MOSCATO: A Supervised Approach for Analyzing Multi-Omic Single-Cell Data

Lorin M Towle-Miller[a]
Jeffrey C Miecznikowski[a]
[a]University at Buffalo

Abstract

Advancements in genomic sequencing continually improve personalized medicine in complex diseases. Recent breakthroughs generate multiple types of signatures (or multi-omics) from each cell, producing different data 'omic' types per single-cell experiment. We introduce MOSCATO, a technique for selecting features across multi-omic single-cell datasets that relate to clinical outcomes. For example, we leverage penalization concepts often used in multi-omic network analytics to accommodate the high-dimensionality where multiple-testing is likely underpowered. We organize the data into multi-dimensional tensors where the dimensions correspond to the different 'omic' types. Using the outcome and the single-cell tensors, we perform regularized tensor regression to return a variable set for each 'omic' type that forms the clinically-associated network. Robustness is assessed over simulations based on available single-cell simulation methods. Real data comparing healthy subjects versus subjects with leukemia is also considered in order to identify genes associated with the disease. The flexibility of our approach enables future extensions on distributional assumptions and covariate adjustments. This algorithm may identify clinically-relevant genetic patterns on a cellular-level that span multiple layers of sequencing data and ultimately inform highly precise therapeutic targets in complex diseases. Code to perform MOSCATO and replicate the real data application is publicly available on GitHub at https://github.com/lorinmil/MOSCATO and https://github.com/lorinmil/MOSCATOLeukemiaExample.

Keywords: Tensor regression, Single-cell sequencing, Network analysis

1 Introduction

Classic bulk genetic sequencing involves averaging signature levels across all cells. Different sequencers may sequence different types of molecules such as ribonucleic acid (RNA), proteins, DNA methyl groups, etc. Disease progression, therapy success, and other clinical outcomes often vary among individuals suffering from complex diseases [3, 7, 17, 32], and the heterogeneity in their outcomes may be better understood through the intricacies of a patient's molecular signatures [5, 25, 24, 1]. This has led to an explosive demand for multi-omics which involves integrating multiple types of molecular information in order to have a more Systems Biology approach. For example, in breast cancer patients with resistance to lapatinib therapy, Komurov et al. were able to suggest additional therapy targets by identifying combinations of RNA and proteins responsible for glucose deprivation that was associated with the resilience [18].

Methods for identifying graphs and gene regulatory networks within a single molecular type has been well studied [20, 14], however, different methods should be considered when integrating multiple types of molecular information in order to accommodate the between and within molecular relationships [6]. Each molecular type often contains thousands of features, and integrating them creates a higher dimensional problem with more sophisticated relationships both within and between molecular types. For example, the Decomposition of Network Summary Matrix via Instability (DNSMI) method decomposes a matrix of network strengths by fitting a series of models for the expected relationships across molecular types and with the disease outcome [35]. Supervised sparse Canonical Correlation Analysis (SCCA) attempts to optimize

the correlation matrix between molecular types through lasso constrained linear combinations of the features and also eliminates features weakly correlated with the outcome [33].

In bulk sequencing experiments, rare cells or smaller cell-types will be diluted due to the averaging across all cells within the sample. This motivated single-cell sequencing techniques where molecular information could then be sequenced on a cell-by-cell basis. While initial protocols were limited to RNA [23, 26], newer technology may now sequence multiple types of molecular information within each cell, denoted as *multi-modal* (or *multi-omic*) single-cell sequencing. For example, CITE-seq simultaneously sequences both cell surface proteins and RNA on each cell of a sample [29]. Although still a growing technology, applications have already been considered using this novel sequencing approach. For example, Kendal et al. utilized CITE-seq technology to compare tendons in healthy individuals to those with tendinopathy [15]. Figure 1 displays an example of single-cell data from each patient.

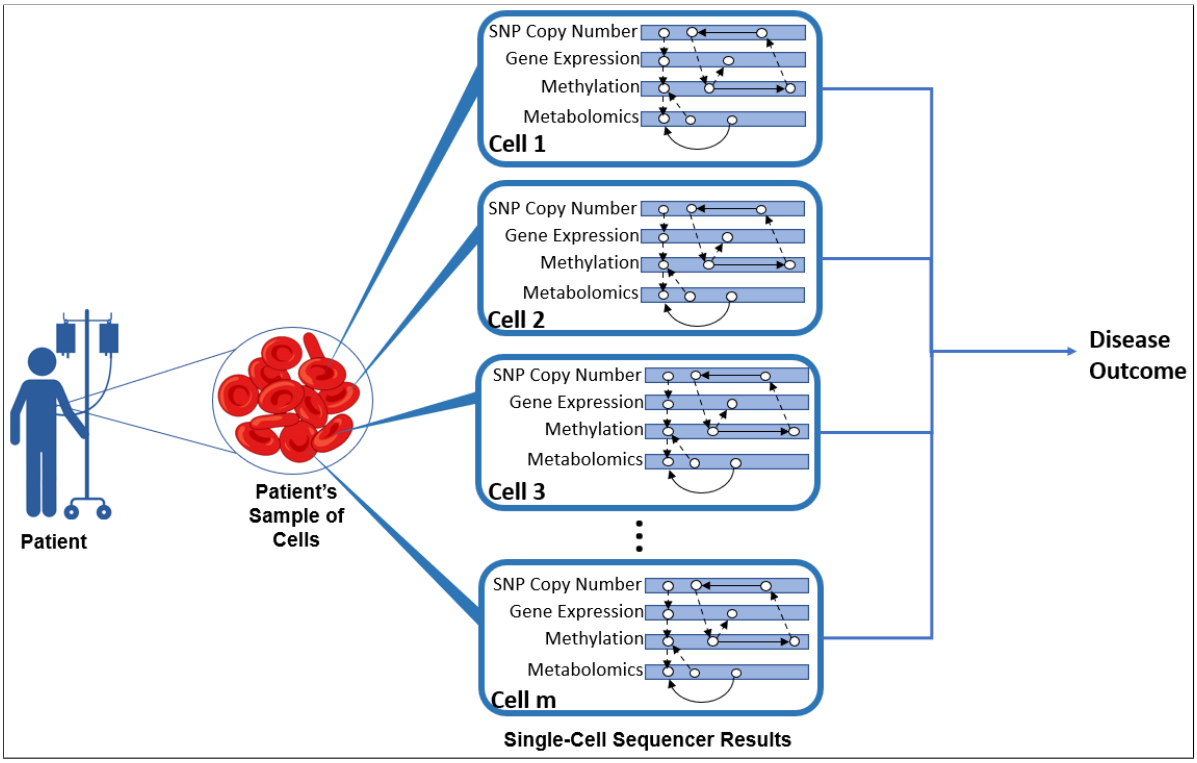

Figure 1: Pictoral demonstration of single-cell network detection experiments and studies.

This manuscript proposes a novel method, Multi-Omic Single-Cell Analysis using TensOr regression (MOSCATO), for identifying the superstructure of a semi-directed graph, or *network*, within multi-omic single-cell data that relates to a disease or phenotypic outcome. Section 2 describes preliminary tensor concepts, Section 3 introduces MOSCATO, Section 4 performs simulations of multi-omic single-cell data and applies MOSCATO under various scenarios, Section 5 applies MOSCATO to real single-cell data, and Section 6 discusses future work and limitations.

2 Preliminaries

MOSCATO utilizes regularized tensor regression, and this section describes existing and relevant tensor concepts. Section 2.1 defines tensors and basic tensor operations, and Section 2.2 uses the operations and definitions from Section 2.1 to describe tensor regression and regularization techniques.

2.1 Tensor Definitions

High dimensional data may be organized into a tensor, and a matrix may be thought of as a 2-dimensional tensor. Utilizing familiar tensor notation as provided by Kolda and Bader [16], we let $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{p_1 \times p_2 \times \cdots \times p_D}$ denote a D-dimensional tensor where dimension d contains p_d variables for $d = 1, ..., D$. For example, $D = 1$ denotes a vector and $D = 2$ denotes a matrix. Many mathematical operations for tensors build on mathematical operations used in matrices. For example, Definition 2.1 describes *outer products* between D vectors to create a D-dimensional tensor, where $\circ$ denotes the *Khatri-Rao product*.

Definition 2.1 *Let $\boldsymbol{b_1}, \boldsymbol{b_2}, ..., \boldsymbol{b_D}$, denote vectors where $\boldsymbol{b_d} \in \mathbb{R}^{p_d}$. Then the **outer product** of those vectors, $\boldsymbol{b_1} \circ \boldsymbol{b_2} \circ \cdots \circ \boldsymbol{b_D}$, creates a D-dimensional tensor of size $p_1 \times p_2 \times \cdots \times p_D$, and each $(i_1, ..., i_D)^{th}$ element equals $\prod_{d=1}^{D} b_{d_{i_d}}$.*

It may also be convenient to reorganize a tensor into a lower dimensional space by *vectorizing* or *mode-d matricizing* the tensor. Definitions 2.2 and 2.3 describe these reorganization techniques.

Definition 2.2 *Let $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{p_1 \times p_2 \times ... \times p_D}$ denote a D-dimensional tensor. Then $\boldsymbol{\mathcal{Z}}$ may be reorganized into a column vector through the **vec** operator $vec(\boldsymbol{\mathcal{Z}}) \in \mathbb{R}^{\prod_{d=1}^{D} p_d}$, where the $j = 1 + \sum_{d=1}^{D} (i_d - 1) \prod_{d'=1}^{d-1} p_{d'}$ element of $vec(\boldsymbol{\mathcal{Z}})$ corresponds to the $(i_1, ..., i_D)^{th}$ value in $\boldsymbol{\mathcal{Z}}$.*

Definition 2.3 *Let $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{p_1 \times p_2 \times ... \times p_D}$ denote a D-dimensional tensor. Then $\boldsymbol{\mathcal{Z}}$ may be reorganized into a matrix through the **mode-d matricization** operator $\boldsymbol{\mathcal{Z}}_{(d)} \in \mathbb{R}^{p_d \times \prod_{d' \neq d} p_{d'}}$, where the $(i_d, j)^{th}$ element within $\boldsymbol{\mathcal{Z}}_{(d)}$ equals the $(i_1, ..., i_D)^{th}$ value within $\boldsymbol{\mathcal{Z}}$ and $j = 1 + \sum_{d' \neq d} (i_{d'-1}) \prod_{d'' < d', d'' \neq d} p_{d''}$.*

Similarly as done in matrix operations, it may be of interest to multiply two tensors with comparable dimensions via *inner products*, as described in Definition 2.4.

Definition 2.4 *Suppose two tensors $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$ and $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$. The **inner product** may be obtained by*

$$\begin{aligned} \langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{Z}} \rangle &= \langle vec(\boldsymbol{\mathcal{B}}), vec(\boldsymbol{\mathcal{Z}}) \rangle \\ &= \sum_{i_1, ..., i_D} b_{i_1, ..., i_D} z_{i_1, ..., i_D}. \end{aligned} \tag{1}$$

Furthermore, it may be of interest to multiply a matrix along the d^{th} dimension of a tensor through *d-mode products* as described in Definition 2.5.

Definition 2.5 *Suppose a tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{p_1 \times \cdots \times p_d \times \cdots \times p_D}$ and a matrix $\boldsymbol{U} \in \mathbb{R}^{q \times p_d}$. The **d-mode product** between $\boldsymbol{\mathcal{Z}}$ and $\boldsymbol{U}$ may be expressed as $\boldsymbol{\mathcal{Z}} \times_d \boldsymbol{U} \in \mathbb{R}^{p_1 \times \cdots \times q \times \cdots \times p_D}$ where the $(i_1, ..., i_{d-1}, j, i_{d+1}, ..., i_D)^{th}$ value equals $\sum_{i_d=1}^{p_d} z_{i_1, ..., i_D} u_{j, i_d}$.*

The rank of a matrix denotes the maximum number of linearly independent rows/columns in the matrix. Building on those concepts, the rank of a tensor may be thought of as the maximum number of vectors that can be multiplied and added to replicate the tensor, as shown in Definition 2.6.

Definition 2.6 *Assume a D-dimensional tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{p_1 \times p_2 \times ... \times p_D}$. $\boldsymbol{\mathcal{Z}}$ is rank-R if there exists vectors $\boldsymbol{z_1^{(r)}} \in \mathbb{R}_1^p$, $\boldsymbol{z_2^{(r)}} \in \mathbb{R}_2^p$,...,$\boldsymbol{z_D^{(r)}} \in \mathbb{R}_D^p$ for $r = 1, 2, ..., R$ such that*

$$\begin{aligned} \boldsymbol{\mathcal{Z}} &= \sum_{r=1}^{R} \boldsymbol{z_1^{(r)}} \circ \boldsymbol{z_2^{(r)}} \circ \cdots \circ \boldsymbol{z_D^{(r)}} \\ &= [[\boldsymbol{Z}_1, \boldsymbol{Z}_2, ..., \boldsymbol{Z}_D]], \end{aligned} \tag{2}$$

where $\boldsymbol{Z}_d = [\boldsymbol{z_d^{(1)}}, ..., \boldsymbol{z_d^{(R)}}] \in \mathbb{R}^{p_d \times R}$.

The true rank of a tensor may often be difficult to determine due to the high dimensionality, motivating decomposition techniques that estimate vectors for a given rank that approximate the tensor, as shown in Definition 2.7.

Definition 2.7 *Assume a D-dimensional tensor* $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{p_1 \times p_2 \times \ldots \times p_D}$. *A* ***rank-R CP decomposition*** *aims to use R vector sets (one vector per dimension) to approximate* $\boldsymbol{\mathcal{Z}}$ *by*

$$\begin{aligned} \boldsymbol{\mathcal{Z}} &\approx \sum_{r=1}^{R} \boldsymbol{z}_{\boldsymbol{1}}^{(\boldsymbol{r})} \circ \boldsymbol{z}_{\boldsymbol{2}}^{(\boldsymbol{r})} \circ \cdots \circ \boldsymbol{z}_{\boldsymbol{D}}^{(\boldsymbol{r})} \\ &= [[\boldsymbol{Z}_1, \boldsymbol{Z}_2, ..., \boldsymbol{Z}_D]], \end{aligned} \tag{3}$$

where $\boldsymbol{Z}_d = [\boldsymbol{z}_{\boldsymbol{d}}^{(\boldsymbol{1})}, ..., \boldsymbol{z}_{\boldsymbol{d}}^{(\boldsymbol{R})}] \in \mathbb{R}^{p_d \times R}$.

Kolda and Bader present additional details on decomposition and other tensor operations [16].

2.2 Tensor Regression

Building on notation covered in Section 2.1, this section will briefly describe tensor regression that was originally presented by Zhou et al. [36]. Tensor regression builds on Generalized Linear Model (GLM) concepts, where we have an outcome y for each subject that follows some exponential family with link function $g(\cdot)$ and mean μ. Classic GLM uses univariate independent variables to predict the outcome, but tensor regression extends those concepts by additionally allowing a predictor tensor. This is accomplished by multiplying the dimensions of the tensor through coefficient vectors that convert the dimensions to a common univariate value that may then predict the outcome.

Figure 2 shows a simple example with rank-1 tensor regression and $D = 2$ dimensions for the predictor tensor. Each dimension in the predictor tensor corresponds to a different feature set, and each subject will contain its own $D = 2$ predictor tensor to be used to predict their univariate outcome y.

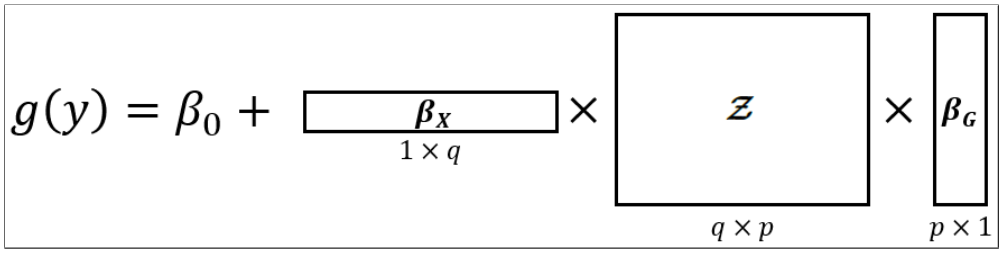

Figure 2: **Simple Example of Rank-1 Tensor Regression with $D = 2$.** Suppose a univariate outcome y with canonical link $g(\cdot)$ and predictor tensor $\boldsymbol{\mathcal{Z}}$ with $D = 2$ dimensions. Each dimension in the predictor tensor corresponds to a feature set, and each feature set contains its own coefficient vectors. The coefficient vectors in this example, $\boldsymbol{\beta_X}$ and $\boldsymbol{\beta_G}$, may be estimated by collecting outcomes and predictor tensors across multiple subjects and applying the Block Relaxation Alogorithm.

A motivating example for when these types of models may be useful is when each subject contains their own magnetic resonance imaging (MRI) image of their brains, and suppose each subject has an outcome specifying their disease status (e.g., brain tumor versus no brian tumor). These images may be expressed as a dataset (i.e., a 2-dimensional tensor) by organizing the images into compably sized grids where each grid point denotes a pixel from the image, and the value within each pixel quantifies the amount of pigment from the image. Referring to the model shown in Figure 2, the MRI image would correspond to the predictor tensor, $\boldsymbol{\mathcal{Z}}$, and coefficients would be estimated such that by inputting a subject's MRI image could then predict whether they had a brain tumor. The coefficient vectors in this example could help describe which

regions of the brain predict whether someone has a brain tumor (i.e., which rows/columns contain high/low coefficient values).

Tensor regression may also involve higher rank problems with the more formal representation

$$g(\mu) = \beta_0 + \boldsymbol{\lambda}^T \mathbf{U} + \langle \sum_{r=1}^{R} \boldsymbol{\beta}_{\mathbf{1}}^{(\boldsymbol{r})} \circ \boldsymbol{\beta}_{\mathbf{1}}^{(\boldsymbol{r})} \circ \cdots \circ \boldsymbol{\beta}_{\boldsymbol{D}}^{(\boldsymbol{r})}, \boldsymbol{\mathcal{Z}} \rangle \tag{4}$$

where $\mathbf{U}$ contains the univariate independent variables. A rank-R tensor regression estimates R coefficient vectors for each dimension in the predictor tensor, but for simplicity, in this manuscript we will assume rank-1. The *Block Relaxation Algorithm* is used to estimate the coefficient vectors with additional details described by Zhou et al. [36]. Zhou et al. [36] also claim that regularization in tensor regression may be accomplished by simply imposing constraints when fitting the models on each dimension.

If one naively vectorized the predictor tensor and fit a classic GLM model, it would require estimating $\prod_{d=1}^{D} p_d$ coefficients for the tensor. This approach would not only ignore the inherent structure of the data by treating each element in the tensor as independent with no distinction between the dimensions, it would also attempt to estimate many more coefficients compared to $R\sum_{d=1}^{D} p_d$ coefficients in tensor regression. Consequently, this naive approach may be unrealistic in high dimensional problems given a typically much smaller sample size. This reduction in parameters highlights the benefits of tensor regression. However, it is subject to limitations such as uniqueness and identifiability. For example, suppose a rank-1 model with $D = 2$, $g(y) = \boldsymbol{\beta_1}^T \boldsymbol{\mathcal{Z}} \boldsymbol{\beta_2}$. Then for any scalar τ, we could derive an equally optimal model $g(y) = \widetilde{\boldsymbol{\beta}}_{\mathbf{1}}^{\ T} \boldsymbol{\mathcal{Z}} \widetilde{\boldsymbol{\beta}}_{\mathbf{2}}$ where $\widetilde{\boldsymbol{\beta}}_{\mathbf{1}} = \tau \boldsymbol{\beta_1}$ and $\widetilde{\boldsymbol{\beta}}_{\mathbf{2}} = \boldsymbol{\beta_2}/\tau$. Additionally, the Block Relaxation Algorithm may converge to a local maxima as opposed to the global maxima when attempting to maximize the log likelihood. Measures to check for these concerns exist and are discussed in further detail by Zhou et al. [36].

3 Methods

3.1 The Model

In classic bulk sequencing, the data contains one record per subject. Supposing n subjects with two data types, bulk sequencing studies would contain two data sets (i.e., a dataset for each data type), $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{n \times p}$ and $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n \times q}$. In single-cell sequencing, there are multiple records per subject where each row corresponds to a cell within the subject. Consequently, for a given subject i with two data types, their single-cell data would contain two datasets $\boldsymbol{\mathcal{G}_i} \in \mathbb{R}^{m_i \times p}$ and $\boldsymbol{\mathcal{X}_i} \in \mathbb{R}^{m_i \times q}$, where m_i denotes the number of cells for subject i. Since the number of cells typically differs across subjects (i.e., $m_i \neq m_{i'}$ where $i \neq i'$), we organize each subject's data into separate datasets (i.e., $\boldsymbol{\mathcal{G}_i}$ and $\boldsymbol{\mathcal{X}_i}$) as opposed to organizing the input data directly into a 3-dimensional tensor with a dimension for cells. It should also be noted that each subject's data often consists of thousands of cells, and concatenating the single-cell data in long format may be computationally inefficient. Furthermore, we assume each subject i contains a univariate outcome y_i for $i = 1, ..., n$, and we may express the outcomes in a vector as $\mathbf{y} = [y_1, y_2, ..., y_n]^T$. For simplicity, we may denote the two data types as $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{X}}$ without the i subscript, although as described previously, each subject's single-cell data will contain separate matrices for the data types as opposed to expressing data in long format as found in bulk sequencing.

MOSCATO aims to identify a subset of features within $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{X}}$ that relate to each other and the outcome. In graphical modelling terms, MOSCATO identifies the superstructure of a semi-directed graph with undirected nodes involving features within $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{X}}$ with some path directed to the outcome $\mathbf{y}$. MOSCATO accomplishes this by imposing elastic net constraints on a tensor regression model [37].

Similarly as in the MRI image example from Section 2.2, multi-omic single-cell data contains multi-dimensional data per subject (i.e., features within $\boldsymbol{\mathcal{G}}$ and features within $\boldsymbol{\mathcal{X}}$) with a univariate outcome. This motivates the use of tensor regression for multi-omic single-cell data. Additionally, tensor regression

not only efficiently accommodates multi-dimensional input data with a univariate outcome, it also handles regularization techniques and allows for additional covariate adjustments (e.g., age, sex, race, etc.). However, tensor regression requires equivalent dimensions for each subject's input tensor. Thus, to standardize the dimensions across each subject, the first step of MOSCATO involves estimating a correlation matrix between their data type matrices,

$$\boldsymbol{\mathcal{Z}_i} = [\hat{\rho}_{jk}] \in \mathbb{R}^{q \times p}, \tag{5}$$

where $\hat{\rho}_{jk} = \hat{corr}(x_{ij}, g_{ik})$ letting x_{ij} denote the j^{th} feature in $\boldsymbol{\mathcal{X}_i}$ and g_{ik} denote the k^{th} feature in $\boldsymbol{\mathcal{G}_i}$ for the i^{th} subject. Although many summary matrices could be considered such as the inverse of the covariance matrix or mutual information, Pearson correlation provides a simple interpretation while also standardizing the values within $\boldsymbol{\mathcal{Z}_i}$ between -1 and 1.

Now using each of the $\boldsymbol{\mathcal{Z}_i}$ tensors for $i = 1, ..., n$ to estimate the coefficients, a tensor regression model similar to (4) and depicted in Figure 2 will be fit with elastic net constraints. The elastic net constraint works to balance by a weighted average between an L^1-norm and L^2-norm, where the L^1-norm truncates small coefficients to zero and the L^2-norm better handles highly correlated features. In summary, the elastic net constraint typically denoted as $\lambda((1-\alpha)/2||\boldsymbol{\beta}||_2^2 + \alpha||\boldsymbol{\beta}||_1)$ in the classical GLM setting will now involve

$$\begin{aligned} &((1-\alpha_X)/2||\boldsymbol{\beta_X}||_2^2 + \alpha_X||\boldsymbol{\beta_X}||_1), \\ &\sum_{j=1}^{q} I(\beta_{X_j} \neq 0) \leq max_X, \\ &((1-\alpha_G)/2||\boldsymbol{\beta_G}||_2^2 + \alpha_G||\boldsymbol{\beta_G}||_1), \\ &\sum_{k=1}^{p} I(\beta_{G_k} \neq 0) \leq max_G, \end{aligned} \tag{6}$$

where $\boldsymbol{\beta_X} \in \mathbb{R}^q$ denotes the coefficient vector for $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\beta_G} \in \mathbb{R}^p$ denotes the coefficient vector for $\boldsymbol{\mathcal{G}}$, β_{X_j} denotes the coefficient for the j^{th} feature in $\boldsymbol{\mathcal{X}}$, and β_{G_k} denotes the coefficient for the k^{th} feature in $\boldsymbol{\mathcal{G}}$. The hyperparameters $\alpha_X \in [0, 1]$ and $\alpha_G \in [0, 1]$ denote the weights to put on the L^1-norm constraints for $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{G}}$, respectively. The hyperparameter λ in the classical GLM setting denotes the overall weight to put on the constraint and it may be any positive number from 0 to infinity. Since tuning λ to the proper range may be difficult due to the nontrivial parameter space, we use max_X and max_G instead to denote the number of non-zero values within $\boldsymbol{\beta_X}$ and $\boldsymbol{\beta_G}$, respectively. This reparameterization of the constraints drastically simplifies the hyperparameter space and subsequent tuning described in Section 3.2.

For some fixed α_X, α_G, max_X, and max_G, the tensor regression model will be fit to obtain $\hat{\boldsymbol{\beta}}_{\boldsymbol{X}} \in \mathbb{R}^q$ and $\hat{\boldsymbol{\beta}}_{\boldsymbol{G}} \in \mathbb{R}^p$. Due to the L^1-norm truncating small values to zero from the elastic net constraint, only a subset of values within $\hat{\boldsymbol{\beta}}_{\boldsymbol{X}}$ and $\hat{\boldsymbol{\beta}}_{\boldsymbol{G}}$ will be nonzero. Thus, final network features within data type $\boldsymbol{\mathcal{X}}$ will be the set $\{j : \hat{\boldsymbol{\beta}}_{\boldsymbol{X_j}} \neq 0\}$, and final network features within data type $\boldsymbol{\mathcal{G}}$ will be the set $\{k : \hat{\boldsymbol{\beta}}_{\boldsymbol{G_k}} \neq 0\}$. Algorithm 1 summarizes the steps to MOSCATO.

3.2 Model Tuning

MOSCATO assumes fixed values for α_X, α_G, max_X, and max_G which will be tuned using an extension of the Stability Approach to Regularization Selection (StARS) method [22]. Tuning on accuracy, such as by cross validation or Bayesian information criterion, tends to result in overly dense solutions in high dimensional problems with results that are not reproducible [22]. The most extreme scenarios for stability are perfectly stable results from selecting no features (i.e., completely sparse) or selecting all features (i.e., no sparseness). Building on that logic, StARS initializes the parameters to the most sparse solution and gradually relaxes

Algorithm 1 Schematic for MOSCATO

Require: $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{G}}$, $\mathbf{y}$
  for $i = 1$ to n **do**
    $\boldsymbol{\mathcal{Z}_i} = Cor(\boldsymbol{\mathcal{X}_i}, \boldsymbol{\mathcal{G}_i})$
  end for
  Tune hyperparameters $\Lambda = \{\alpha_X, \alpha_G, max_X, max_G\}$ using Algorithm 2
  Fit $g(\mathbf{y}) = \beta_0 + \langle \boldsymbol{\beta_X} \circ \boldsymbol{\beta_G}, \boldsymbol{\mathcal{Z}} \rangle$, with elastic net penalties using the tuned Λ
  Network features within $\mathcal{X}$= $\{j : \hat{\boldsymbol{\beta}}_{\boldsymbol{X_j}} \neq 0\}$
  Network features within $\mathcal{G}$= $\{k : \hat{\boldsymbol{\beta}}_{\boldsymbol{G_k}} \neq 0\}$

the sparsity until some instability threshold ϕ is met. Instability is estimated based on subsamples from the data by performing the feature selection under each subsample and summarizing the consistancy in results across different subsamples. Although ϕ may initially be thought of as an arbitrary cutoff between 0 and 1, it may be easily interpretted as the amount of allowable instability. In essence, a smaller ϕ would imply a more sparse but stable result. The motivation behind allowing some instability as opposed to fixing ϕ to 0 is to allow some noise to be selected in order to ensure that no true signal is missed in the final feature selection. In statistical terms, this means that StARS prioritizes reducing type II errors.

Although the StARS method was developed for tuning a single sparsity parameter, the four hyperparameters α_X, α_G, max_X, and max_G will be tuned using similar logic. Focusing on tuning one dimension at a time, we initialize to a sparse solution with some small max_X. Fixing max_X, we estimate the instability for a range of α_X values between 0 and 1. Select the α_X value resulting in the lowest instability, and if that instability is less than ϕ, increase max_X and repeat the process. This continues until the ϕ instability is hit to select max_X and α_X. Using the highest max_X and corresponding optimal α_X with instability less than ϕ, a similar process is then repeated for tuning max_G and α_G. In this case with two dimensions, one for $\boldsymbol{\mathcal{X}}$ and another for $\boldsymbol{\mathcal{G}}$, we first tune α_X and max_X for some fixed α_G and max_G, and then use $\hat{max}_X$ and $\hat{\alpha}_X$ when tuning α_G and max_G. The initial fixed α_G and max_G may be kept large, suppose $\alpha_G = 0.5$ and $max_G = floor(p/2)$ such that an overly sparse $\boldsymbol{\mathcal{G}}$ does not impact the stability on $\boldsymbol{\mathcal{X}}$ for the first dimension of tuning. This is summarized in Algorithm 2.

4 Simulations

To benchmark MOSCATO's performance, it was applied to various simulations. The details on how the data was simulated is described in Section 4.1 and the results from MOSCATO are summarized in Section 4.2.

4.1 Simulation Details

Splatter [34] is a popular technique to simulate single-cell RNA-seq (scRNA-seq) data, and it has been shown to mimic distributions from real scRNA-seq data. The general Splatter schematic initiates by simulating a gene mean and then adjusts the gene mean to account for variation in outliers, library size, and dispersion. It then simulates the "observed" scRNA-seq values through a Poisson distribution using the adjusted gene mean, and the values are then randomly truncated to zero to replicate dropouts. Although Splatter realistically portrays scRNA-seq distributions, a few extensions were required in order to simulate multi-omic single-cell data with supervised gene networks.

To accomplish this, we leverage latent structures for multi-omic supervised networks detailed in Zhang et al. [35]. Figure 3 demonstrates the expected causal relationships within the data containing a supervised multi-omic network. In summary, we expect there to be a subgroup of features within $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{X}}$ that relate to the outcome but not with each other; a subgroup of features within $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{X}}$ that relate to each other but not with the outcome; a subgroup of features within $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{X}}$ that are independent of each other and

Algorithm 2 StARS Method for Tuning Tensor Hyperparameter

Require: $\boldsymbol{\mathcal{G}}$, $\boldsymbol{\mathcal{X}}$, $\mathbf{y}$, $grid_\alpha$, ϕ, S, c, max_{G_0}, max_{X_0}

Generate S random samples, each of size c
Initialize max_G to $floor(0.5 * n)$ and $\alpha_G = 0.5$
for $max_X = max_{X_0}$ to q **do**
  Initialize $\hat{\alpha}_X = grid_{\alpha_1}$
  for α_X in $grid_\alpha$ **do**
    $\Lambda_1 = \{\alpha_G, \alpha_X, max_G, max_X\}$
    for $s = 1$ to S **do**
      $\hat{\boldsymbol{\beta}}_{\boldsymbol{X}_s}(\alpha_X)$ = fitted coefficient vector using Λ_1 and subsample s
    end for
    $\hat{\theta}_{X_j}(\Lambda_1) = 1/S(\sum_{s=1}^{S} I(\hat{\boldsymbol{\beta}}_{\boldsymbol{X}_j s}(\alpha_X) \neq 0))$, $j = 1, ..., q$
    $\hat{\xi}_{X_j}(\Lambda_1) = 2\hat{\theta}_{X_j}(\Lambda_1)(1 - \hat{\theta}_{X_j}(\Lambda))$, $j = 1, ..., q$
    $\hat{D}_X(\Lambda_1) = 1/q \sum_{j=1}^{q} \hat{\xi}_{X_j}(\Lambda_1)$
    if $\alpha_X = grid_{\alpha_1}$ **then**
      $D_{optimal} = \hat{D}_X(\Lambda_1)$
      $\hat{\Lambda} = \Lambda_1$
    else
      if $\hat{D}_X(\Lambda_1) < D_{optimal}$ **then**
        $D_{optimal} = \hat{D}_X(\Lambda_1)$
        $\hat{\alpha}_X = \alpha_X$
        $\hat{\Lambda} = \Lambda_1$
      end if
    end if
  end for
  if $D_{optimal} > \phi$ **then**
    $\hat{\Lambda} = \Lambda_0$
    break
  else
    $\Lambda_0 = \hat{\Lambda}$
  end if
end for
Repeat the process to estimate $\hat{max}_G$ and $\hat{\alpha}_G$ using optimal $\hat{max}_X$ and $\hat{\alpha}_X$

the outcome; and finally the target subgroups of the analysis, subgroup of network features within $\mathcal{G}$ that relate to the network features within $\mathcal{X}$ that ultimately relates to the outcome. These expected relationships among these latent components may be represented through a covariance matrix where the off diagonals will be non-zero where relationships exist and zero where independence is expected. More details are provided in the Appendix.

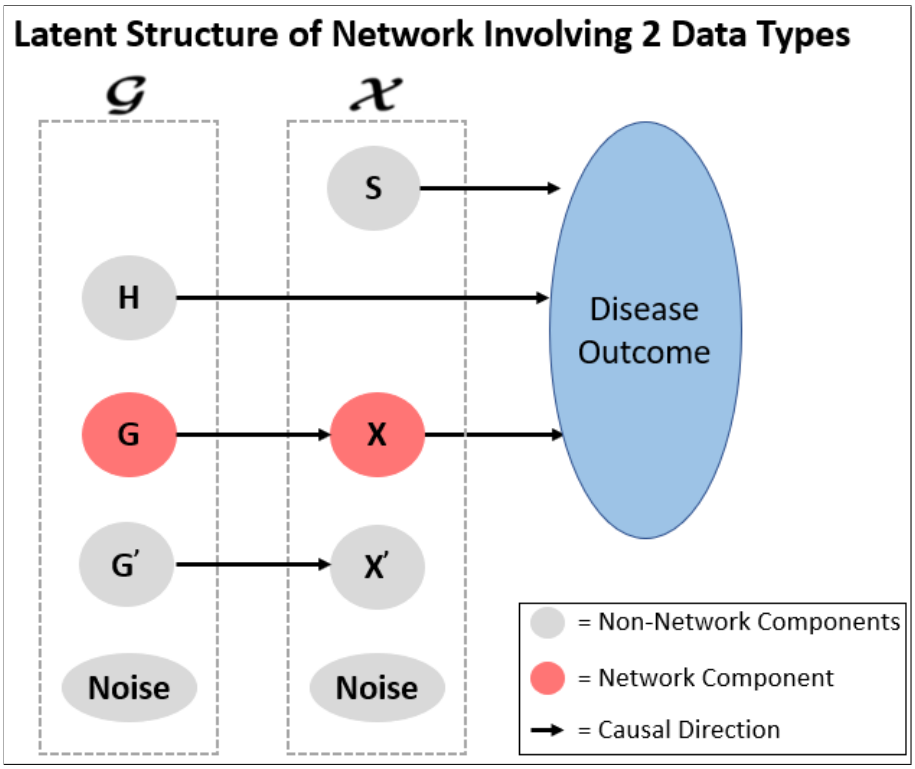

Figure 3: **Latent structure for supervised multi-omic networks.** From [30] and adapted from [35]. Assume two data types, $\mathcal{G}$ and $\mathcal{X}$. **S** describes the subgroup of features within $\mathcal{X}$ that relate to the disease outcome but not with any features within $\mathcal{G}$; **H** describes the subgroup of features within $\mathcal{G}$ that relate to the disease outcome but not with any features within $\mathcal{X}$; **G** and **X** describe the network where the group of features within $\mathcal{G}$ relate to the group of features within $\mathcal{X}$ that ultimately relate to the outcome; $\mathbf{G}'$ and $\mathbf{X}'$ denote the group of features that relate to each other but not with the outcome; and finally each data type will have independent noise not related to each other or the outcome.

To extend Splatter for supervised multi-omic networks, we will first simulate the latent values (S, H, X, G, G', X', and y) for each subject using a multivariate normal distribution with zero mean and the latent variables' covariance matrix. Since the latent values were simulated from a multivariate normal, they will each be normally distributed with mean 0. The Splatter simulation assumes the initial gene mean comes from a gamma distribution, so we take the square of the latent value divided by its standard deviation. By doing so, the transformed latent values then become gamma distributed with shape equal to 1/2 and scale equal to 2 times its variance. These transformed latent values will be used as the initial gene means for each subgroup of features, and the latent value for y will be used as the mean to simulate an observed outcome from a normal distribution. Initial gene means for the noise subgroup of features are randomly simulated independently. Additional theoretical details may be found in the Appendix.

Dispersion is adjusted on a subject level, library size is adjusted on a cellular level within a subject, outliers are adjusted on a feature level within a subject, and dropouts are accounted for on a cellular/feature

level within a subject. The Splatter simulation is performed using the transformed gene means (combining all latent components within $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{X}}$), and then the features are later separated by data type for each subject.

4.2 Results

MOSCATO was applied to a series of simulations using the techniques described in Section 4.1. Simulations were performed under 9 different settings accounting for the average number of cells per subject (250, 500, or 1000) and amount of technical noise (low, moderate, or high). Simulations were replicated 50 times under each simulation setting and each simulation had 100 subjects. $\boldsymbol{\mathcal{G}}$ contained 1440 total features where only 10 belonged to the network, and $\boldsymbol{\mathcal{X}}$ contained 1555 total features where only 15 belonged to the network. Table 1 describes the total number of features contained within each of the latent subgroups described in Figure 3.

Table 1: **Number of Features in Simulations**

Latent Set of Features	# Features
$\mathbf{H}$	15
$\mathbf{G}$	10
$\mathbf{G}'$	15
Noise in $\mathcal{G}$	1400
$\mathbf{S}$	20
$\mathbf{X}$	15
$\mathbf{X}'$	20
Noise in $\mathcal{X}$	1500

The table summarizes the number of features within each latent subgroup used in the simulations. These latent subgroups are displayed in Figure 3. $\mathbf{H}$ describes the subgroup of features within $\mathcal{G}$ that relate to the outcome but not any features within $\mathcal{X}$, $\mathbf{G}$ describes the subgroup of features within $\mathcal{G}$ belonging to the network, $\mathbf{G}'$ describes the subgroup of features within $\mathcal{G}$ related to some features within $\mathcal{X}$ but not the outcome, and $\mathbf{G}'$ describes the subgroup of features within $\mathcal{G}$ unrelated to features within $\mathcal{X}$ and the outcome. $\mathbf{S}$, $\mathbf{X}$, $\mathbf{X}'$, and Noise in $\mathcal{G}$ describe subgroups of features within $\mathcal{X}$ with similar relationships as the subgroups within $\mathcal{G}$.

For tuning the hyperparameters, we set $grid_\alpha = \{0.2, 0.5, 0.7\}$, $\phi = 0.02$, $R = 50$, and $c = 50$. max_{G_0} and max_{X_0} were initially set to 5, although this number may be increased in order to reduce runtimes as long as the instability remains below ϕ for its initialization. Ideally, MOSCATO would tune max_G to 10 and max_X to 15 in order to select the proper network size according to Table 1, but this will be unlikely due to the mechanics behind the StARS tuning method described in Section 3.2 which prioritizes reducing type II error over type I error. Figure 4 displays the tuned max_G and max_X across the simulations. As expected, all simulations tuned max_G and max_X to values greater than the true number of network features, regardless of the simulation setting. For max_X, smaller values were tuned as technical noise decreased and number of cells increased, but this trend did not persist when tuning max_G.

In addition to applying MOSCATO, we also applied competing methods using the area under the receiver operating curve (AUC). Seurat provides a popular single-cell sequencing workflow, and following similar methods used by the authors of Seurat [10], this AUC approach was done using the presto version 1.0.0 R package [19]. Selections using AUC were performed using two different criteria. One criteria was based on whether the Bonferroni adjusted p-value was less than the nominal significance level (set to 0.05) under the null hypothesis that the AUC equals 0.5. Additionally, selection criteria using cutoff values where features with an AUC either less than 0.3 or greater than 0.7 were selected. Since AUC requires categorical outcomes, we use the median of the outcome to binarize it (i.e., if the outcome is less than the median then recode the

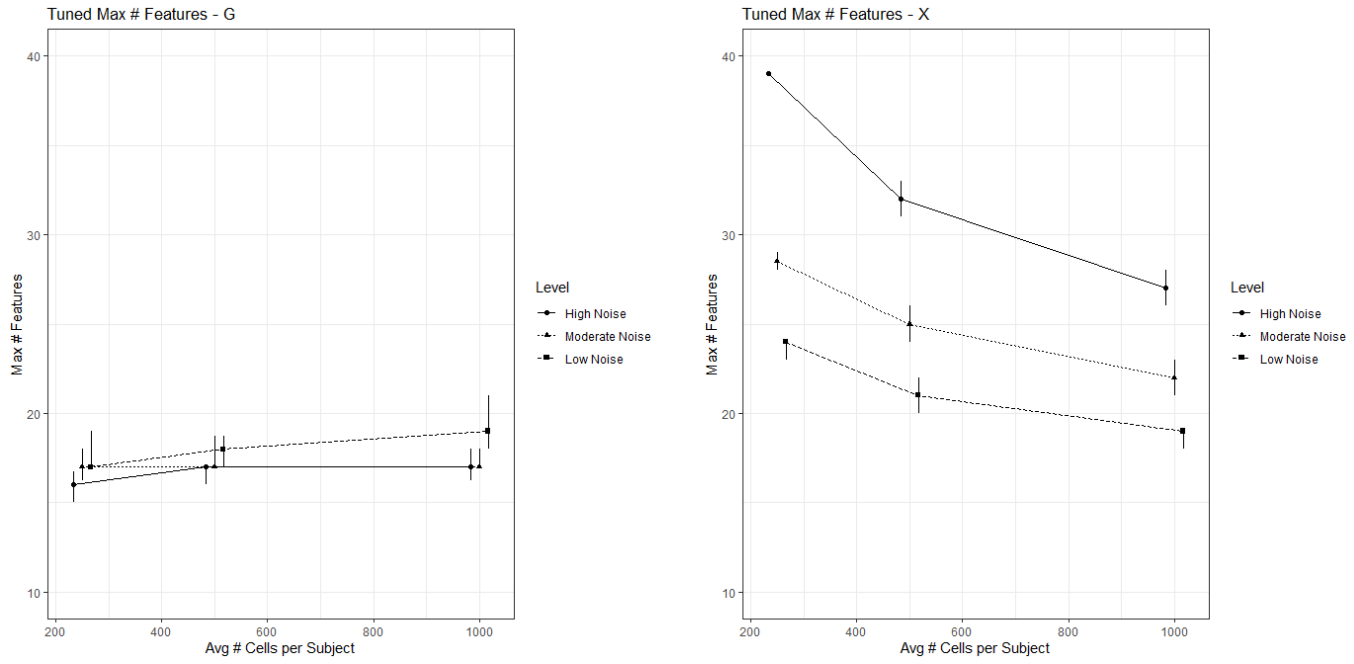

Figure 4: This figure displays the tuned max_G and max_X hyperparameters using the StARS method across the 9 different simulation settings accounting for different numbers of average cells per subject and different levels of technical noise. The points represent the median tuned values and the bars represent the first and third quartiles across 50 iterations for each simulation setting. (A) displays the tuned max network size within $\boldsymbol{\mathcal{G}}$, and (B) displays the tuned max network size within $\boldsymbol{\mathcal{X}}$. Ideally, MOSCATO would tune max_G to 10 and max_X to 15 in order to select the proper network size according to Table 1.

outcome as '0', otherwise if the outcome is greater than the median then recode as '1').

Figure 5 displays the sensitivity and specificity across the 9 simulation settings for data types $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{X}}$ for the 3 different methods (MOSCATO, AUC using p-values, and AUC using cutoffs). Sensitivity measures the probability that network features are properly included in the selections, and specificity measures the probability that non-network features are properly excluded from the selections. As shown in Figure 5, the sensitivity and specificity under MOSCATO generally improve as the number of cells increases per subject and as technical noise decreases. Conversely, the specificity declines for AUC selections using p-values as the number of cells increases and the technical noise improves. AUC based on p-values not only produced counterintuitive results where the performance actually degraded as the technical noise reduced, it also selected too many features such that the results were not remotely sparse. This explains that while the sensitivity remained high for all simulations, this is simply due to the fact that nearly all features were selected using that criteria. AUC selections based on cutoffs resulted in opposite issues where it did not select nearly any features and produced poor sensitivity with nearly perfect specificity.

MOSCATO reproduced the superstructure of the network reasonably well with generally high sensitivity and also limited false positives present. This is especially true when comparing against approaches using the AUC. However, when it is expected that high levels of technical noise is present with limited cells per subject, caution should be used when considering MOSCATO.

5 Real Data Example

Leukemia encompasses all cancers that occur in blood cells. The 5-year survival rate is about 65% according to data from 2011 to 2017, and about 459,000 people were living with leukemia in 2018 in the United States [12]. Leukemia may be classified based on progression speed where chronic denotes slow progression and acute denotes aggressive progression. In addition to cancer progression, Leukemia may be subtyped by the type of cells where the cancer forms. For example, lymphocytic leukemia describes cancer developing from white blood cells and myelogenous leukemia describes cancer developing in blood forming cells within the bone marrow. Although rare, one may also have both lymphocytic and myelogenous leukemia which is denoted as mixed phenotype leukemia.

To assess MOSCATO in practice, we applied it to real single-cell data with multiple data types. Limited data is currently available due to the infancy of multi-omic single-cell sequencing, so data across multiple studies that all used CITE-seq protocols [29] on bone marrow / peripheral blood cells were used. CITE-seq produces cellular level RNA information and cell surface protein abundance (i.e., antibody derived tags (ADT)) simultaneously, and our outcome of interest will be leukemia versus healthy patients. Our goal will be to apply MOSCATO to this data in order to obtain a subset of RNA and ADT features associated with leukemia. After combining the data across studies, we have 14 healthy patients and 7 patients with leukemia. Of the 7 leukemia subjects, 1 had chronic lymphocytic leukemia (CLL) while the other 6 had mixed-phenotype acute leukemia (MPAL). The studies used to obtain the data are summarized in Table 2. Only non-perturbed, baseline cells were considered.

Table 2: **Studies Used with CITE-seq Protocols and Healthy or Leukemia Subjects**

Study ID	Tissue Type	# Healthy Subjects	# Leukemia Subjects
ERP124005 [21]	Blood	10	0
GSE152469 [4]	Blood	0	1
GSE139369 [9]	Blood	1	4
GSE139369 [9]	Bone Marrow	3	2

Seurat version 4.0.3 [10] was used to normalize the data, cluster cells, and integrate the cell types across subjects. Figure 6 displays the Uniform Manifold Approximation and Projection (UMAP) [2] plots from the integrated data.

After integration, 8 of the cell clusters (clusters 0, 1, 2, 3, 4, 5, 6 and 14 from Figure 6), 17991 RNA features, and 5 ADTs (CD3, CD4, CD14, CD19, and CD56) were measured across all subjects. We applied MOSCATO to each of these cell clusters separately, each with $grid_\alpha = \{0.01, 0.05, 0.1\}$. Since Seurat clusters cells by maximizing correlation between features, the multicollinearity across features would consequently be high and require more weight on the L^2-norm (i.e., lower α within the elastic net constraint).

Due to the modest sample size, we tuned the hyperparameters using a subsampling size of 20 (out of 21 total subjects) to estimate stability based on a "leave one out" scheme. Although StARS suggests setting the instability threshold ϕ to 0.05 for most applications [22], in this application with a small sample size (i.e., only 21 subsamples) and large number of RNA features (i.e., 17991 variables), the estimated instability under most sparse solutions will be much smaller than 0.05. For example, suppose all 21 subsamples select completely disjoint feature sets, but due to the high number of variables in consideration, many variables are consistently excluded from any selections in across all of the subsampled results. Since StARS considers both consistency in selections and consistency in exclusions, the estimated instability will be quite small due to the consistency in exclusions despite that the small number of features selected may be completely disjoint across all subsamples. Therefore, ϕ was set to 0.001.

To compare selections with another method, the MOSCATO results were compared to selections based on the AUC. The AUC approach was done using the presto version 1.0.0 R package [19]. Similarly as was done for MOSCATO, the AUC feature selections were performed on each cell cluster separately. Feature selections were made under two different selection criteria for AUC: if the Bonferroni adjusted p-value was

less than 0.05 under the null hypothesis that the AUC equals 0.5 or whether the AUC was less tha 0.3 or greater than 0.7. Since the p-value would likely be small in situations with many cells (i.e., large sample sizes producing sensitive p-values for miniscule AUC deviations from 0.5), both a p-value approach and an approach based on the AUC values were considered.

5.1 Results

The complete results from MOSCATO, AUC selections based on p-values, and AUC selections based on the AUC cutoffs for each of the 8 cell clusters are provided in the supplementary files. In summary, the number of features selected by MOSCATO and AUC cutoffs were similarly sized, but AUC selections based on p-values resulted in nonsparse feature sets. DAVID [11, 28] was used to analyze and organize the gene ontology information from the RNA gene selections. DAVID clusters genes based on common annotations and functional information, and DAVID only allows clustering on gene sets with less than 3000 genes. Since the AUC selections based on p-values resulted in RNA selections well over this 3000 restriction for most cell clusters, we only focused on gene clusters from the MOSCATO and AUC cutoff selections.

MOSCATO selected 96 RNA features within cell cluster 0, and DAVID identified 2 gene clusters. The strongest gene cluster (based on highest enrichment score) used 7 of these 96 genes. This was the most notable gene cluster which included the genes CD3D, CD3E, and CD3G which are part of the KEGG pathway for Human T-cell Leukaemia Virus type 1 (HTLV-I) infection (KEGG pathway hsa05166), and HTLV-I infections are a known risk factor for developing adult T-cell leukaemia/lymphoma (ATL) [13]. Additionally, the genes CD2, CD3D, CD3E, CD3G, and CD4 within this gene cluster belong to the KEGG pathway for Hematopoietic cell lineage (hsa04640) which assists in producing blood cells. Given that the disease of interest in this application is based on leukemia (i.e., cancer in tissues which produce blood), it is reassuring that this gene cluster contains genes associated with blood production. Also, genes CD2, CD3D, CD3E, CD3G, KLRB1, CD247, and CD4 within the gene cluster are associated with the gene ontology for the cell surface receptor signaling pathway (GO:0007166) which makes sense given that MOSCATO summarized the single-cell data between RNA and cell surface proteins. Of the 5 cell surface proteins considered, MOSCATO selected the ADT's CD3 and CD4 for this cell cluster. Figure 7 displays the RNA features within the strongest functional DAVID cluster, along with the ADT selections. No features selected by MOSCATO under cell cluster 0 were selected by AUC cutoffs, and although 83 of the 96 MOSCATO RNA selections were also selected by AUC based on p-values, the AUC selections based on p-values selected nearly half of all RNA features considered. AUC selections based on cutoffs selected 11 RNA features and 0 ADTs for cell cluster 0, and DAVID was not able to discern any gene clusters based on the genes selected.

In conclusion, the selections made by MOSCATO under cell cluster 0 resulted in a concise gene cluster discovered by previously known annotations and functionalities. These functionalities not only related to the disease of interest (i.e., leukemia), it also related to cell surface functionalities. This highlights that MOSCATO not only considers supervised information (e.g., disease versus no disease), it also considers the relationships across data types (e.g., RNA and cell surface proteins). Performing feature selection using solely AUC not only neglects the cross data type relationships, it also was not able to return a concise set of genes that related to leukemia. Furthermore, it is arbitrary to select pre-specified AUC cutoffs for selections, and the p-value selections did not produce sparse solutions. Also, AUC selections were based directly on normalized expression/sequenced values, but MOSCATO performs selections based on the similarities between data types. This possibly helps reduce batch effects found in individual subjects by standardizing each value between -1 and 1.

The real data application may be reproduced by following the steps provided at `https://github.com/lorinmil/MOSCATOLeukemiaExample`.

6 Discussion/Conclusions

MOSCATO was performed on both simulated and real data. The simulations produced fairly accurate results with a sensitivity and specificity close to 1 for many of the simulations, although MOSCATO did not perform as well in situations with high technical noise or low cell counts per subject. Since MOSCATO calculates its predictor tensor to be the estimated correlation of the datasets per individual, it is unsurprising that cell count would contribute to more accurate correlation estimates. Although not used in this manuscript, covariate adjustments could easily be made in MOSCATO by simply adding them to the tensor regression.

MOSCATO currently assumes all cells come from the same cell type, but it might be more interesting to accommodate situations in which multiple cell types are present. We are currently working on higher dimensional applications of MOSCATO in a future manuscript. A reasonable solution could incorporate another step which estimates a similarity matrix for each cell type and includes another dimension to the predictor tensor for cell type. Additionally, MOSCATO was only tested on experiments with 2 data types, but extensions should be considered in situations where more than two data types are present. This could be accomplished by extending the predictor tensor to accommodate a dimension per data type extracted, although higher dimensional summary measures would need to be considered.

MOSCATO was only tested using Pearson's correlation as the summary measure to construct the predictor tensor $\boldsymbol{\mathcal{Z}}$, but other summaries should be considered. For example, mutual information or inverse covariance matrices might be interesting avenues to explore for future consideration. Graphical lasso [8] is a popular technique to obtain both the nodes and edges of a graph by applying lasso regressions to estimate the inverse of the covariance matrix, and this estimated inverse of the covariance matrix could be explored as the predictor tensor input for MOSCATO.

This study only used rank-1 tensors, but higher ranks could be considered that may unveil other patterns and networks available in the data. The proper rank could be obtained using typical model selection criteria such as cross validation, although interpreting the results may not be as straight forward.

Zhou et al. discuss hypothesis testing via asymptotic normality results [36], and these hypothesis testing schemes could be explored to assess network strength. For example, one could perform a global test whether the model coefficients equal zero for the network selections.

Although MOSCATO returns the superstructure for a graph, it does not provide information on directionality and does not currently consider directional consistency. For example, suppose two genes are positively correlated with another, but they contain opposing correlative directions with the outcome. This inconsistency in directionality makes interpreting the results more difficult, and may be mitigated by additionally tuning based on optimizing *balance*. This concept has been considered in bulk level analyses with a single omic type [31], and it could be considered for future work.

In summary, MOSCATO is a useful tool to indicate multi-omic features that relate to disease outcomes of interest to better accommodate multi-omic, single-cell data which is continuously growing in popularity.

7 Availability

All code in this manuscript was done using R version 4.1.0 [27]. Code to perform MOSCATO and replicate the simulations may be found publicly on GitHub at `https://github.com/lorinmil/MOSCATO`. Steps and code to replicate the real data application may be found at `https://github.com/lorinmil/MOSCATOLeukemiaExample`.

References

[1] A. Balmain, J. Gray, and B. Ponder. The Genetics and Genomics of Cancer. *Nature Genetics*, 33(3):238–244, 2003.

[2] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I. W. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell. Di-

mensionality Reduction for Visualizing Single-Cell Data Using UMAP. *Nature Biotechnology*, 37(1):38–44, 2019.

[3] D. Benatar, M. Bondmass, J. Ghitelman, and B. Avitall. Outcomes of Chronic Heart Failure. *Archives of Internal Medicine*, 163(3):347–352, 2003.

[4] S. Cadot, C. Valle, M. Tosolini, F. Pont, L. Largeaud, C. Laurent, J. J. Fournie, L. Ysebaert, and A. Quillet-Mary. Longitudinal CITE-Seq Profiling of Chronic Lymphocytic Leukemia During ibrutinib Treatment: Evolution of Leukemic and Immune Cells at Relapse. *Biomarker Research*, 8(1):1–13, 2020.

[5] W. Cookson, L. Liang, G. Abecasis, M. Moffatt, and M. Lathrop. Mapping Complex Disease Traits with Global Gene Epression. *Nature Reviews Genetics*, 10(3):184–194, 2009.

[6] P. Creixell, J. Reimand, S. Haider, G. Wu, T. Shibata, M. Vazquez, V. Mustonen, A. Gonzalez-Perez, J. Pearson, C. Sander, et al. Pathway and Network Analysis of Cancer Genomes. *Nature Methods*, 12(7):615, 2015.

[7] A. M. Elissen, L. M. Steuten, L. C. Lemmens, H. W. Drewes, K. M. Lemmens, J. A. Meeuwissen, C. A. Baan, and H. J. Vrijhoef. Meta-Analysis of the Effectiveness of Chronic Care Management for Diabetes: Investigating Heterogeneity in Outcomes. *Journal of Evaluation in Clinical Practice*, 19(5):753–762, 2013.

[8] J. Friedman, T. Hastie, and R. Tibshirani. Sparse Inverse Covariance Estimation with the Graphical Lasso. *Biostatistics*, 9(3):432–441, 2008.

[9] J. M. Granja, S. Klemm, L. M. McGinnis, A. S. Kathiria, A. Mezger, M. R. Corces, B. Parks, E. Gars, M. Liedtke, G. X. Zheng, et al. Single-Cell Multiomic Analysis Identifies Regulatory Programs in Mixed-Phenotype Acute Leukemia. *Nature Biotechnology*, 37(12):1458–1465, 2019.

[10] Y. Hao, S. Hao, E. Andersen-Nissen, W. M. Mauck III, S. Zheng, A. Butler, M. J. Lee, A. J. Wilk, C. Darby, M. Zagar, P. Hoffman, M. Stoeckius, E. Papalexi, E. P. Mimitou, J. Jain, A. Srivastava, T. Stuart, L. B. Fleming, B. Yeung, A. J. Rogers, J. M. McElrath, C. A. Blish, R. Gottardo, P. Smibert, and R. Satija. Integrated Analysis of Multimodal Single-Cell Data. *Cell*, 2021.

[11] D. W. Huang, B. T. Sherman, and R. A. Lempicki. Bioinformatics Enrichment Tools: Paths Toward the Comprehensive Functional Analysis of Large Gene Lists. *Nucleic Acids Research*, 37(1):1–13, 2009.

[12] S. R. P. S. in NCI's Division of Cancer Control and P. S. (DCCPS). Cancer Stat Facts: Leukemia. https://seer.cancer.gov/statfacts/html/leuks.html, 2021. [Online; accessed 25-August-2021].

[13] K. Ishitsuka and K. Tamura. Human T-cell Leukaemia Virus Type I and Adult T-cell Leukaemia-lymphoma. *The Lancet Oncology*, 15(11):e517–e526, 2014.

[14] G. Karlebach and R. Shamir. Modelling and Analysis of Gene Regulatory Networks. *Nature Reviews Molecular Cell Biology*, 9(10):770–780, 2008.

[15] A. R. Kendal, T. Layton, H. Al-Mossawi, L. Appleton, S. Dakin, R. Brown, C. Loizou, M. Rogers, R. Sharp, and A. Carr. Multi-Omic Single Cell Analysis Resolves Novel Stromal Cell Populations in Healthy and Diseased Human Tendon. *Scientific Reports*, 10(1):1–14, 2020.

[16] T. G. Kolda and B. W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, 2009.

[17] N. L. Komarova and C. J. Thalhauser. High Degree of Heterogeneity in Alzheimer's Disease Progression Patterns. *PLoS Computational Biology*, 7(11):e1002251, 2011.

[18] K. Komurov, J.-T. Tseng, M. Muller, E. G. Seviour, T. J. Moss, L. Yang, D. Nagrath, and P. T. Ram. The Glucose-Deprivation Network Counteracts Lapatinib-Induced Toxicity in Resistant ErbB2-Positive Breast Cancer Cells. *Molecular Systems Biology*, 8(1):596, 2012.

[19] I. Korsunsky, A. Nathan, N. Millard, and S. Raychaudhuri. Presto Scales Wilcoxon and auROC Analyses to Millions of Observations. *BioRxiv*, page 653253, 2019.

[20] P. Langfelder and S. Horvath. Wgcna: An R Package for Weighted Correlation Network Analysis. *BMC Nioinformatics*, 9(1):1–13, 2008.

[21] N. Lawlor, D. Nehar-Belaid, J. D. Grassmann, M. Stoeckius, P. Smibert, M. L. Stitzel, V. Pascual, J. Banchereau, A. Williams, and D. Ucar. Single Cell Analysis of Blood Mononuclear Cells Stimulated Through Either LPS or Anti-CD3 and Anti-CD28. *Frontiers in Immunology*, 12:691, 2021.

[22] H. Liu, K. Roeder, and L. Wasserman. Stability approach To Regularization Selection (StARS) for High Dimensional Graphical Models. *Advances in Neural Information Processing Systems*, 24(2):1432, 2010.

[23] E. Z. Macosko, A. Basu, R. Satija, J. Nemesh, K. Shekhar, M. Goldman, I. Tirosh, A. R. Bialas, N. Kamitaki, E. M. Martersteck, et al. Highly Parallel Genome-Wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell*, 161(5):1202–1214, 2015.

[24] M. I. McCarthy. Genomics, Type 2 Diabetes, and Obesity. *New England Journal of Medicine*, 363(24):2339–2350, 2010.

[25] C. J. O'Donnell and E. G. Nabel. Genomics of Cardiovascular Disease. *New England Journal of Medicine*, 365(22):2098–2109, 2011.

[26] S. Picelli, O. R. Faridani, Å. K. Björklund, G. Winberg, S. Sagasser, and R. Sandberg. Full-length RNA-seq from Single Cells Using Smart-seq2. *Nature Protocols*, 9(1):171–181, 2014.

[27] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021.

[28] B. T. Sherman, R. A. Lempicki, et al. Systematic and Integrative Analysis of Large Gene Lists Using DAVID Bioinformatics Resources. *Nature Protocols*, 4(1):44–57, 2009.

[29] M. Stoeckius, C. Hafemeister, W. Stephenson, B. Houck-Loomis, P. K. Chattopadhyay, H. Swerdlow, R. Satija, and P. Smibert. Simultaneous Epitope and Transcriptome Measurement in Single Cells. *Nature Methods*, 14(9):865–868, 2017.

[30] L. M. Towle-Miller, J. C. Miecznikowski, F. Zhang, and D. L. Tritchler. Sumo-fil: Supervised multi-omic filtering prior to performing network analysis. *Plos One*, 16(8):e0255579, 2021.

[31] D. Tritchler, L. M. Towle-Miller, and J. C. Miecznikowski. Balanced Functional Module Detection in Genomic Data. *bioRxiv*, 2020.

[32] N. C. Turner and J. S. Reis-Filho. Genetic Heterogeneity and Cancer Drug Resistance. *The Lancet Oncology*, 13(4):e178–e185, 2012.

[33] D. M. Witten and R. J. Tibshirani. Extensions of Sparse Canonical Correlation Analysis with Applications to Genomic Data. *Statistical Applications in Genetics and Molecular Biology*, 8(1):1–27, 2009.

[34] L. Zappia, B. Phipson, and A. Oshlack. Splatter: Simulation of Single-Cell RNA Sequencing Data. *Genome Biology*, 18(1):174, 2017.

[35] F. Zhang, J. C. Miecznikowski, and D. L. Tritchler. Identification of Supervised and Sparse Functional Genomic Pathways. *Statistical Applications in Genetics and Molecular Biology*, 19(1), 2020.

[36] H. Zhou, L. Li, and H. Zhu. Tensor Regression with Applications in Neuroimaging Data Analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.

[37] H. Zou and T. Hastie. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

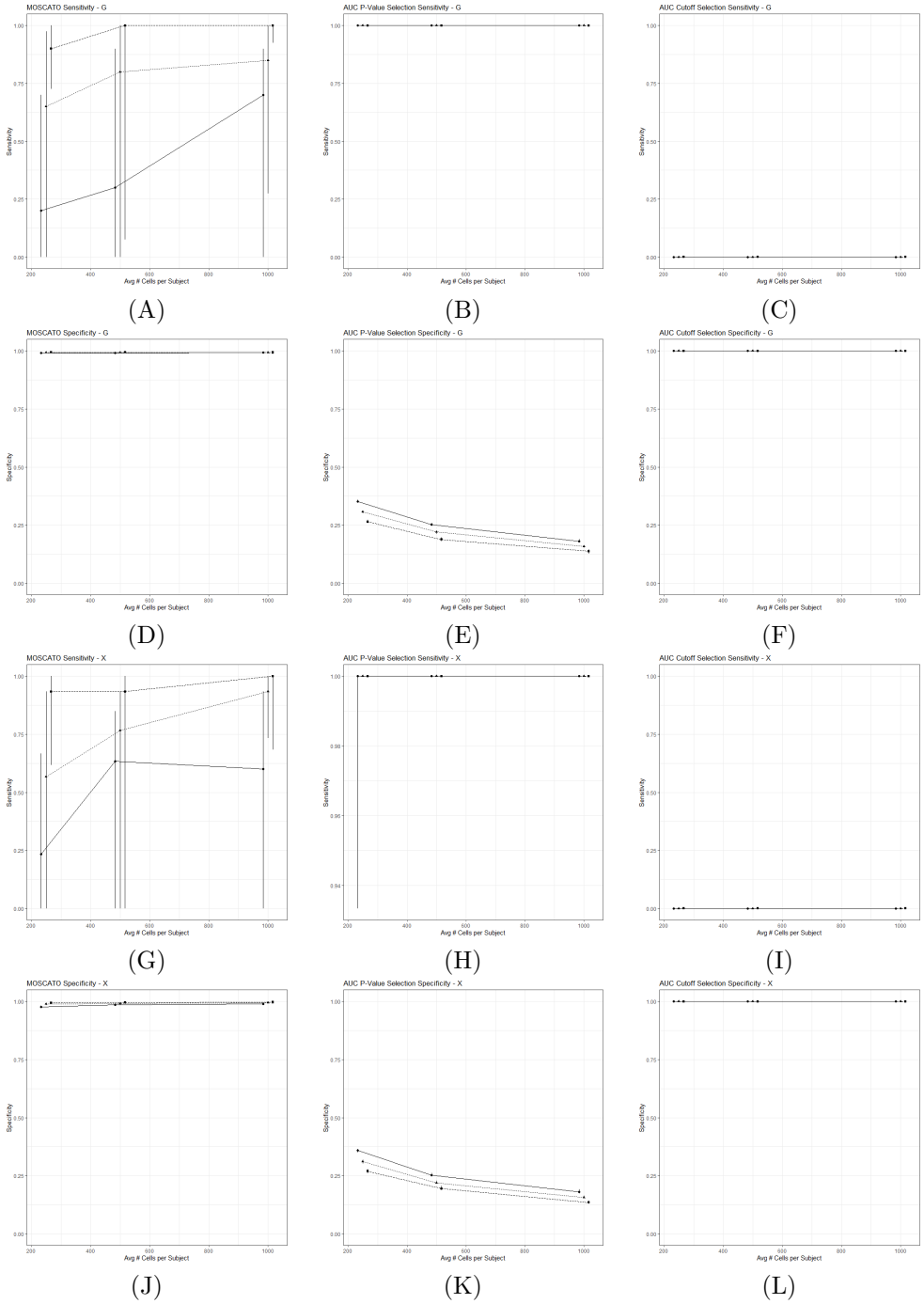

Figure 5: This figure displays the sensitivity and specificity across the 9 different simulation settings accounting for different numbers of average cells per subject and different levels of technical noise. The points represent the median sensitivity/specificity and the bars represent the first and third quartiles across 50 iterations for each simulation setting. (A)-(C) display the sensitivity for $\boldsymbol{\mathcal{G}}$ using MOSCATO, AUC selections using Bonferroni adjusted p-values, and AUC selections using cutoffs (< 0.3 or > 0.7). (D)-(F) display the specificity for $\boldsymbol{\mathcal{G}}$ under the three methods in the same order. Similarly, (G)-(I) displays the sensitivity for $\boldsymbol{\mathcal{X}}$ and (J)-(L) displays the specificity for $\boldsymbol{\mathcal{X}}$. Under perfect selections, the sensitivity and specificity should equal 1.

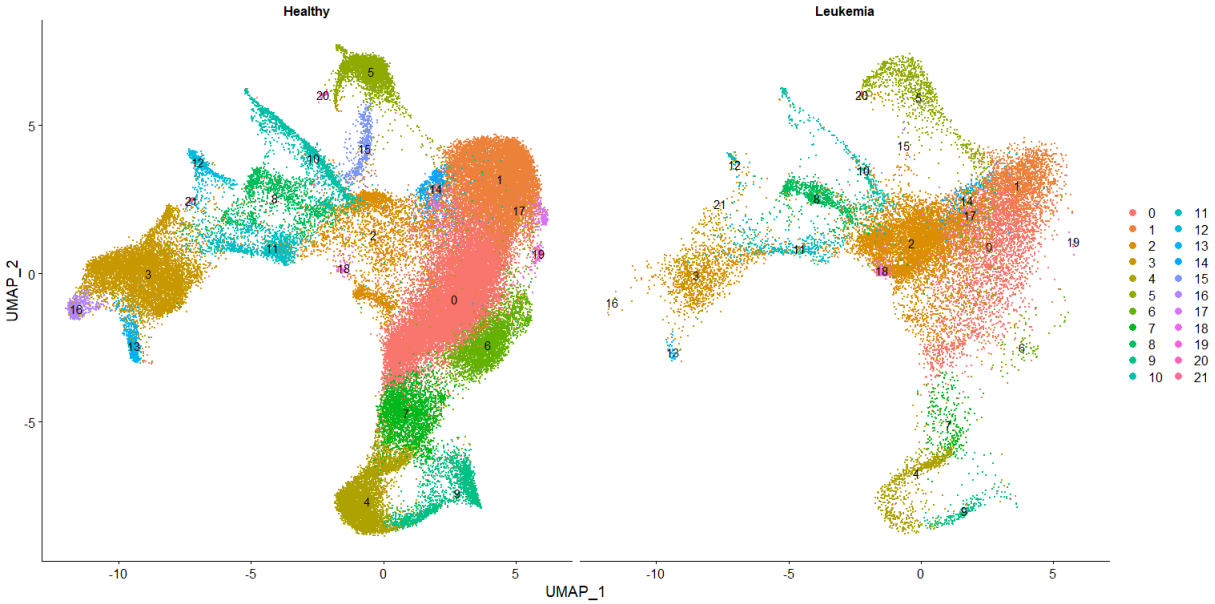

Figure 6: The UMAP results from the integrated cell types. The plots are split by healthy versus leukemia subjects.

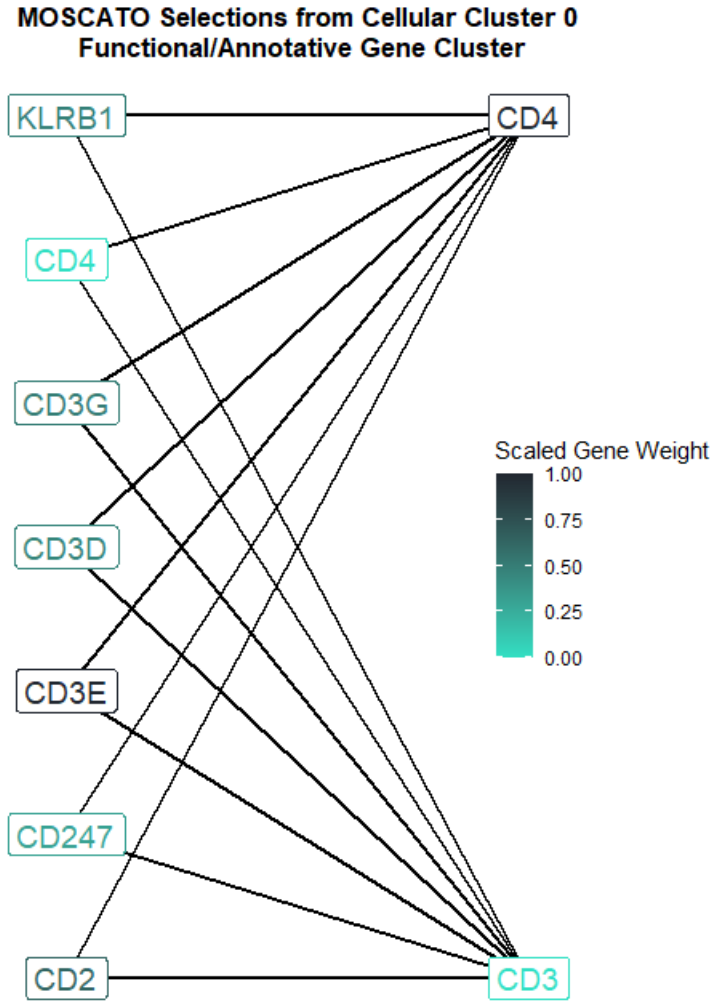

Figure 7: MOSCATO selected 96 RNA features and 2 ADT features from cell cluster 0 shown in Figure 6. This figure displays the strongest functional RNA gene cluster (left labels) within the 96 MOSCATO selections as determined by DAVID [11, 28], along with all ADT selections (right labels). DAVID determines gene clusters based on similar gene ontology and annotations. These RNA genes correspond to KEGG pathways for blood cell production (CD2, CD3D, CD3E, CD3G, and CD4) and HTLV-I infection (CD3D, CD3E, and CD3G), as well as genes with ontology information for cell surface receptor signaling (CD2, CD3D, CD3E, CD3G, KLRB1, CD247, and CD4). The label colors display the absolute scaled weight of the coefficient vectors from the tensor regression used in MOSCATO so that higher values correspond to a higher weight put on that gene/protein in the tensor regression.