boldsymbol{Q}]_{ij} = \langle \tilde{\boldsymbol{K}}_1^{(i)}, \tilde{\boldsymbol{K}}_2^{(j)} \rangle_F = tr(\tilde{\boldsymbol{K}}_1^{(i)} \tilde{\boldsymbol{K}}_2^{(j)}) &= tr(\boldsymbol{\phi}_1(\boldsymbol{x_i})^\top \boldsymbol{\phi}_1(\boldsymbol{x_i}) \boldsymbol{\phi}_2(\boldsymbol{x_j})^\top \boldsymbol{\phi}_2(\boldsymbol{x}_j)) \\ &= tr(\boldsymbol{\phi}_1(\boldsymbol{x_i})^\top \boldsymbol{\phi}_2(\boldsymbol{x_j}) \boldsymbol{\phi}_2(\boldsymbol{x_j})^\top \boldsymbol{\phi}_1(\boldsymbol{x}_i)) \\ &= \langle \boldsymbol{\phi}_1(\boldsymbol{x_i})^\top \boldsymbol{\phi}_2(\boldsymbol{x_j}) \rangle_F^2 \geq 0 \end{aligned} \tag{16}$$

■

The following theorem provides the solution to Program 12 (or equivalently Program 14).

Theorem 4 *The solution to Program 12 is given by,*

$$\boldsymbol{\zeta}_1^* = \arg\max_{\boldsymbol{\zeta}_1 \in \mathcal{S}_+^{Q_1}} \sum_{i=1}^{Q_2} [\boldsymbol{q}_i^T \boldsymbol{\zeta}_1 - \gamma_2]_+^2 - \gamma_1 \boldsymbol{1} \cdot \boldsymbol{\zeta}_1 \tag{17}$$

and

$$\zeta_{2i}^* = \zeta_{2i}^*(\gamma_2) = \frac{[\boldsymbol{q}_i^T \boldsymbol{\zeta}_1 - \gamma_2]_+}{\sqrt{\sum_{k=1}^{Q_2} [\boldsymbol{q}_k^T \boldsymbol{\zeta}_1 - \gamma_2]_+^2}}, \quad i = 1, \ldots, Q_2. \tag{18}$$

Proof

$$\begin{aligned}
\phi_{l_1}(\boldsymbol{\gamma}) &= \max_{\boldsymbol{\zeta}_1 \in \mathcal{B}_+^{Q_1}} \max_{\boldsymbol{\zeta}_2 \in \mathcal{B}_+^{Q_2}} \boldsymbol{\zeta}_1^\top \boldsymbol{Q} \boldsymbol{\zeta}_2 - \gamma_1 \mathbf{1} \cdot \boldsymbol{\zeta}_1 - \gamma_2 \mathbf{1} \cdot \boldsymbol{\zeta}_2 \\
&= \max_{\boldsymbol{\zeta}_1 \in \mathcal{B}_+^{Q_1}} \max_{\boldsymbol{\zeta}_2 \in \mathcal{B}_+^{Q_2}} \sum_{i=1}^{Q_2} \zeta_{2i} (\boldsymbol{q}_i^\top \boldsymbol{\zeta}_1) - \gamma_2 \mathbf{1} \cdot \boldsymbol{\zeta}_2 - \gamma_1 \mathbf{1} \cdot \boldsymbol{\zeta}_1 \qquad (19) \\
&= \max_{\boldsymbol{\zeta}_1 \in \mathcal{B}_+^{Q_1}} \max_{\boldsymbol{\zeta}_2 \in \mathcal{B}_+^{Q_2}} \sum_{i=1}^{Q_2} \zeta_{2i} (\boldsymbol{q}_i^\top \boldsymbol{\zeta}_1 - \gamma_2) - \gamma_1 \mathbf{1} \cdot \boldsymbol{\zeta}_1
\end{aligned}$$

Solving for $\boldsymbol{\zeta}_2$ while keeping $\boldsymbol{\zeta}_1$ fixed, we get Equation 18. Substituting back in 19,

$$\phi_{l_1}^2(\boldsymbol{\gamma}) = \arg\max_{\boldsymbol{\zeta}_1 \in \mathcal{B}_+^{Q_1}} \sum_{i=1}^{Q_2} [\boldsymbol{q}_i^T \boldsymbol{\zeta}_1 - \gamma_2]_+^2 - \gamma_1 \mathbf{1} \cdot \boldsymbol{\zeta}_1 \qquad (20)$$

which is a maximization program of a convex objective over a convex set, which is equivalently a concave minimization program. Hence, we can shrink the search space drastically from a euclidean ball to its boundary, i.e. a sphere, as the minimum of a concave function is always located on the boundaries of the convex optimization domain, resulting in,

$$\phi_{l_1}^2(\boldsymbol{\gamma}) = \arg\max_{\boldsymbol{\zeta}_1 \in \mathcal{S}_+^{Q_1}} \sum_{i=1}^{Q_2} [\boldsymbol{q}_i^T \boldsymbol{\zeta}_1 - \gamma_2]_+^2 - \gamma_1 \mathbf{1} \cdot \boldsymbol{\zeta}_1 \qquad (21)$$

■

This theorem paves the way towards concluding the first stage of our two stage *Multiple Kernel Learning (MKL)* approach. In the following corollary we establish using which we're able to compute the sparsity pattern $\boldsymbol{\tau}_2 \in \{0,1\}^{Q_2}$ using only the solution $\boldsymbol{\zeta}^*$ of Program 21.

Corollary 5 *For any solution $\boldsymbol{\zeta}_1^*$ of Program 21 given sparsity parameters γ_i, $\tau_{2i} = 0$ iff $\boldsymbol{q}_i^\top \boldsymbol{\zeta}_1^* \leq \gamma_2$.*

Proof

According to Equation 18 of Theorem 4,

$$\zeta_{2i}^* = 0 \Leftrightarrow [\boldsymbol{q}_i^T \boldsymbol{\zeta}_1^* - \gamma_2]_+ = 0 \Leftrightarrow \boldsymbol{q}_i^T \boldsymbol{\zeta}_1^* \leq \gamma_2 \qquad (22)$$

We can go further and find a sufficient condition for $\boldsymbol{\tau}_{2i} = 0$ without solving for $\boldsymbol{\zeta}_1^*$. Consider Equation 18 once again,

$$\boldsymbol{q}_i^T \boldsymbol{\zeta}_1 \leq \|\boldsymbol{q}_i\|_2 \|\boldsymbol{\zeta}_1\|_2 = \|\boldsymbol{q}_i\|_2 \qquad (23)$$

Hence, $\zeta_{2i} = 0$ for $i \in 1, \ldots, Q_2$ if $\|\boldsymbol{q}_i\|_2 \leq \gamma_2$ without regard to $\boldsymbol{\zeta}_1^*$. ■

Now 21 is a concave minimization algorithm which we solve using a first order optimization algorithm. Once the sparsity pattern of $\boldsymbol{\zeta}_2^*$ is inferred, we shrink $\boldsymbol{Q}$ to $[\tilde{\boldsymbol{Q}}]_{ij} = \boldsymbol{Q}_{i\boldsymbol{\tau}_2^{(j)}}$, where $\boldsymbol{\tau}_2^{(j)}$ is the j-th non-zero element of $\boldsymbol{\tau}_2$. Now we apply the results in Theorem 4 to $\tilde{\boldsymbol{Q}}^\top$ to find the sparsity pattern of $\boldsymbol{\zeta}_1^*$ denoted by $\boldsymbol{\tau}_1$.

Now we shrink $\tilde{\boldsymbol{Q}}$ once more using $\boldsymbol{\tau}_i, i = 1, 2$ to $\boldsymbol{Q}_s \in \mathbb{R}^{|\boldsymbol{\tau}_1|\times|\boldsymbol{\tau}_2|}$, and in the second stage of our approach, we estimate the active elements of $\boldsymbol{\zeta}_i$ via our active entry estimation procedure, presented in the following subsection. It is important to realize that $|\boldsymbol{\tau}_i| \sim n << Q_i$.

2.2 Second Stage: Active Entry Estimation

Let $\boldsymbol{\zeta}_i' \in \mathbb{R}^{|\boldsymbol{\tau}_i|}$, $i = 1, 2$ denote $\boldsymbol{\zeta}_i$ which are shrunk according to the sparsity pattern vectors $\boldsymbol{\tau}_i$ computed in the first stage. We then estimate the active entries via the following optimization program,

$$\phi'_{l_1}(\boldsymbol{\gamma}) = \max_{\boldsymbol{\zeta}_1 \in \mathcal{B}_+^{|\boldsymbol{\tau}_1|}, \boldsymbol{\zeta}_2 \in \mathcal{B}_+^{|\boldsymbol{\tau}_2|}} \boldsymbol{\zeta}_1'^\top \boldsymbol{Q}_s \boldsymbol{\zeta}_2' \tag{24}$$

Where $\boldsymbol{Q}_s \in \mathbb{R}^{|\boldsymbol{\tau}_1|} \times \mathbb{R}^{|\boldsymbol{\tau}_2|}$ is derived by shrinking the kernel alignment matrix $\boldsymbol{Q}$ according to the computed sparsity patterns. Now we can utilize any of the SVD algorithms to maximize Program 24. We used the power method to estimate these shrunken vectors.

Once $\boldsymbol{\zeta}_i^*$, $i = 1, 2$ are estimated, we can form the learned kernel matrices $\boldsymbol{K}_i, i = 1, 2$; and we can use a regular KCCA procedure to find $\boldsymbol{\alpha}_i^*, i = 1, 2$ in Program 4. We used a L_2 regularized KCCA in our package `SparKLe`.

2.3 Sparse Multiple Kernel learning Algorithm

In the previous Subsection we cast the sparsity pattern estimation problem as a concave minimization problem over a compact set. Here we apply a gradient descent algorithm, see Appendix B.1, to optimize Program 21. Below is our proposed algorithm.

Algorithm 1: Gradient descent algorithm for optimizing Program 21

```
Data: Kernel alignment matrix $\boldsymbol{Q} \in \mathbb{R}^{Q_1 \times Q_2}$
      $l_1$ sparsity controlling parameters $\gamma_i, i = 1, 2$.
      Initial value $\boldsymbol{\zeta}_1 \in \mathcal{S}^{Q_1}$
Result: $\boldsymbol{\tau}_2$, optimal sparsity pattern for $\boldsymbol{\zeta}_2^*$
initialization;
while convergence criterion is not met do
    $\boldsymbol{\zeta}_1 \leftarrow [\sum_{i=1}^{p_2} [\boldsymbol{q}_i^\top \boldsymbol{\zeta}_1 - \gamma_2]_+ \boldsymbol{q}_i - \gamma_1 \mathbf{1}]_+$
    $\boldsymbol{\zeta}_1 \leftarrow \frac{\boldsymbol{\zeta}_1}{\|\boldsymbol{\zeta}_1\|_2}$
Output $\boldsymbol{\tau}_2 \in \{0, 1\}^{p_2}$ where $\tau_{2i} = 0$ if $\boldsymbol{q}_i^\top \boldsymbol{\zeta}_1^* \leq \gamma_2$ and 1 otherwise.
```

As explained before, once the sparsity patterns are computed, which are, by design, sparse in the original feature space, and therefore interpretable for domain scientist, we fill in the active entries of kernel combination vectors via SVD decomposition of the shrunken alignment matrix $\boldsymbol{Q}_s$.

In the next section we extend our approach more than a pair of views/datasets.

3. Multi-View Sparse MKL

Sometimes multiple sets of covariates on matching subjects are observed, hence termed *Multi-View*, and the objective is to discover the association structures among these multiple views. Here, we extend the sMKL approach for a pair of views, introduced in Section 2, to $\boldsymbol{X}_i \in \mathbb{R}^{n \times p_i}, i = 1, \dots, m$.

Consider $\boldsymbol{Q}_{rs} \in \mathbb{R}^{Q_r \times Q_s}$, $r < s = 1, \dots, m$, where

$$[\boldsymbol{Q}_{rs}]_{ij} = \frac{1}{(n-1)^2} tr(\boldsymbol{K}_r^{(i)} \boldsymbol{H} \boldsymbol{K}_s^{(j)} \boldsymbol{H}) \tag{25}$$

We adopt the approach introduced in Solari et al. (2019), and maximize pairwise alignment matrices in the following program,

$$\phi_{l_x}^M(\boldsymbol{\Gamma}) = \max_{\substack{\boldsymbol{\zeta}_i \in \mathcal{B}_+^{Q_i} \\ \forall i=1,\dots,m}} \sum_{r<s=2}^{m} \boldsymbol{\zeta}_r^T \boldsymbol{Q}_{rs} \boldsymbol{\zeta}_s - \sum_{s=2}^{m} \sum_{\substack{r=1 \\ r \neq s}}^{s-1} \Gamma_{sr} \mathbf{1} \cdot \boldsymbol{\zeta}_s \tag{26}$$

where m is the number of observed views, $\boldsymbol{\Gamma} \in \mathbb{R}^{m \times m}$, $\Gamma_{ij} \geq 0$ is the sparsity parameter matrix, and $\boldsymbol{Q}_{rs}$ is as defined in 25. Following similar procedure as in 2, we analyze the solution to Program 26.

3.1 First Stage: Sparsity Pattern Estimation

Theorem 6 *Kernel mixture weights $\boldsymbol{\zeta}_1^*, \dots, \boldsymbol{\zeta}_m^*$ maximizing the optimization problem 3 is,*

$$\zeta_{si}^* = \zeta_{si}^*(\boldsymbol{\Gamma}) = \frac{[\sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{q}}_{rsi}^\top \boldsymbol{\zeta}_r - \sum_{\substack{r=1 \\ r \neq s}}^{m} \Gamma_{sr}]_+}{\sqrt{\sum_{k=1}^{Q_2} [\sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{q}}_{rsk}^\top \boldsymbol{\zeta}_r - \sum_{\substack{r=1 \\ r \neq s}}^{m} \Gamma_{sr}]_+^2}} \tag{27}$$

and for $r = 1, \dots, m$ and $r \neq s$,

$$\begin{aligned} \boldsymbol{\zeta}_r(\boldsymbol{\Gamma}) = \max_{\substack{\boldsymbol{\zeta}_r \in \mathcal{S}_+^{p_r} \\ r \neq s, r=1,\dots,m}} & \sum_{i=1}^{Q_s} [\sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{q}}_{rsi}^\top \boldsymbol{\zeta}_r - \sum_{\substack{r=1 \\ r \neq s}}^{m} \Gamma_{sr}]_+^2 + \\ & \sum_{\substack{i<j=2 \\ i,j \neq s}}^{m} \boldsymbol{\zeta}_i^\top \boldsymbol{Q}_{ij} \boldsymbol{\zeta}_j - \sum_{\substack{i=1 \\ i \neq s}}^{m} \sum_{\substack{j=1 \\ i \neq j}}^{m-1} \Gamma_{ij} \mathbf{1} \cdot \boldsymbol{\zeta}_i \end{aligned} \tag{28}$$

Proof Here we follow a progression similar to the proof of Theorem 4.

$$\phi_{l_1}^{m}(\boldsymbol{\Gamma}) = \max_{\substack{\boldsymbol{\zeta}_r \in \mathcal{B}_+^{Q_r}\\ r\neq s, r=1,\ldots,m}} \max_{\boldsymbol{\zeta}_s \in \mathcal{B}_+^{Q_s}} \sum_{r<s=2}^{m} \boldsymbol{\zeta}_r^\top \boldsymbol{Q}_{rs} \boldsymbol{\zeta}_s - \sum_{s=1}^{m} \sum_{\substack{r=1\\ r\neq s}}^{m-1} \Gamma_{sr} \mathbf{1} \cdot \boldsymbol{\zeta}_s \tag{29}$$

$$= \max_{\substack{\boldsymbol{\zeta}_r \in \mathcal{B}_+^{Q_r}\\ r\neq s, r=1,\ldots,m}} \max_{\boldsymbol{\zeta}_s \in \mathcal{B}_+^{Q_s}} \sum_{i=1}^{Q_s} \zeta_{si} \Big(\sum_{\substack{r=1\\ r\neq s}}^{m} \tilde{\boldsymbol{q}}_{rsi}^\top \boldsymbol{\zeta}_r - \sum_{\substack{r=1\\ r\neq s}}^{m} \Gamma_{sr} \Big) + \tag{30}$$

$$\sum_{\substack{i<j=2\\ i,j\neq s}}^{m} \boldsymbol{\zeta}_i^\top \boldsymbol{Q}_{ij} \boldsymbol{\zeta}_j - \sum_{\substack{i=1\\ i\neq s}}^{m} \sum_{\substack{j=1\\ i\neq j}}^{i-1} \Gamma_{ij} \mathbf{1} \cdot \boldsymbol{\zeta}_i \tag{31}$$

where $\tilde{\boldsymbol{q}}_{rsi} = \boldsymbol{q}_{rsi}$ if $r < s$, and $\tilde{\boldsymbol{q}}_{rsi} = \boldsymbol{q}_{rsi}^\top$ if $r > s$ where $\boldsymbol{q}_{rsi}$ is the ith row of $\boldsymbol{Q}_{rs}$. Optimizing for $\boldsymbol{\zeta}_s$ and normalizing, we get the local optimum in 27. Substituting back to 30,

$$\begin{aligned} \phi_{l_1}^{m2}(\boldsymbol{\Gamma}) = \max_{\substack{\boldsymbol{\zeta}_r \in \mathcal{B}_+^{Q_r}\\ r\neq s, r=1,\ldots,m}} & \sum_{i=1}^{Q_s} \Big[\sum_{\substack{r=1\\ r\neq s}}^{m} \tilde{\boldsymbol{q}}_{rsk}^\top \boldsymbol{\zeta}_r - \sum_{\substack{r=1\\ r\neq s}}^{m} \Gamma_{sr} \Big]_+^2 + \\ & \sum_{\substack{i<j=2\\ i,j\neq s}}^{m} \boldsymbol{\zeta}_i^\top \boldsymbol{Q}_{ij} \boldsymbol{\zeta}_j - \sum_{\substack{i=1\\ i\neq s}}^{m} \sum_{\substack{j=1\\ i\neq j}}^{m-1} \Gamma_{ij} \mathbf{1} \cdot \boldsymbol{\zeta}_i \end{aligned} \tag{32}$$

Now, as in Theorem 4, this program is a maximization of convex objective over a convex set, which is equivalent to a concave minimization program. Therefore, we can shrink the search domain to the boundaries of the search domain; i.e. a half sphere,

$$\begin{aligned} \phi_{l_1}^{m2}(\boldsymbol{\Gamma}) = \max_{\substack{\boldsymbol{\zeta}_r \in \mathcal{S}_+^{Q_r}\\ r\neq s, r=1,\ldots,m}} & \sum_{i=1}^{Q_s} \Big[\sum_{\substack{r=1\\ r\neq s}}^{m} \tilde{\boldsymbol{q}}_{rsk}^\top \boldsymbol{\zeta}_r - \sum_{\substack{r=1\\ r\neq s}}^{m} \Gamma_{sr} \Big]_+^2 + \\ & \sum_{\substack{i<j=2\\ i,j\neq s}}^{m} \boldsymbol{\zeta}_i^\top \boldsymbol{Q}_{ij} \boldsymbol{\zeta}_j - \sum_{\substack{i=1\\ i\neq s}}^{m} \sum_{\substack{j=1\\ i\neq j}}^{m-1} \Gamma_{ij} \mathbf{1} \cdot \boldsymbol{\zeta}_i \end{aligned} \tag{33}$$

■

As pointed out in Section 2, we're only interested in optimizing 33 in order to find the sparsity pattern $\boldsymbol{\tau}_s \in \{0, 1\}^{p_s}$.

Corollary 7 *For a sparsity parameter matrix* $\boldsymbol{\Gamma}$ *and the solution* $\boldsymbol{\zeta}_r^*$ *for* $s \neq r = 1, \ldots, m$ *to the Program 33,*

$$\boldsymbol{\tau}_{2i} = \begin{cases} 0 & \sum_{\substack{r=1\\ r\neq s}}^{m} \tilde{\boldsymbol{q}}_{rsi}^\top \boldsymbol{\zeta}_r \leq \sum_{\substack{r=1\\ r\neq s}}^{m} \Gamma_{sr} \\ 1 & \textit{otherwise} \end{cases} \tag{34}$$

Proof Scanning Equation 27,

$$\zeta^*_{si} = 0 \Leftrightarrow [\sum_{\substack{r=1 \\ r\neq s}}^{m} \tilde{\boldsymbol{q}}^\top_{rsi}\boldsymbol{\zeta}_r - \sum_{\substack{r=1 \\ r\neq s}}^{m} \Gamma_{sr}]^2_+ = 0 \Leftrightarrow \sum_{\substack{r=1 \\ r\neq s}}^{m} \tilde{\boldsymbol{q}}^\top_{rsi}\boldsymbol{\zeta}_r \leq \sum_{\substack{r=1 \\ r\neq s}}^{m} \Gamma_{sr} \tag{35}$$

Regardless of $\boldsymbol{\zeta}^*_r$ we have,

$$\sum_{\substack{r=1 \\ r\neq s}}^{m} \tilde{\boldsymbol{q}}^\top_{rsi}\boldsymbol{\zeta}_r \leq \sum_{\substack{r=1 \\ r\neq s}}^{m} \|\tilde{\boldsymbol{q}}_{rsi}\|_2 \|\boldsymbol{\zeta}_r\|_2 = \sum_{\substack{r=1 \\ r\neq s}}^{m} \|\tilde{\boldsymbol{q}}_{rsi}\|_2 \tag{36}$$

Hence, $\tau_{si} = 0$ for $i \in 1, \ldots, p_s$ if $\sum_{\substack{r=1 \\ r\neq s}}^{m} \|\tilde{\boldsymbol{q}}_{rsi}\|_2 \leq \sum_{\substack{r=1 \\ r\neq s}}^{m} \Gamma_{sr}$ regardless of $\boldsymbol{\zeta}^*_r$. ■

Once $\boldsymbol{\tau}_s$ is computed, we shrink all $\boldsymbol{Q}_{rs}$ along s to contain only the features along non-zero elements of $\boldsymbol{\tau}_s$. We repeat this process, each time shrinking alignment matrices to reflect the computed sparsity patterns until all $\boldsymbol{\tau}_i, i = 1, \ldots, m$ are estimated.

Computing $\boldsymbol{\tau}_i$ is the first stage of our two-stage multi-view multiple kernel learning approach, for which a fast algorithm is proposed in 5 as part of the `SparKLe` package.

3.2 Second Stage: Active Entry Estimation

Similar to Section 3.2, we shrink all kernel alignment matrices according to the estimated sparsity patterns. Let $[\boldsymbol{Q}'_{rs}]_{ij} = \{[\boldsymbol{Q}_{rs}]_{ij} | \boldsymbol{\tau}_{ri} = \boldsymbol{\tau}_{sj} = 1\}$ and $\boldsymbol{\zeta}'_{ri} = \{\boldsymbol{\zeta}_{ri} | \boldsymbol{\tau}_{ri} = 1\}$. We estimate $\boldsymbol{\zeta}'_r \in \mathbb{R}^{|\boldsymbol{\tau}_r|}$ for $r = 1, \ldots, m$ via the following optimization program.

$$\phi'^{M}_{l_x}(\boldsymbol{\Gamma}) = \max_{\substack{\boldsymbol{\zeta}'_i \in \mathcal{B}^{|\boldsymbol{\tau}_i|}_+ \\ \forall i=1,\ldots,m}} \sum_{r<s=2}^{m} \boldsymbol{\zeta}'^{T}_r \boldsymbol{Q}'_{rs} \boldsymbol{\zeta}'_s \tag{37}$$

We propose a simple power-method for this program, which is initialized using the estimates of the previous stage. This stage can be regarded as a post-processing stage, since the estimates from the previous stage have already high quality.

3.3 Multi-View Sparse Multiple Kernel Learning Algorithm

We propose the following first-order algorithm to optimize Program 33.

Algorithm 2: First-order algorithm for optimizing Program 33

```
Data: Kernel alignment matrices $\boldsymbol{Q}_{rs}$,   $1 \leq r < s \leq m$
      Sparsity parameter matrix $\boldsymbol{\Gamma} \in [0,1]^{m \times m}$
      Initial values $\boldsymbol{\zeta}_r \in \mathcal{S}_+^{Q_r}$,   $1 \leq r \leq m$
Result: $\boldsymbol{\tau}_s$, optimal sparsity pattern for $\boldsymbol{\zeta}_s$
initialization;
while convergence criterion is not met do
    for $r = 1, \ldots, m,\ r \neq s$ do
        $\boldsymbol{\zeta}_r \leftarrow [\sum_{i=1}^{Q_s} [\sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{q}}_{rsi}^\top \boldsymbol{\zeta}_r - \sum_{\substack{l=1 \\ l \neq s}}^{m} \Gamma_{sl}]_+ \tilde{\boldsymbol{q}}_{rsi} + \sum_{\substack{l=1 \\ l \neq r,s}}^{m} [\tilde{\boldsymbol{Q}}_{rl} \boldsymbol{\zeta}_l + \Gamma_{rl}]]_+$
        $\boldsymbol{\zeta}_r \leftarrow \frac{\boldsymbol{\zeta}_r}{\|\boldsymbol{\zeta}_r\|_2}$
Output $\boldsymbol{\tau}_s \in \{0,1\}^{p_s}$, where $\tau_{si} = 0$ if $\sum_{\substack{r=1 \\ r \neq s}}^{m} \tilde{\boldsymbol{q}}_{rsi}^\top \boldsymbol{\zeta}_r \leq \sum_{\substack{r=1 \\ r \neq s}}^{m} \Gamma_{sr}$ and 1 otherwise.
```

We repeat this algorithm m times, each time estimating a single $\boldsymbol{\tau}_i$ and shrinking any alignment matrix $\boldsymbol{Q}_{ri}$ according to the proposed procedure in 3.2. We propose the following algorithm for optimization problem 37,

Algorithm 3: Power iterations algorithm for optimizing Program 37

```
Data: Shrunk kernel alignment matrices $\boldsymbol{Q}'_{rs}$,   $1 \leq r < s \leq m$
      Initial values $\boldsymbol{z}'_r \in \mathcal{S}^{|\boldsymbol{\tau}_r|}$,   $1 \leq r \leq m$
Result: $\boldsymbol{z}'_r$, $r = 1, \ldots, m$, estimated active elements of $\boldsymbol{z}_r$
initialization;
for $r = m, \ldots, 1$ do
    while convergence criterion is not met do
        $\boldsymbol{z}'_r \leftarrow \sum_{s=1}^{r} \boldsymbol{C}_{sr}(\boldsymbol{C}_{sr}^\top \boldsymbol{z}'_r) + \sum_{s=r+1}^{m} \boldsymbol{C}'_{rs} \boldsymbol{z}'_s$
        $\boldsymbol{z}_r \leftarrow \frac{\boldsymbol{z}_r}{\|\boldsymbol{z}_r\|_2}$
```

Note that Algorithm 2 already results in high-quality solutions; hence we strongly suggest using them as the initial values in Algorithm 3.

4. Kernel CCA

So far we have discussed the multiple Kernel Learning problem; where we learned kernel matrices in the form of a convex combination of base kernels, i.e. $\boldsymbol{K}_i = \sum_{q=1}^{Q_i} \zeta_{iq} \boldsymbol{K}_i^{(q)}$, $i = 1, \ldots, m$. In the last stage we estimate the canonical directions , $\boldsymbol{\alpha}_i^*$ in Program 4, via a simple regularized kernel CCA as follows,

$$(\boldsymbol{\alpha}_1^*, \boldsymbol{\alpha}_2^*) = \underset{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \in \mathbb{R}^n}{\arg\max} \frac{\boldsymbol{\alpha}_1^\top \boldsymbol{K}_1^\top \boldsymbol{K}_2 \boldsymbol{\alpha}_2}{\sqrt{\boldsymbol{\alpha}_1^\top \boldsymbol{K}_1'^\top \boldsymbol{K}_1' \boldsymbol{\alpha}_1} \sqrt{\boldsymbol{\alpha}_2'^\top \boldsymbol{K}_2'^\top \boldsymbol{K}_2' \boldsymbol{\alpha}_2}} \tag{38}$$

where $\boldsymbol{K}_i' = \boldsymbol{K}_i + \eta \boldsymbol{I}$, and $\eta \geq 0$ is the regularization parameter. Inspired by Kettenring (1971), we extend the KCCA approach to multi-view settings. Let

$$\boldsymbol{K}_O = \begin{bmatrix} \mathbf{0} & \boldsymbol{K}_{12} & \cdots & \boldsymbol{K}_{1m} \\ \boldsymbol{K}_{21} & \mathbf{0} & & \vdots \\ \vdots & & \ddots & \boldsymbol{K}_{(m-1)m} \\ \boldsymbol{K}_{m1} & & \boldsymbol{K}_{m(m-1)} & \mathbf{0} \end{bmatrix}, \quad \boldsymbol{K}_D = \begin{bmatrix} \boldsymbol{K}_{11} + \eta \boldsymbol{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{K}_{22} + \eta \boldsymbol{I} & & \vdots \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{K}_{mm} + \eta \boldsymbol{I} \end{bmatrix} \tag{39}$$

and $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_m]$ and $\eta \geq 0$. We compute $\boldsymbol{\alpha}$ by solving a *Generalized Eigenvalue Problem* which can be cast into a regular eigenvalue problem,

$$\boldsymbol{K}_O \boldsymbol{\alpha} = \lambda \boldsymbol{K}_D \boldsymbol{\alpha} \Rightarrow \boldsymbol{K}_D^{-1} \boldsymbol{K}_O \boldsymbol{\alpha} = \lambda \boldsymbol{\alpha} \Rightarrow \boldsymbol{K}_O^{\frac{1}{2}} \boldsymbol{K}_D^{-1} \boldsymbol{K}_O^{\frac{1}{2}} (\boldsymbol{K}_O^{\frac{1}{2}} \boldsymbol{\alpha}) = \lambda (\boldsymbol{K}_O^{\frac{1}{2}} \boldsymbol{\alpha}) \tag{40}$$

The `geigen`[1] package provides sufficient functionality to solve such GEV problems. It uses the routines implemented in `LAPACK`[2].

In the next section We introduce a new, the only one in fact, `Apache Spark` API implementation of the algorithms proposed so far in this article along with the ones proposed in Solari et al. (2019) to make a comprehensive large-scale multi-view learning package, called `SparKLe`, which is able to deploy these algorithms on extreme-scale datasets in cloud computing environments, where individual views are large, and/or where large numbers of views are available.

5. SparKLe

In multi-omics, each view may have millions of features. When the number of views is large, naive formulations of sCCA, and its kernel variant, require many terabytes of RAM, which is not often available to users of these methods. While map-reduce frameworks have often been applied to large numbers of "observations", here the challenge is enormously large numbers of parameters – yet the problem is the same. We need to distribute data to compute nodes to avoid the need for intractably large memory machines. The methods in the present work lend themselves to datasets with extremely large numbers of covariates, which we call *wide* datasets, but require the computation of the sample cross-covariance matrix $X_1^\top X_2$. In such high-dimensional settings, this matrix can easily reach memory requirements far beyond the capabilities of most off-the-shelf computing environments. For example, in the relatively modest case in which $X1$ and $X2$ each has $50,000$ columns, $X_1^\top X_2$ requires approximately 20 GB of memory to compute. This problem is compounded in the multi-view setting in which we must also compute all pairwise cross-covariance matrices. Rather than hit the problem on the head with the hammer of big RAM computational environments, a more practical and scalable approach involves splitting $X_1^\top X_2$ into chunks and spreading storage and computation across a number of nodes in a cluster.

This was the motivation behind the development of a new `PySpark` package known as `SparKLe`. `SparKLe` uses the `Apache Spark` API for `Python` (`PySpark`) to circumvent the above-mentioned memory and performance constraints inherent in high-dimensional data analysis. `Apache Spark` [Zaharia et al. (2016)] is an open-source distributed data analytics framework in the `MapReduce` lineage of computational paradigms that was originally developed at the University of California, Berkeley's AMPLab. The main data structure on which Spark is built is the Resilient Distributed Dataset (`RDD`), a fault-tolerant partitioning mechanism that distributes chunks of data across nodes in a cluster computing environment. Operations on that data are carried out in parallel (the `map` stage) and combined two at a time to form the final result (the `reduce` stage). `SparKLe` takes advantage of this framework to compute and deflate cross-covariance matrices for an arbitrary

[1] https://CRAN.R-project.org/package=geigen

[2] http://github.com/Reference-LAPACK

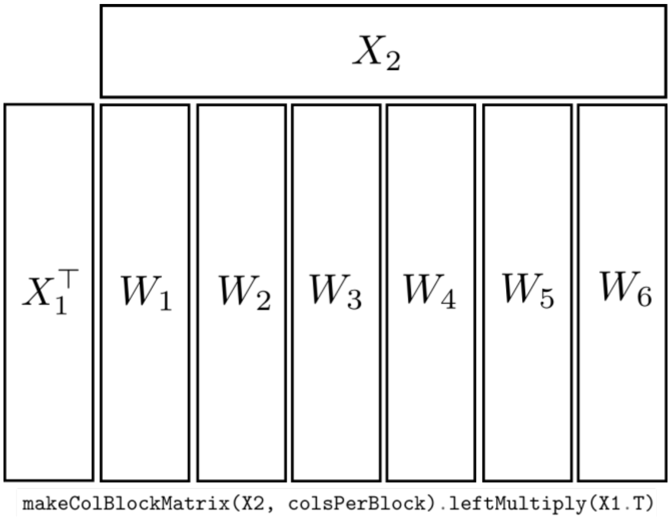

Figure 1: The `ColBlockMatrix` data structure. X_1 and X_2 both fit in memory but their product $X_1^\top X_2$ is too large and must be distributed to the six workers $W_1, \ldots, W_6$. The `makeColBlockMatrix` utility function splits X_1 into six equal-sized blocks and returns a `ColBlockMatrix`. Calling the `ColBlockMatrix`'s `leftMultipy` method with $X_1^\top$ completes the cross-covariance matrix creation in parallel on the six worker nodes.

number of views, learn sparse non-linear associations between those views, and carry out a cross-validated hyperparameter grid search, scalably and in parallel.

5.1 `SparKLe` Design Paradigm

`SparKLe`'s design follows the pipeline paradigm introduced by the `scikit-learn` project and carried over to the Spark ML module[3] [Meng et al. (2016)]. However, unlike the `Spark` ML API, `SparKLe` is designed to work with an arbitrary number of high dimensional numeric datasets that fit in memory (stored as `numpy.ndarrays`) but whose matrix product may be too large to store in RAM without a specialized big memory compute node. As an alternative, the cross-covariance matrix can instead be distributed across a number of worker nodes and operated on in parallel.

`SparKLe` achieves this via a new `RDD`-backed data structure known as `ColBlockMatrix`. A `ColBlockMatrix` is similar to the `BlockMatrix` in textttSpark ML, but is specialized to high dimensional datasets and provides a number of methods that are useful for the methods presented in this paper, such as quickly finding the column with maximum L_2 norm and subtracting the outer product of two vectors. Another method that `ColBlockMatrix` provides is matrix multiplication on the left, which is demonstrated in Figure 1. While the `ColBlockMatrix` was developed with CCA in mind, it can be used and extended for efficient computation in any setting in which high dimensional matrices are operated on.

5.2 `SparKLe` Basic Usage

The main entry point to the functionality provided in the `SparKLe` package is the `CCA` class which implements the multi-view sparse kernel CCA learning methods described in the present work. The `CCA` class, in keeping with the abstractions first developed in `scikit-learn`, is an `Estimator` and, having learned on the data,

[3] https://spark.apache.org/docs/latest/ml-guide.html

returns a Transformer (known as CCAModel) which can operate on the same or new data. In our context, CCA learns a transformation from the high dimensional space of the original dataset to a low dimensional space in which the pairwise correlation between datasets is maximized subject to the sparsity penalty previously described. It then saves that transformation in a CCAModel object for future use.

To make the implementation more concrete, we provide the following example of basic package usage:

```python
from sparkle.cca import *

# instantiate a new CCA object with hyperparameters k and rhos
cca = CCA(k = 5, rhos = (0.1, 0.3, 0.7, 0.4), verbose = True)

# train on four datasets, returning a CCAModel instance
cca_model = cca.fit([data1, data2, data3, data4])

# the four canonical loadings matrices are stored in the ZZ
# field of the CCAModel
cca_model.ZZ

# we can see what the pairwise canonical correlations were between
# the four datasets
cca_model.canonicalCorrelations([data1, data2, data3, data4])

# we can use the transform method to produce predictions for each
# dataset using the three other datasets
cca_model.transform([data1, data2, data3, data4])

# finally, save the CCAModel instance for later use
cca_model.save("path/to/project")
```

In this example, we requested $k = 5$ canonical components and arbitrarily chose the sparsity parameters $\rho_1 = 0.1, \rho_2 = 0.3, \rho_3 = 0.7$ and $\rho_4 = 0.4$. One way we could improve this workflow is by splitting our data into training and test sets and use the methods in sparkle.cca.CCAModel to evaluate the fit of our hyperparameter choices. However, the next section presents our package's built-in functionality to automate this workflow via cross validation.

5.3 SparKLe Cross Validation Workflow

A typical cross validation workflow is as follows:

1. Instantiate new sparkle.CCA and sparkle.cca.CCAEvaluator objects.
2. Instantiate a pyspark.ml.tuning.ParamGridBuilder with the hyperparameters to test.
3. Instantiate a sparkle.cca.CCACrossValidator object and add to it the previously instantiated objects.
4. Call the fit method of the sparkle.cca.CCACrossValidator object with any number of datasets, stored in memory as numpy.ndarrays.
5. The fit method returns a sparkle.cca.CCACrossValidatorModel object which has the field bestModel containing the sparkle.cca.CCAModel with the highest training accuracy.

The following code example demonstrates this multiview cross-validation workflow:

```python
from sparkle.cca import *
from pyspark.ml.tuning import ParamGridBuilder

# instantiate CCA and CCAEvaluator objects
cca = CCA()
evaluator = CCAEvaluator()

# create a ParamGridBuilder instance with a grid of hyperparameters
paramGrid = ParamGridBuilder() \
  .baseOn({cca.broadcast: True}) \ # shared settings / parameters
  .addGrid(cca.k, range(3, 11)) \ # number of components
  .addGrid(cca.rhos, [0.1, 0.5, 0.9]) \ # tuning parameter
  .build()

# create a CCACrossValidator instance
cv = CCACrossValidator(parallelism = 4) \ # num. cores to use for CV
  .setEstimator(cca) \
  .setEvaluator(evaluator) \
  .setEstimatorParamMaps(paramGrid) \
  .setNumFolds(10) \ # number of CV training folds
  .setVerbose(True)

# run the cross validation on training data
cv_fit = cv.fit([train1, train2, train3])

# see the best tuning parameters
cv_fit.bestModel.getK()
cv_fit.bestModel.getRhos()

# see the canonical correlations for training and test data
cv_fit.bestModel.canonicalCorrelations([train1, train2, train3])
cv_fit.bestModel.canonicalCorrelations([test1, test2, test3])

# save the best model for later use
cv_fit.save("path/to/project")
```

One advantage of the modular approach we have taken here is that a user can easily extend `Sparkle` in ways that fit their needs and are familiar to anyone with experience of Spark ML module. For example, users are not tied to our CV evaluation metric in `sparkle.cca.CCAEvaluator`, which simply returns the mean canonical correlation between all pairs of datasets, but may instead write their own custom `Evaluator` to fit their needs.

6. Experiments

Now we apply different *SparKLe* modules to simulated datasets and compare the results to other popular approaches.

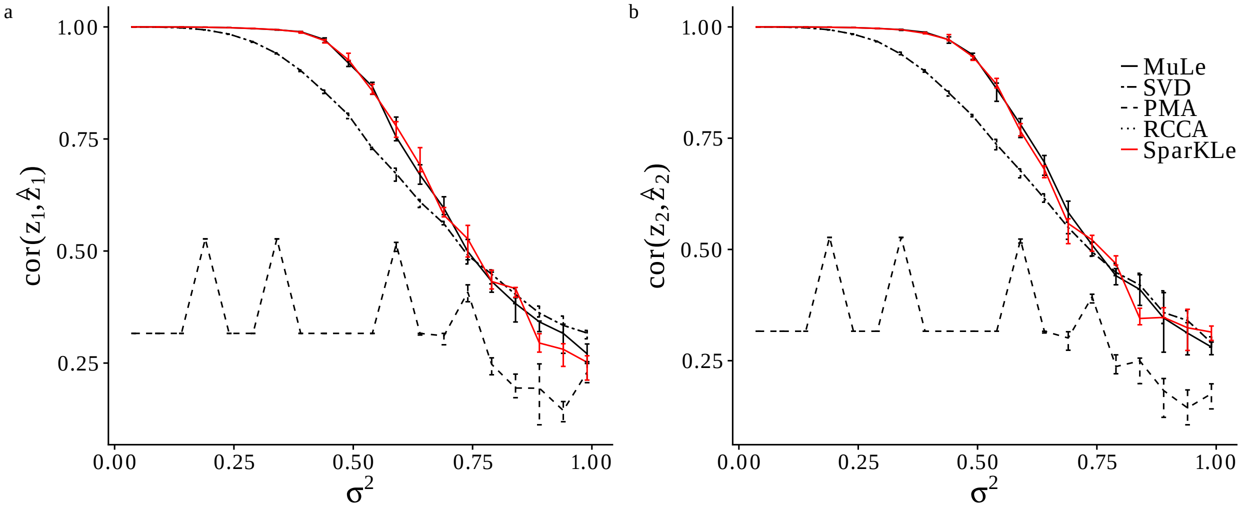

Figure 2: Correlation of the estimated and true canonical directions plotted for both views.

6.1 Group Linear Association

We first benchmark `sparkle.CCA` module against `PMA` [Witten and Tibshirani (2009)], `MuLe` [Solari et al. (2019)], simple SVD, and hard-thresholding applied to the output of the ridge regularized CCA of the `CCA` package[4]. To this end, we repeat the same test as in Solari et al. (2019). Hence, we created a pair of views $\boldsymbol{X}_i \in \mathbb{R}^{n \times p_i}$, $i = 1, 2$ according to the following procedure,

$$\boldsymbol{X}_1 = (\boldsymbol{z}_1 + \boldsymbol{\epsilon}_1)\boldsymbol{u}^\top, \quad \boldsymbol{X}_2 = (\boldsymbol{z}_2 + \boldsymbol{\epsilon}_2)\boldsymbol{u}^\top \tag{41}$$

where $\boldsymbol{z}_1 \in \mathbb{R}^{1000}$ and $\boldsymbol{z}_2 \in \mathbb{R}^{800}$ have the following sparsity patterns,

$$\begin{aligned} \boldsymbol{z}_1 &= \Big[\underbrace{1, \ldots, 1}_{50} \quad \underbrace{-1, \ldots, -1}_{50} \quad \underbrace{0, \ldots, 0}_{900}\Big] \\ \boldsymbol{z}_2 &= \Big[\underbrace{1, \ldots, 1}_{50} \quad \underbrace{-1, \ldots, -1}_{50} \quad \underbrace{0, \ldots, 0}_{700}\Big] \end{aligned} \tag{42}$$

$\boldsymbol{\epsilon}_1 \in \mathbb{R}^{800}$ and $\boldsymbol{\epsilon}_2 \in \mathbb{R}^{1000}$ are added Gaussian noise.

$$\begin{aligned} \boldsymbol{\epsilon}_1 &\sim \mathcal{N}(0, \sigma^2), \forall i = 1, \ldots 1000, \\ \boldsymbol{\epsilon}_2 &\sim \mathcal{N}(0, \sigma^2), \forall i = 1, \ldots 800, \end{aligned} \tag{43}$$

and

$$\boldsymbol{u}_i \sim \mathcal{N}(0, 1), \forall i = 1, \ldots, 100. \tag{44}$$

We computed the correlation of the estimated, $\hat{\boldsymbol{z}}_i$, and true, $\boldsymbol{z}_i$, canonical directions with varying level of noise σ^2. We plotted the results in Figure 2 for multiple algorithms for both canonical directions; according to which, `MuLe` and `SparKLe` outperform other methods, especially the alternating optimization based

[4] https://cran.r-project.org/web/packages/CCA/CCA.pdf

method of Witten and Tibshirani (2009), throughout the range of noise amplitude, and their performance is almost identical. The built-in parameter tuning procedure of `PMA` also mis-specified the correct sparsity parameters, so we provided the correct hyperparameters manually which did not improve the results considerably. Interestingly, a simple thresholding algorithm like `RCCA` outperforms `PMA` both in terms of support recovery and direction estimation.

6.2 Single Non-Linear Association

First, we compare our approach's capability in identifying non-linear single associations between two high-dimensional views. To this end, we create the views $\boldsymbol{X}_i \in \mathbb{R}^{n\times p_i}$, $i = 1, 2$, where

$$\begin{aligned}
&[\boldsymbol{X}_1]_{.j} \sim \mathcal{U}(-1,1), \quad j = 1,\dots,p_1 \\
&[\boldsymbol{X}_2]_{.1} = ([\boldsymbol{X}_1]_{.1} + 1)^2 + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0,\sigma^2) \\
&[\boldsymbol{X}_2]_{.j} \sim \mathcal{U}(-1,1), \quad j = 2,\dots,p_2
\end{aligned} \tag{45}$$

We took samples of size $n = 50$ with $p_1 = p_2 = 500$. For $0.01 \leq \sigma^2 \leq 1$, we compared the canonical correlations computed via `SparKLe` and `TSKCCA` where hyper-parameter tuning was performed using 5-fold cross-validation. We used a ridge regularized KCCA for the second stage with exactly the same parameters for the two approaches compared, hence the difference in canonical correlation is solely due to the multiple kernel learning procedures. We used simple *radial basis function (rbf)* as our kernel basis functions,

$$k(\boldsymbol{x},\boldsymbol{x}') = exp(\frac{-\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\boldsymbol{\sigma}^2}) \tag{46}$$

For each value of σ, we repeated the analysis 10 times and plotted the mean canonical correlation in Figure 3.a. `SparKLe` obviously does a better job of capturing the association structure with much lower variation in the fitted models throughout the added noise range. One of the most important observations that we made was that while our approach successfully selected $\boldsymbol{K}_1^{(1)}$ and $\boldsymbol{K}_2^{(1)}$ to be associated in the MKL stage for σ^2 of almost up to 0.25, `TSKCCA` selected many more base kernel matrices and chose much much less sparse models, which is the reason behind almost uniform performance independent the noise amplitude.

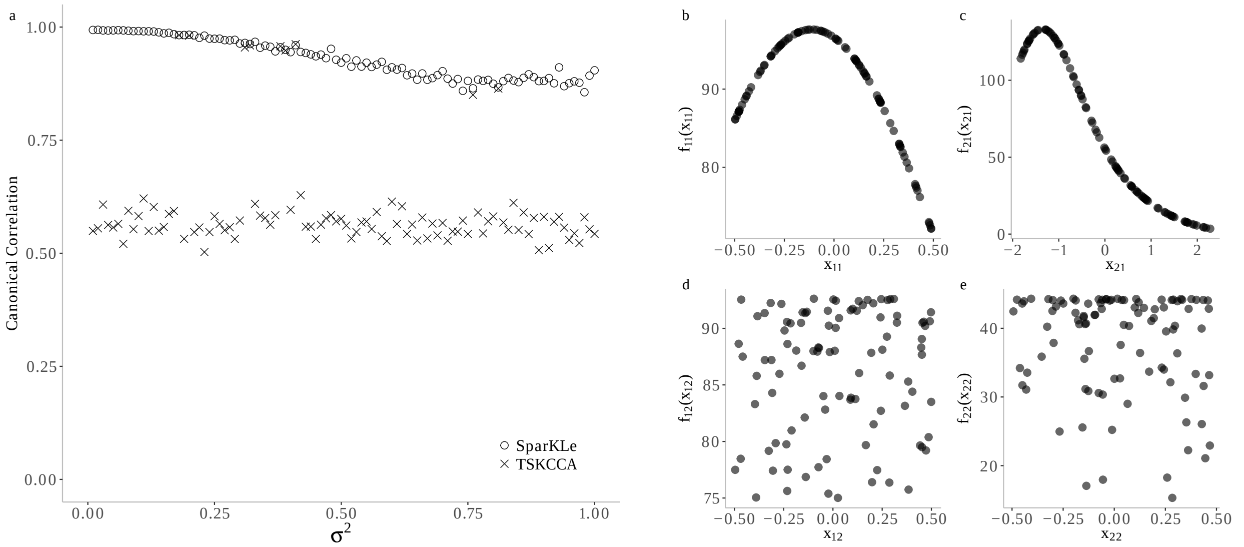

Figure 3: Single non-linear association experiment. **a.** Median canonical correlation of `SparKLe` and `TSKCCA`. **(b,c,d,e).** Learned transformations via `SparKLe` **(b,c)** shows the associated feature functions and **(d,e)** demonstrate the unassociated feature functions.

6.3 Runtime Experiment

In this experiment we compare the run-time of our `Spark` based package to other methods. We exclude `TSKCCA` here since their sparse Multiple Kernel Learning method is based on `PMA`, which is an alternating maximization algorithm and in Solari et al. (2019) its super-quadratic time-complexity is established. Besides at this time, no python implementation of their method is available. Therefore, we compare the running time of `SparKLe` with that of `Pyrcca` [Bilenko and Gallant (2016)], which is a python implementation regularized CCA, on a two-view problem while keeping the number of columns of the two datasets fixed and allowing the number of observations to grow by a factor of 1.1. We chose linear kernels for this experiment. As apparent in Figure 4, `SparKLe`'s running time stayed nearly linear in the number of observations while `Pyrcca`'s was quadratic.

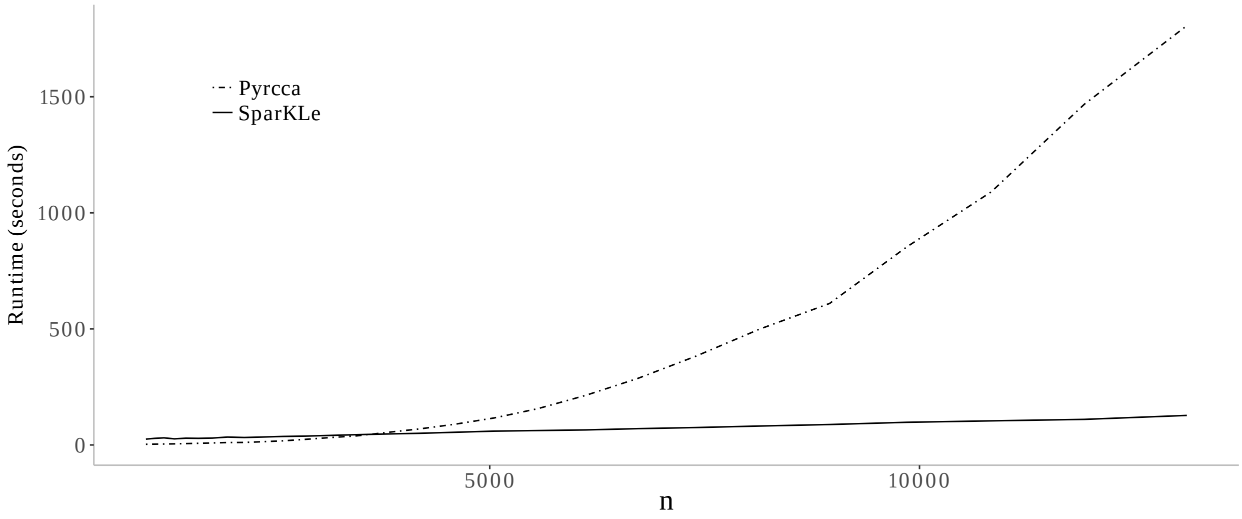

Figure 4: Running time of `SparKLe` and `pyrcca` on simulated data. Each algorithm was run 3 times for each value of n starting at 1,000 and increasing by a factor of 1.1 to 13,110. The number of columns is fixed at $p_1 = 7000$ and $p_2 = 10,000$. Simulations were carried out in Spark local mode with 1 core (no parallelism).

Appendix A. Proofs

A.1 Proposition 2

Substituting $\boldsymbol{K}_i = \sum_{q=1}^{Q_i} \zeta_{iq} \boldsymbol{K}_i^{(q)}$ in 12,

$$
\begin{aligned}
\phi_{l_1}(\boldsymbol{\gamma}) &= \max_{\boldsymbol{\zeta}_1 \in \mathcal{S}_+^{Q_1}, \boldsymbol{\zeta}_2 \in \mathcal{S}_+^{Q_2}} tr(\boldsymbol{K}_1 \boldsymbol{H} \boldsymbol{K}_2 \boldsymbol{H}) - \gamma_1 \mathbf{1} \cdot \boldsymbol{\zeta}_1 - \gamma_2 \mathbf{1} \cdot \boldsymbol{\zeta}_2 \\
&= \max_{\boldsymbol{\zeta}_1 \in \mathcal{S}_+^{Q_1}, \boldsymbol{\zeta}_2 \in \mathcal{S}_+^{Q_2}} tr(\sum_{q=1}^{Q_1} \zeta_{1q} \boldsymbol{K}_1^{(q)} \boldsymbol{H} \sum_{q=1}^{Q_2} \zeta_{2q} \boldsymbol{K}_2^{(q)} \boldsymbol{H}) - \gamma_1 \mathbf{1} \cdot \boldsymbol{\zeta}_1 - \gamma_2 \mathbf{1} \cdot \boldsymbol{\zeta}_2 \\
&= \max_{\boldsymbol{\zeta}_1 \in \mathcal{S}_+^{Q_1}, \boldsymbol{\zeta}_2 \in \mathcal{S}_+^{Q_2}} \sum_{q_1=1}^{Q_1} \sum_{q_2=1}^{Q_2} \zeta_{1q_1} tr(\boldsymbol{K}_1^{(q)} \boldsymbol{H} \boldsymbol{K}_2^{(q)} \boldsymbol{H}) \zeta_{2q_2} - \gamma_1 \mathbf{1} \cdot \boldsymbol{\zeta}_1 - \gamma_2 \mathbf{1} \cdot \boldsymbol{\zeta}_2 \\
&= \max_{\boldsymbol{\zeta}_1 \in \mathcal{S}_+^{Q_1}, \boldsymbol{\zeta}_2 \in \mathcal{S}_+^{Q_2}} \boldsymbol{\zeta_1 Q \zeta_2} - \gamma_1 \mathbf{1} \cdot \boldsymbol{\zeta}_1 - \gamma_2 \mathbf{1} \cdot \boldsymbol{\zeta}_2
\end{aligned}
\tag{47}
$$

where $\boldsymbol{Q} \in \mathbb{R}_+^{Q_1 \times Q_2}$ is the kernel alignment matrix given by Equation 15.

Appendix B. Algorithms

B.1 First-Degree Concave Minimization Algorithm

Algorithm 4: A first-order concave minimization method.

```
Data: $\boldsymbol{z}_0 \in \mathcal{Q}$
Result: $\boldsymbol{z}^* = \arg\min_{\boldsymbol{z} \in \mathcal{Q}} -f(\boldsymbol{z})$ where $f(x)$ is convex.
$k \leftarrow 0$
while convergence criterion is not met do
    $\boldsymbol{z}_{k+1} \leftarrow \arg\min_{x \in \mathcal{Q}} (f(z_k) + (x - z_k)^T f'(z_k))$
    $k \leftarrow k + 1$
```

References

F.R. Bach and M.I. Jordan. Kernel independent component analysis. *Journal of machine learning research*, pages 1–48, 2002.

Francis R Bach, Gert RG Lanckriet, and Michael I Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6. ACM, 2004.

Natalia Y Bilenko and Jack L Gallant. Pyrcca: regularized kernel canonical correlation analysis in python and its applications to neuroimaging. *Frontiers in neuroinformatics*, 10:49, 2016.

C Christoudias, Raquel Urtasun, and Trevor Darrell. Multi-view learning in the presence of view disagreement. *arXiv preprint arXiv:1206.3242*, 2012.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Two-stage learning kernel algorithms. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 239–246. Omnipress, 2010.

Michele Covell and Malcolm Slaney. Canonical correlation analysis of image/control-point location coupling for the automatic location of control points, June 4 2002. US Patent 6,400,828.

Nello Cristianini, John Shawe-Taylor, Andre Elisseeff, and Jaz S Kandola. On kernel-target alignment. In *Advances in neural information processing systems*, pages 367–373, 2002.

Ying Cui, Xiaoli Z Fern, and Jennifer G Dy. Non-redundant multi-view clustering via orthogonalization. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 133–142. IEEE, 2007.

Maria Giovanna Dainotti, Richard Willingale, Salvatore Capozziello, Vincenzo Fabrizio Cardone, and Michał Ostrowski. Discovery of a tight correlation for gamma-ray burst afterglows with "canonical" light curves. *The Astrophysical Journal Letters*, 722(2):L215, 2010.

Siamak Faridani. Using canonical correlation analysis for generalized sentiment analysis, product recommendation and search. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 355–358. ACM, 2011.

O. Friman, J. Cedefamn, P. Lundberg, M. Borga, and H. Knutsson. Detection of neural activity in functional mri using canonical correlation analysis. *Magnetic Resonance in Medicine*, 45:323–330, 2001.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

C.E. Hopkins. Statistical analysis by canonical correlation: a computer application. *Health services research*, 4(4):304, 1969.

H. Hotelling. The most predictable criterion. *Journal of Educational Psychology*, 26:139–142, 1935.

W.W. Hsieh. Nonlinear canonical correlation analysis by neural networks. *Neural Networks*, 10:1095–1105, 2000.

Kendrick N Kay, Thomas Naselaris, Ryan J Prenger, and Jack L Gallant. Identifying natural images from human brain activity. *Nature*, 452(7185):352, 2008.

Jon R Kettenring. Canonical analysis of several sets of variables. *Biometrika*, 58(3):433–451, 1971.

Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine learning research*, 5(Jan):27–72, 2004.

H. Lindsey, J.T. Webster, , and S. Halper. Canonical correlation as a discriminant tool in a periodontal problem. *Biometrical journal*, 3(27):257–264, 1985.

Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, et al. Mllib: Machine learning in apache spark. *The Journal of Machine Learning Research*, 17(1):1235–1241, 2016.

James Mercer. Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209(441-458):415–446, 1909.

M.S. Monmonier and F.E. Finn. Improving the interpretation of geographical canonical correlation models. *The Professional Geographer*, 25:140–142, 1973.

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

S.V. Schell and W.A. Gardner. Programmable canonical correlation analysis: A flexible framework for blind adaptive spatial filtering. *IEEE transactions on signal processing*, 43(12):2898–2908, 1995.

D. Simonson, J. Stowe, and C. Watson. A canonical correlation analysis of commercial bank asset/liability structures. *Journal of Financial and Quantitative Analysis*, 10:125–140, 1983.

Omid S Solari, James B Brown, and Peter J Bickel. Sparse canonical correlation analysis via concave minimization. *arXiv preprint arXiv:1909.07947*, 2019.

Giulia Tini, Luca Marchetti, Corrado Priami, and Marie-Pier Scott-Boyer. Multi-omics integration—a comparison of unsupervised clustering methodologies. *Briefings in bioinformatics*, 2017.

X.M. Tu, D.S. Burdick, D.W. Millican, and L.B. McGown. Canonical correlation technique for rank estimation of excitation-emission matrices. *Analytical Chemistry*, 19(61):2219–2224, 1989.

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

F.V. Waugh. Regressions between sets of variables. *Econometrica, Journal of the Econometric Society*, page 290–310, 1942.

D. Witten and R. Tibshirani. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical Applications in Genomics and Molecular Biology*, 8, 2009.

K.W. Wong, P.C.W. Fung, and C.C. Lau. Study of the mathematical approximations made in the basis correlation method and those made in the canonical-transformation method for an interacting bose gas. *Physical Review*, 3(22):1272, 1980.

Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

Kosuke Yoshida, Junichiro Yoshimoto, and Kenji Doya. Sparse kernel canonical correlation analysis for discovery of nonlinear interactions in high-dimensional data. *BMC bioinformatics*, 18(1):108, 2017.

Matei Zaharia, Reynold S Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J Franklin, et al. Apache spark: a unified engine for big data processing. *Communications of the ACM*, 59(11):56–65, 2016.