Variable Selection for High-Dimensional Longitudinal Omics Data with a Continuous or Misclassified Binary Outcome

Steve Broll

July 2025

Alternative Title: TB or not TB

General Motivation

Collaborator gives you hundreds (or thousands) of longitudinal -omics variables and a clinical outcome of interest, with the motivating question depending on the type of outcome:

- Longitudinal continuous outcome, e.g. some measure of disease severity
  - Which omic variables co-vary over time with the outcome?
- Binary outcome that could be misclassified, e.g. disease recurrence or an indicator for some condition
  - Which omic variables are associated with the (latent) outcome?

General Motivation

Collaborator gives you hundreds (or thousands) of longitudinal -omics variables and a clinical outcome of interest, with the motivating question depending on the type of outcome:

- Longitudinal continuous outcome, e.g. some measure of disease severity
  - Which omic variables co-vary over time with the outcome?
- Binary outcome that could be misclassified, e.g. disease recurrence or an indicator for some condition
  - Which omic variables are associated with the (latent) outcome?

General Motivation

Collaborator gives you hundreds (or thousands) of longitudinal -omics variables and a clinical outcome of interest, with the motivating question depending on the type of outcome:

- Longitudinal continuous outcome, e.g. some measure of disease severity
    - Which omic variables co-vary over time with the outcome?
- Binary outcome that could be misclassified, e.g. disease recurrence or an indicator for some condition
    - Which omic variables are associated with the (latent) outcome?

General Motivation

- High-dimensional omics data is recently more feasible to collect and will be increasingly common in clinical data
- Variables selected from the -omics pool could be used as non-invasive markers for early disease progression, and provide insight into biological processes
- There is extensive work in the omics arena, especially in differential expression/abundance between groups. However, we have identified two main gaps in the longitudinal omics setting.

General Motivation

- High-dimensional omics data is recently more feasible to collect and will be increasingly common in clinical data
- Variables selected from the -omics pool could be used as non-invasive markers for early disease progression, and provide insight into biological processes
- There is extensive work in the omics arena, especially in differential expression/abundance between groups. However, we have identified two main gaps in the longitudinal omics setting.

General Motivation

- High-dimensional omics data is recently more feasible to collect and will be increasingly common in clinical data
- Variables selected from the -omics pool could be used as non-invasive markers for early disease progression, and provide insight into biological processes
- There is extensive work in the omics arena, especially in differential expression/abundance between groups. However, we have identified two main gaps in the longitudinal omics setting.

Targeted Gaps

For longitudinal continuous outcomes,

- clinicians typically apply univariate linear mixed effects models or generalized estimating equations, depending on the magnitude of n
- We propose a joint model that leverages correlation over time as well as correlation between variables to select a sparse set of biomarkers
- We provide a framework for uncertainty quantification AND inference with two or more treatment groups

Targeted Gaps

For longitudinal continuous outcomes,

- clinicians typically apply univariate linear mixed effects models or generalized estimating equations, depending on the magnitude of n
- We propose a joint model that leverages correlation over time as well as correlation between variables to select a sparse set of biomarkers
- We provide a framework for uncertainty quantification AND inference with two or more treatment groups

Targeted Gaps

For longitudinal continuous outcomes,

- clinicians typically apply univariate linear mixed effects models or generalized estimating equations, depending on the magnitude of n
- We propose a joint model that leverages correlation over time as well as correlation between variables to select a sparse set of biomarkers
- We provide a framework for uncertainty quantification AND inference with two or more treatment groups

Targeted Gaps

For binary outcomes,

- Some methods handle potential misclassification, and others address variable selection for large sets of omics variables, but no methods deal with both simultaneously
- We provide a penalized EM algorithm that accounts for covariate-related misclassification that leverages the between-variable correlations and can perform variable selection

Targeted Gaps

For binary outcomes,

- Some methods handle potential misclassification, and others address variable selection for large sets of omics variables, but no methods deal with both simultaneously
- We provide a penalized EM algorithm that accounts for covariate-related misclassification that leverages the between-variable correlations and can perform variable selection

Roadmap

1. First-differencing longitudinal omics data and incorporating observed covariate dependence via weighted graph Laplacian
2. PROLONG - **P**enalized **r**egression on **o**utcome-guided **l**ongitudinal **o**mics data with **n**etwork and **g**roup constraints
3. Uncertainty quantification in a network-constrained sparse group lasso model for outcome-guided high-dimensional omics data with multiple treatment groups
4. Penalized logistic regression on binary clinical outcomes with potential covariate-related misclassification, with application to longitudinal omics data

Roadmap

1. First-differencing longitudinal omics data and incorporating observed covariate dependence via weighted graph Laplacian
2. PROLONG - **P**enalized **r**egression on **o**utcome-guided **l**ongitudinal **o**mics data with **n**etwork and **g**roup constraints
3. Uncertainty quantification in a network-constrained sparse group lasso model for outcome-guided high-dimensional omics data with multiple treatment groups
4. Penalized logistic regression on binary clinical outcomes with potential covariate-related misclassification, with application to longitudinal omics data

Roadmap

1. First-differencing longitudinal omics data and incorporating observed covariate dependence via weighted graph Laplacian
2. PROLONG - **P**enalized **r**egression on **o**utcome-guided **l**ongitudinal **o**mics data with **n**etwork and **g**roup constraints
3. Uncertainty quantification in a network-constrained sparse group lasso model for outcome-guided high-dimensional omics data with multiple treatment groups
4. Penalized logistic regression on binary clinical outcomes with potential covariate-related misclassification, with application to longitudinal omics data

Roadmap

1. First-differencing longitudinal omics data and incorporating observed covariate dependence via weighted graph Laplacian
2. PROLONG - **P**enalized **r**egression on **o**utcome-guided **l**ongitudinal **o**mics data with **n**etwork and **g**roup constraints
3. Uncertainty quantification in a network-constrained sparse group lasso model for outcome-guided high-dimensional omics data with multiple treatment groups
4. Penalized logistic regression on binary clinical outcomes with potential covariate-related misclassification, with application to longitudinal omics data

Section 1 - First Differencing and graph Laplacian

1. First-differencing longitudinal omics data and incorporating observed covariate dependence via weighted graph Laplacian
2. PROLONG - **P**enalized **r**egression on **o**utcome-guided **l**ongitudinal **o**mics data with **n**etwork and **g**roup constraints
3. Uncertainty quantification in a network-constrained sparse group lasso model for outcome-guided high-dimensional omics data with multiple treatment groups
4. Penalized logistic regression on binary clinical outcomes with potential covariate-related misclassification, with application to longitudinal omics data

First-Differencing

- Capture trends over time, de-emphasize baseline value (intercept)
- Controls first-order time dependence in the outcome
- Remove any time-invariant heterogeneous effects, simplifying the model parameters
- Analogous to running a paired test

First-Differencing

- Capture trends over time, de-emphasize baseline value (intercept)
- Controls first-order time dependence in the outcome
- Remove any time-invariant heterogeneous effects, simplifying the model parameters
- Analogous to running a paired test

First-Differencing

- Capture trends over time, de-emphasize baseline value (intercept)
- Controls first-order time dependence in the outcome
- Remove any time-invariant heterogeneous effects, simplifying the model parameters
- Analogous to running a paired test

First-Differencing

- Capture trends over time, de-emphasize baseline value (intercept)
- Controls first-order time dependence in the outcome
- Remove any time-invariant heterogeneous effects, simplifying the model parameters
- Analogous to running a paired test

Graph Laplacian Penalty

- The graph Laplacian is often used when there is a known graph associated with the data. Lacking a prior graph, we use a weighted graph extracted from the observed absolute correlation matrix
- Leverages potentially useful correlation information in predictors by nudging coefficients for correlated variables together (smoothing over dependence network)
- Alleviates numerical challenges typical to lasso type problems in presence of multicollinearity
- If two variables are identical, the respective coefficients will be identical
- The penalization allows us to obtain a maximum likelihood estimator (MLE) for the noise variance $\hat{\sigma}^2$ of Y

Graph Laplacian Penalty

- The graph Laplacian is often used when there is a known graph associated with the data. Lacking a prior graph, we use a weighted graph extracted from the observed absolute correlation matrix
- Leverages potentially useful correlation information in predictors by nudging coefficients for correlated variables together (smoothing over dependence network)
- Alleviates numerical challenges typical to lasso type problems in presence of multicollinearity
- If two variables are identical, the respective coefficients will be identical
- The penalization allows us to obtain a maximum likelihood estimator (MLE) for the noise variance $\hat{\sigma}^2$ of Y

Graph Laplacian Penalty

- The graph Laplacian is often used when there is a known graph associated with the data. Lacking a prior graph, we use a weighted graph extracted from the observed absolute correlation matrix
- Leverages potentially useful correlation information in predictors by nudging coefficients for correlated variables together (smoothing over dependence network)
- Alleviates numerical challenges typical to lasso type problems in presence of multicollinearity
- If two variables are identical, the respective coefficients will be identical
- The penalization allows us to obtain a maximum likelihood estimator (MLE) for the noise variance $\hat{\sigma}^2$ of Y

Graph Laplacian Penalty

- The graph Laplacian is often used when there is a known graph associated with the data. Lacking a prior graph, we use a weighted graph extracted from the observed absolute correlation matrix
- Leverages potentially useful correlation information in predictors by nudging coefficients for correlated variables together (smoothing over dependence network)
- Alleviates numerical challenges typical to lasso type problems in presence of multicollinearity
- If two variables are identical, the respective coefficients will be identical
- The penalization allows us to obtain a maximum likelihood estimator (MLE) for the noise variance $\hat{\sigma}^2$ of Y

Graph Laplacian Penalty

- The graph Laplacian is often used when there is a known graph associated with the data. Lacking a prior graph, we use a weighted graph extracted from the observed absolute correlation matrix
- Leverages potentially useful correlation information in predictors by nudging coefficients for correlated variables together (smoothing over dependence network)
- Alleviates numerical challenges typical to lasso type problems in presence of multicollinearity
- If two variables are identical, the respective coefficients will be identical
- The penalization allows us to obtain a maximum likelihood estimator (MLE) for the noise variance $\hat{\sigma}^2$ of Y

Grouping Effect Example

Theorem
(Grouping Effect) *Given data (Y, X) and fixed scalar λ, let $\hat{\beta}(\lambda)$ be a Laplacian-penalized estimator. Suppose $\hat{\beta}_i \hat{\beta}_j > 0$ and the two vertices i and j are only linked to each other on the network, $d_i = d_j = w(i,j)$. Define*

$$D_\lambda(i,j) = \frac{1}{||Y||_1} \left| \hat{\beta}_i(\lambda) - \hat{\beta}_j(\lambda) \right|.$$

then

$$D_\lambda(i,j) \leq \frac{\sqrt{2(1-\rho)}}{2\lambda}$$

where $||Y||_1 = \sum_{i=1}^n |Y_i|$ and $\rho = X_{\cdot i}^T X_{\cdot j}$ captures the sample correlation.

Section 2 - PROLONG

1. First-differencing longitudinal omics data and incorporating observed covariate dependence via weighted graph Laplacian
2. PROLONG - **P**enalized **r**egression on **o**utcome-guided **l**ongitudinal **o**mics data with **n**etwork and **g**roup constraints
3. Uncertainty quantification in a network-constrained sparse group lasso model for outcome-guided high-dimensional omics data with multiple treatment groups
4. Penalized logistic regression on binary clinical outcomes with potential covariate-related misclassification, with application to longitudinal omics data

Structure of the Data

- Outcome is TB mycobacterial load measured by Time to Positivity (TTP), which is inversely related to TB severity
- 15 TB patients taking the same combination drug RHEZ
- 4 time points, one at baseline and 3 following RHEZ treatment
- 352 untargeted urinary metabolites as our predictors
- These untargeted metabolites are typically known to have low signal and high noise

Structure of the Data

- Outcome is TB mycobacterial load measured by Time to Positivity (TTP), which is inversely related to TB severity
- 15 TB patients taking the same combination drug RHEZ
- 4 time points, one at baseline and 3 following RHEZ treatment
- 352 untargeted urinary metabolites as our predictors
- These untargeted metabolites are typically known to have low signal and high noise

Structure of the Data

- Outcome is TB mycobacterial load measured by Time to Positivity (TTP), which is inversely related to TB severity
- 15 TB patients taking the same combination drug RHEZ
- 4 time points, one at baseline and 3 following RHEZ treatment
- 352 untargeted urinary metabolites as our predictors
- These untargeted metabolites are typically known to have low signal and high noise

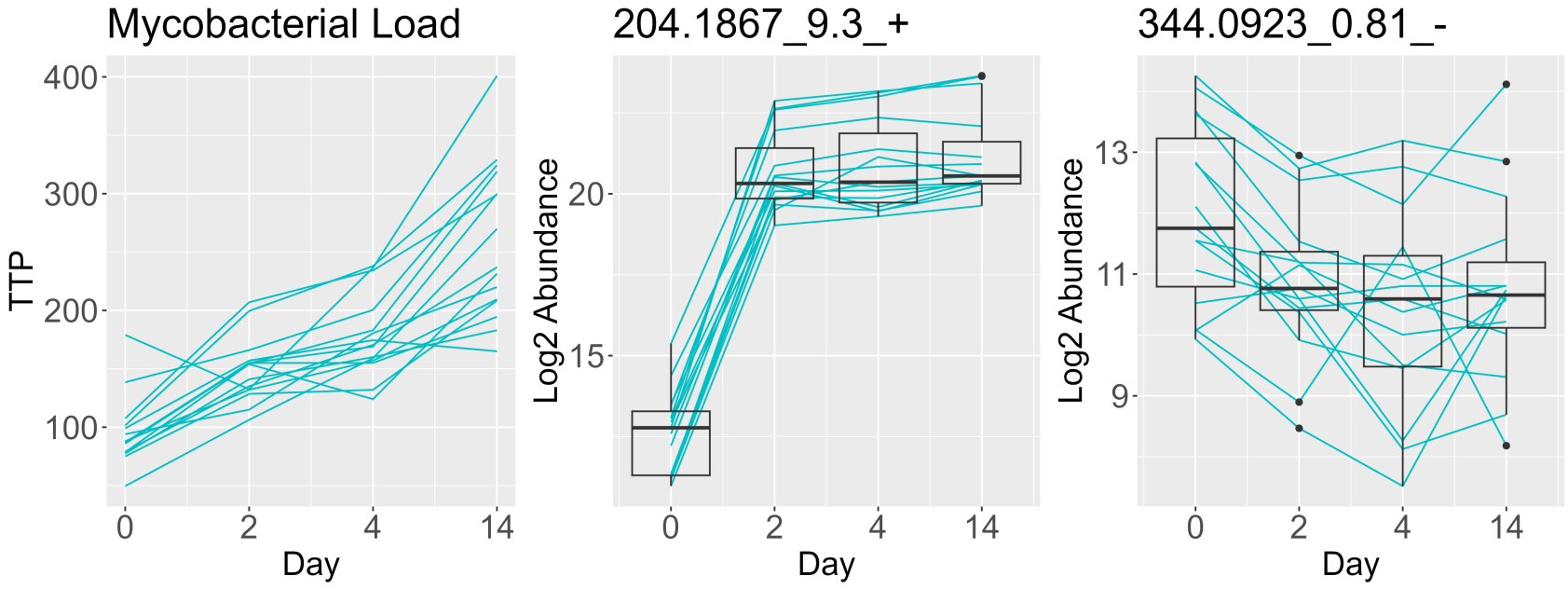
Structure of the Data

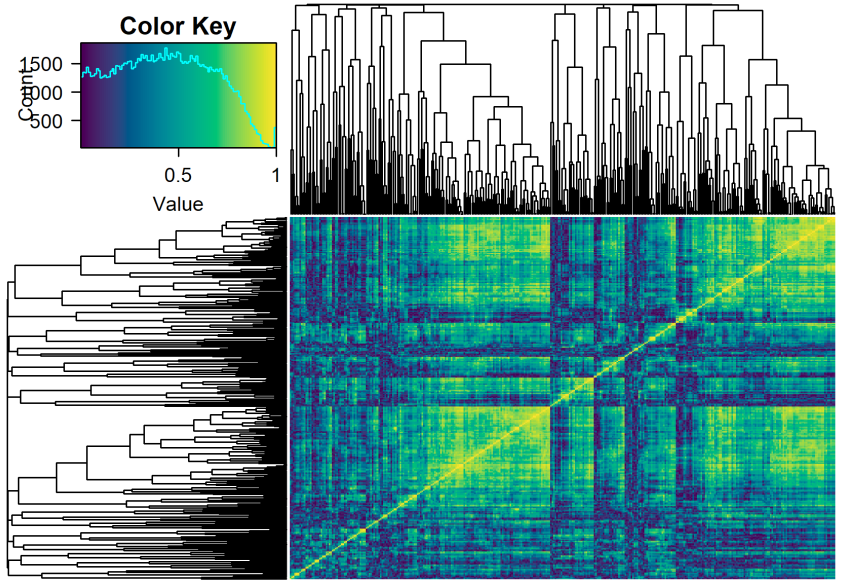
- Outcome is TB mycobacterial load measured by Time to Positivity (TTP), which is inversely related to TB severity
- 15 TB patients taking the same combination drug RHEZ
- 4 time points, one at baseline and 3 following RHEZ treatment
- 352 untargeted urinary metabolites as our predictors
- These untargeted metabolites are typically known to have low signal and high noise

Structure of the Data

- Outcome is TB mycobacterial load measured by Time to Positivity (TTP), which is inversely related to TB severity
- 15 TB patients taking the same combination drug RHEZ
- 4 time points, one at baseline and 3 following RHEZ treatment
- 352 untargeted urinary metabolites as our predictors
- These untargeted metabolites are typically known to have low signal and high noise

Tuberculosis Data





General Idea

- First-difference the data, then stack our $t-1$ values of X and Y so we have

$$Y = [Y_4 - Y_3 \qquad Y_3 - Y_2 \qquad Y_2 - Y_1]^T$$

  And for each variable j we have

$$X_j = [X_{j4} - X_{j3} \qquad X_{j3} - X_{j2} \qquad X_{j2} - X_{j1}]^T$$

- Set up design matrix so that each first-differenced Y value is regressed on all prior first-differenced values of X to account for potential lags
- Apply Laplacian and group lasso penalties to induce sparsity while utilizing correlation and inherent group structure

General Idea

- First-difference the data, then stack our $t-1$ values of X and Y so we have

$$Y = [Y_4 - Y_3 \qquad Y_3 - Y_2 \qquad Y_2 - Y_1]^T$$

And for each variable j we have

$$X_j = [X_{j4} - X_{j3} \qquad X_{j3} - X_{j2} \qquad X_{j2} - X_{j1}]^T$$

- Set up design matrix so that each first-differenced Y value is regressed on all prior first-differenced values of X to account for potential lags
- Apply Laplacian and group lasso penalties to induce sparsity while utilizing correlation and inherent group structure

General Idea

- First-difference the data, then stack our $t-1$ values of X and Y so we have

$$Y = [Y_4 - Y_3 \qquad Y_3 - Y_2 \qquad Y_2 - Y_1]^T$$

And for each variable j we have

$$X_j = [X_{j4} - X_{j3} \qquad X_{j3} - X_{j2} \qquad X_{j2} - X_{j1}]^T$$

- Set up design matrix so that each first-differenced Y value is regressed on all prior first-differenced values of X to account for potential lags
- Apply Laplacian and group lasso penalties to induce sparsity while utilizing correlation and inherent group structure

Vectorized Y

$$\tilde{Y} = \begin{bmatrix} \tilde{Y}_{11} & \cdots & \tilde{Y}_{1T} \\ & \vdots & \\ \tilde{Y}_{n1} & \cdots & \tilde{Y}_{nT} \end{bmatrix}_{n \times T} \rightarrow \begin{bmatrix} \Delta \tilde{Y}_{11} & \cdots & \Delta \tilde{Y}_{1(T-1)} \\ & \vdots & \\ \Delta \tilde{Y}_{n1} & \cdots & \Delta \tilde{Y}_{n(T-1)} \end{bmatrix}_{n \times (t-1)}$$

$$\rightarrow Y = \begin{bmatrix} \Delta \tilde{Y}_{11} \\ \vdots \\ \Delta \tilde{Y}_{n1} \\ \Delta \tilde{Y}_{1(T-1)} \\ \vdots \\ \Delta \tilde{Y}_{n(T-1)} \end{bmatrix}_{n(T-1) \times 1}$$

Moving X from Tensor to Matrix

$$\tilde{X}^{[j]} = \begin{bmatrix} \tilde{X}^{[j]}_{11} & \cdots & \tilde{X}^{[j]}_{1T} \\ & \vdots & \\ \tilde{X}^{[j]}_{n1} & \cdots & \tilde{X}^{[j]}_{nT} \end{bmatrix}_{n \times T} \rightarrow \begin{bmatrix} \Delta\tilde{X}^{[j]}_{11} & \cdots & \Delta\tilde{X}^{[j]}_{1(T-1)} \\ & \vdots & \\ \Delta\tilde{X}^{[j]}_{n1} & \cdots & \Delta\tilde{X}^{[j]}_{n(T-1)} \end{bmatrix}_{n \times (T-1)}$$

$$\rightarrow X^{[j]} = \left[\begin{array}{c|cc|c|ccc} \Delta\tilde{X}^{[j]}_{11} & & & & & & \\ \vdots & 0 & & 0 & & 0 & \\ \Delta\tilde{X}^{[j]}_{n1} & & & & & & \\ \hline 0 & \Delta\tilde{X}^{[j]}_{11} & \Delta\tilde{X}^{[j]}_{12} & & & & \\ & \vdots & & 0 & & 0 & \\ & \Delta\tilde{X}^{[j]}_{n1} & \Delta\tilde{X}^{[j]}_{n2} & & & & \\ \hline 0 & 0 & & \ddots & & 0 & \\ \hline & & & & \Delta\tilde{X}^{[j]}_{11} & \cdots & \Delta\tilde{X}^{[j]}_{1(T-1)} \\ 0 & 0 & & 0 & & \vdots & \\ & & & & \Delta\tilde{X}^{[j]}_{n1} & \cdots & \Delta\tilde{X}^{[j]}_{n(T-1)} \end{array}\right]_{n(T-1) \times T(T-1)/2}$$

Moving X from Tensor to Matrix

Now replace each $\Delta\tilde{X}_{it}^{[j]}$ with row vector

$$\Delta\tilde{X}_{it} = |\Delta\tilde{X}_{it}^{[1]}\Delta\tilde{X}_{it}^{[2]}\ldots\Delta\tilde{X}_{it}^{[p]}|$$

$$\rightarrow X = \left[\begin{array}{c|cc|c} \Delta\tilde{X}_{11} & & & \\ \vdots & 0 & 0 & 0 \\ \Delta\tilde{X}_{n1} & & & \\ \hline & \Delta\tilde{X}_{11} \quad \Delta\tilde{X}_{12} & & \\ 0 & \vdots & 0 & 0 \\ & \Delta\tilde{X}_{n1} \quad \Delta\tilde{X}_{n2} & & \\ \hline 0 & 0 & \ddots & 0 \\ \hline & & & \Delta\tilde{X}_{11} \quad \cdots \quad \Delta\tilde{X}_{1(T-1)} \\ 0 & 0 & 0 & \vdots \\ & & & \Delta\tilde{X}_{n1} \quad \cdots \quad \Delta\tilde{X}_{n(T-1)} \end{array}\right]_{n(T-1)\times pT(T-1)/2}$$

Group Lasso Laplacian Penalty

Given our first-differenced and stacked response vector Y, and our first-differenced and stacked design matrix X, we seek to minimize

$$(Y - X\beta)^T(Y - X\beta) + \lambda_1 \sum_{j=1}^{p} \left\|\beta_{(j)}\right\|_2 + \lambda_2 \beta^T L \beta,$$

- λ_1 is the tuning parameter for our group lasso penalty
- λ_2 is the tuning parameter for the network penalty
- L is the Laplacian matrix for the weighted graph where the edge weights between each pair of variables are their absolute correlation

Group Lasso Laplacian Penalty

Given our first-differenced and stacked response vector Y, and our first-differenced and stacked design matrix X, we seek to minimize

$$(Y - X\beta)^T(Y - X\beta) + \lambda_1 \sum_{j=1}^{p} \left\|\beta_{(j)}\right\|_2 + \lambda_2 \beta^T L \beta,$$

- λ_1 is the tuning parameter for our group lasso penalty
- λ_2 is the tuning parameter for the network penalty
- L is the Laplacian matrix for the weighted graph where the edge weights between each pair of variables are their absolute correlation

Group Lasso Laplacian Penalty

Given our first-differenced and stacked response vector Y, and our first-differenced and stacked design matrix X, we seek to minimize

$$(Y - X\beta)^T(Y - X\beta) + \lambda_1 \sum_{j=1}^{p} \left\|\beta_{(j)}\right\|_2 + \lambda_2 \beta^T L \beta,$$

- λ_1 is the tuning parameter for our group lasso penalty
- λ_2 is the tuning parameter for the network penalty
- L is the Laplacian matrix for the weighted graph where the edge weights between each pair of variables are their absolute correlation

Group Lasso Laplacian Penalty

Given our first-differenced and stacked response vector Y, and our first-differenced and stacked design matrix X, we seek to minimize

$$(Y - X\beta)^T(Y - X\beta) + \lambda_1 \sum_{j=1}^{p} \left\|\beta_{(j)}\right\|_2 + \lambda_2 \beta^T L \beta,$$

- λ_1 is the tuning parameter for our group lasso penalty
- λ_2 is the tuning parameter for the network penalty
- L is the Laplacian matrix for the weighted graph where the edge weights between each pair of variables are their absolute correlation

Data Augmentation

Instead of directly minimizing

$$(Y - X\beta)^T(Y - X\beta) + \lambda_1 \sum_{j=1}^{p} \left\|\beta_{(j)}\right\|_2 + \lambda_2 \beta^T L \beta,$$

we fit group lasso to the Laplacian augmented data

$$\begin{bmatrix} Y \\ \mathbf{0} \end{bmatrix}, \quad \frac{1}{\sqrt{1+\lambda_2}} \begin{bmatrix} X \\ \sqrt{\lambda_2} S^T \end{bmatrix}$$

where $L = SS^T$. We then rescale $\hat{\beta}$ by $\frac{1}{\sqrt{1+\lambda_2}}$.

Note that this is similar to the elastic net but with S instead of I, and like with the elastic net we can potentially select all p variables even when $p > n$.

Data Augmentation

Instead of directly minimizing

$$(Y - X\beta)^T(Y - X\beta) + \lambda_1 \sum_{j=1}^{p} \left\|\beta_{(j)}\right\|_2 + \lambda_2 \beta^T L \beta,$$

we fit group lasso to the Laplacian augmented data

$$\begin{bmatrix} Y \\ \mathbf{0} \end{bmatrix}, \quad \frac{1}{\sqrt{1+\lambda_2}} \begin{bmatrix} X \\ \sqrt{\lambda_2} S^T \end{bmatrix}$$

where $L = SS^T$. We then rescale $\hat{\beta}$ by $\frac{1}{\sqrt{1+\lambda_2}}$.

Note that this is similar to the elastic net but with S instead of I, and like with the elastic net we can potentially select all p variables even when $p > n$.

Grouping Effect

Theorem
(Grouping Effect) *Given data (Y, X) and fixed scalars λ_1, λ_2, let $\hat{\beta}(\lambda_1, \lambda_2)$ be the PROLONG solution. Suppose $\hat{\beta}_i \hat{\beta}_j > 0$, the group sizes p_i, p_j are the same, and the two vertices i and j are only linked to each other on the network, $d_i = d_j = w(i, j)$. Define*

$$D_{\lambda_1,\lambda_2}(i,j) = \frac{1}{||Y||_1} \left| \hat{\beta}_i(\lambda_1, \lambda_2) - |\hat{\beta}_j(\lambda_1, \lambda_2) \right|.$$

then

$$D_{\lambda_1,\lambda_2}(i,j) \leq \frac{\sqrt{2(1-\rho)}}{2\lambda_2} + \frac{\lambda_1 \sqrt{p_i}}{\lambda_2 ||Y||_1}$$

where $||Y||_1 = \sum_{i=1}^{n} |Y_i|$ and $\rho = X_{\cdot i}^T X_{\cdot j}$ captures the sample correlation.

Performance with Real Data

- Univariate linear mixed effect models do not pick up a single metabolite from our 352 at an FDR of 0.05
- PROLONG selects $\sim$ 30 metabolites, including targets identified by our clinician collaborators and during our EDA

Performance with Real Data

- Univariate linear mixed effect models do not pick up a single metabolite from our 352 at an FDR of 0.05
- PROLONG selects $\sim$ 30 metabolites, including targets identified by our clinician collaborators and during our EDA

Applying PROLONG

- R package ‘prolong‘, available on Github currently, takes in raw time-scale data and
  - First-differences and shapes the data into the block design structure
  - Automatically selects hyper-parameters and fits the model
  - Provides visualizations for the full data and for selected variables

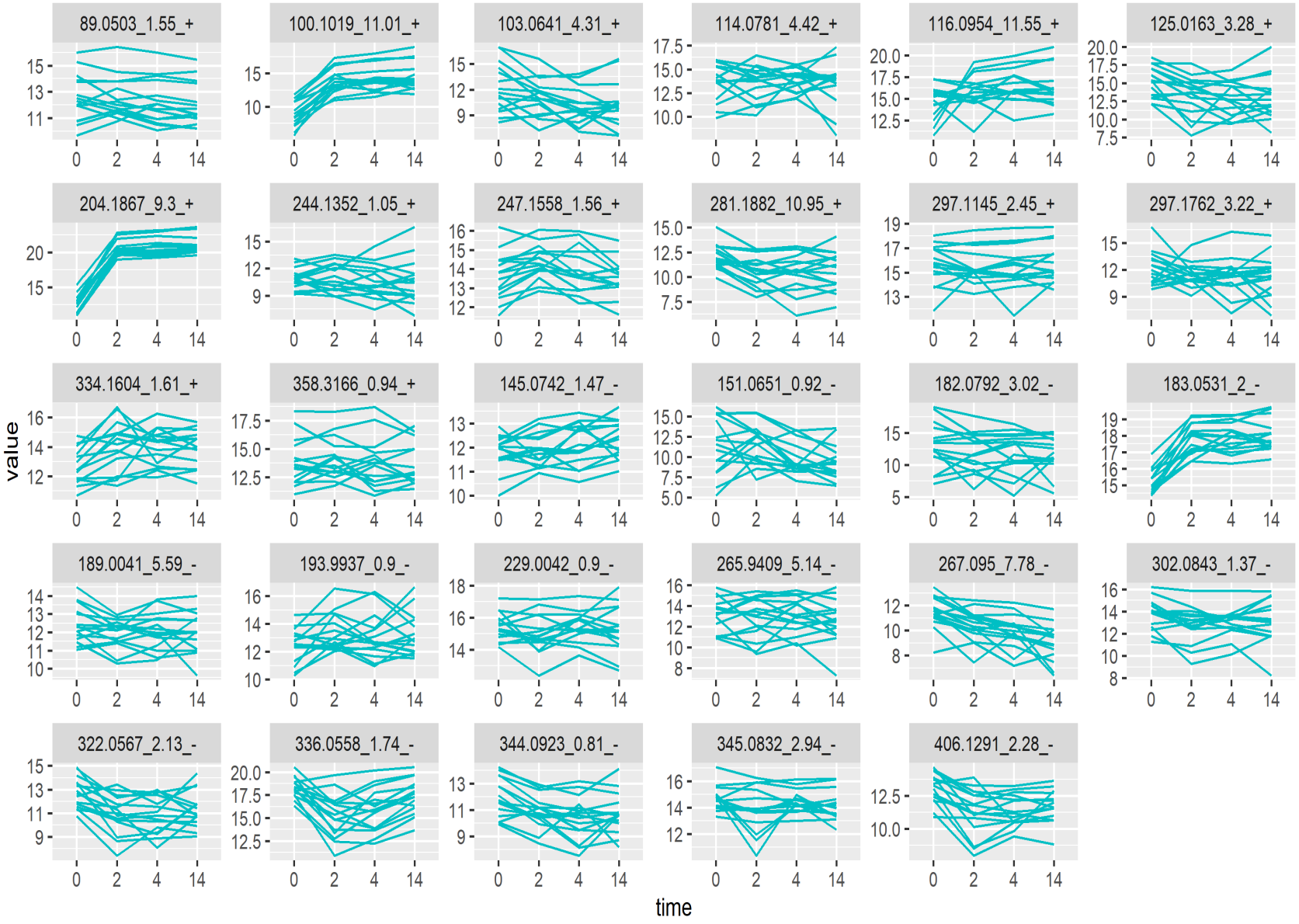
Applying PROLONG

- R package ‘prolong‘, available on Github currently, takes in raw time-scale data and
  - First-differences and shapes the data into the block design structure
  - Automatically selects hyper-parameters and fits the model
  - Provides visualizations for the full data and for selected variables

Applying PROLONG

- R package ‘prolong‘, available on Github currently, takes in raw time-scale data and
  - First-differences and shapes the data into the block design structure
  - Automatically selects hyper-parameters and fits the model
  - Provides visualizations for the full data and for selected variables

R Package Selected Variable Trajectories



Section 3 - Uncertainty Quantification

1. First-differencing longitudinal omics data and incorporating observed covariate dependence via weighted graph Laplacian
2. PROLONG - **P**enalized **r**egression on **o**utcome-guided **l**ongitudinal **o**mics data with **n**etwork and **g**roup constraints
3. Uncertainty quantification in a network-constrained sparse group lasso model for outcome-guided high-dimensional omics data with multiple treatment groups
4. Penalized logistic regression on binary clinical outcomes with potential covariate-related misclassification, with application to longitudinal omics data

General Idea

- Motivated by recent results on the sparse group lasso, we move PROLONG from group lasso + network to sparse group lasso + network penalization.
- We construct a debiased estimator to enable uncertainty quantification and inference.
- Given multiple treatment groups, we provide a framework that allows for joint inference while allowing model hyper-parameters to vary by treatment group.
- If we tune hyper-parameters across all groups, low signal groups like our NTZ data in the following slide can lead to over-sparsifying the other groups.

General Idea

- Motivated by recent results on the sparse group lasso, we move PROLONG from group lasso + network to sparse group lasso + network penalization.
- We construct a debiased estimator to enable uncertainty quantification and inference.
- Given multiple treatment groups, we provide a framework that allows for joint inference while allowing model hyper-parameters to vary by treatment group.
- If we tune hyper-parameters across all groups, low signal groups like our NTZ data in the following slide can lead to over-sparsifying the other groups.

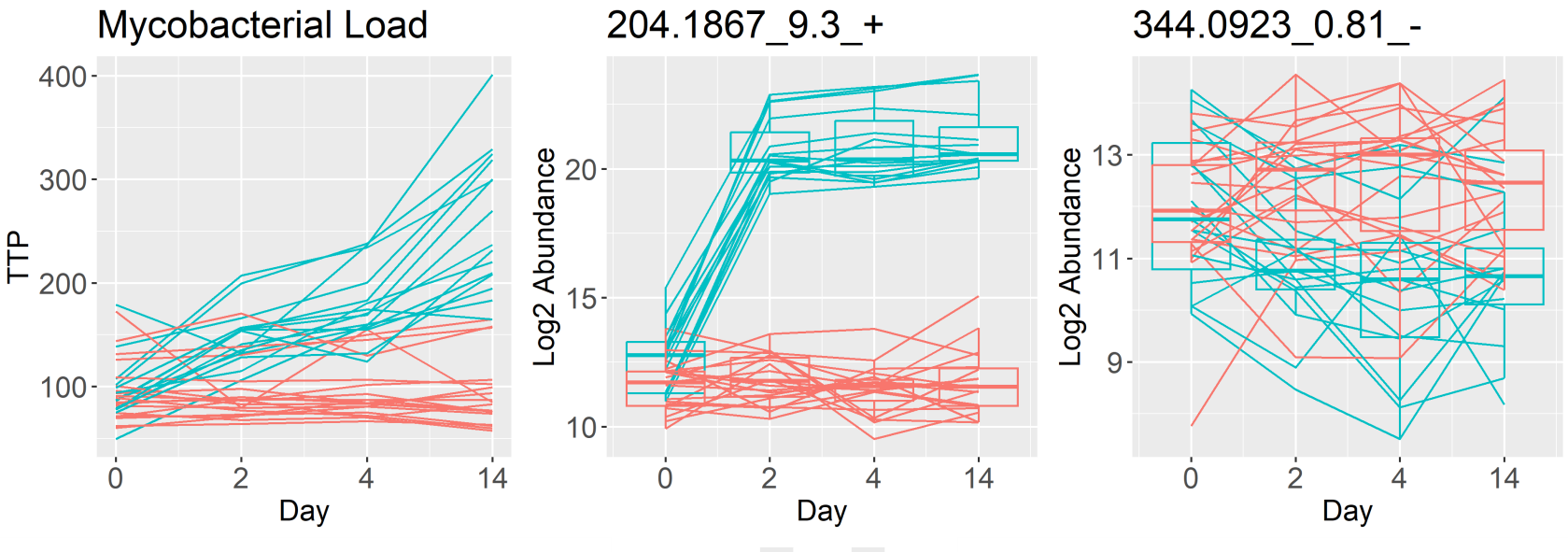
General Idea

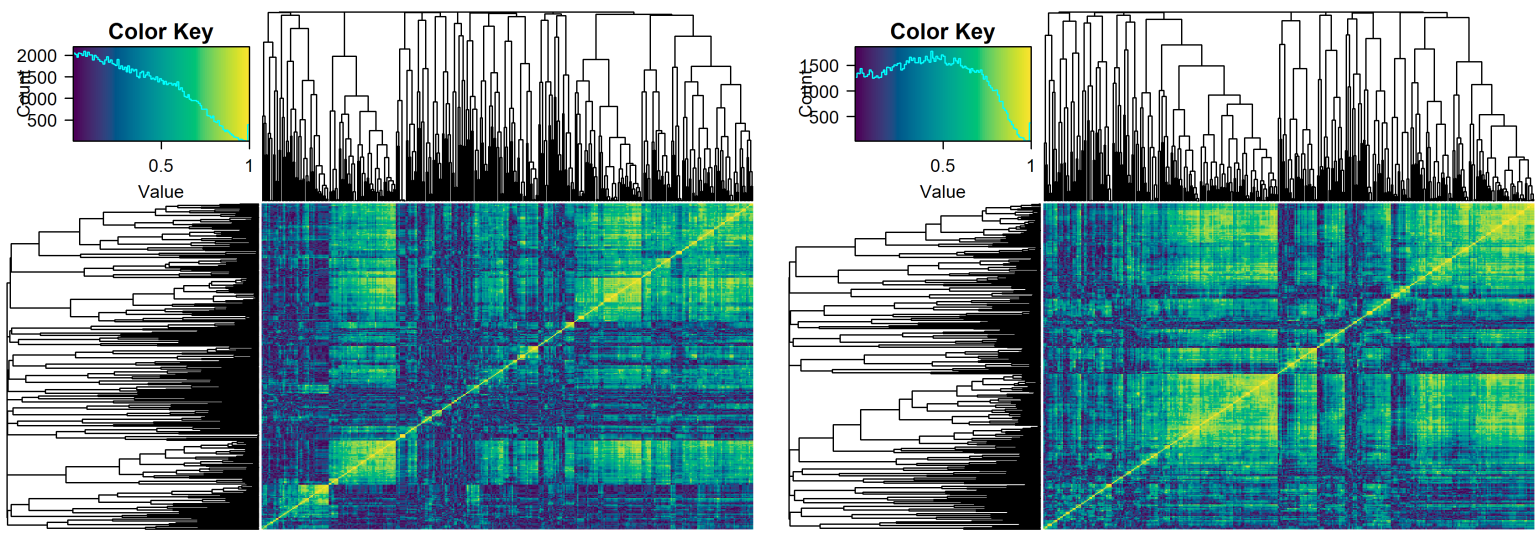
- Motivated by recent results on the sparse group lasso, we move PROLONG from group lasso + network to sparse group lasso + network penalization.
- We construct a debiased estimator to enable uncertainty quantification and inference.
- Given multiple treatment groups, we provide a framework that allows for joint inference while allowing model hyper-parameters to vary by treatment group.
- If we tune hyper-parameters across all groups, low signal groups like our NTZ data in the following slide can lead to over-sparsifying the other groups.

General Idea

- Motivated by recent results on the sparse group lasso, we move PROLONG from group lasso + network to sparse group lasso + network penalization.
- We construct a debiased estimator to enable uncertainty quantification and inference.
- Given multiple treatment groups, we provide a framework that allows for joint inference while allowing model hyper-parameters to vary by treatment group.
- If we tune hyper-parameters across all groups, low signal groups like our NTZ data in the following slide can lead to over-sparsifying the other groups.

Pooled Data





NTZ

RHEZ

Sparse Group Lasso

Our combined sparse group lasso + network penalty is

$$\lambda_1|\beta|_1 + \lambda_1\sqrt{(\tau)}\sum_{k=1}^{K}\sqrt{p_k}\left\|\beta^{(k)}\right\|_2 + \lambda_2\beta^\top L\beta,$$

where we have a weighted sum of lasso and group lasso penalty terms.

The underlying assumption here is that our true β is both sparse not only between groups but also within groups.

Grouping Effect

Theorem
(Grouping Effect). *Given dataset* (Y, X) *and fixed scalars* $(\lambda_1, \lambda_2, \tau)$, *the response* Y *is centered and predictors* X *are standardized. Let* $\hat{\beta}(\lambda_1, \lambda_2, \tau)$ *be the doubly-sparse PROLONG estimator. Suppose that* $\hat{\beta}_i(\lambda_1, \lambda_2, \tau)\,\hat{\beta}_j(\lambda_1, \lambda_2, \tau) > 0$, *the group sizes* p_i *and* p_j *are the same, and the two vertices* i *and* j *are only linked to each other on the network,* $d_i = d_j = w(i, j)$. *Define*

$$D_{\lambda_1,\lambda_2,\tau}(i,j) = \frac{1}{||Y||_1}\left|\hat{\beta}_i(\lambda_1, \lambda_2, \tau) - \hat{\beta}_j(\lambda_1, \lambda_2, \tau)\right|.$$

Then

$$D_{\lambda_1,\lambda_2,\tau}(i,j) \leq \frac{1}{2\lambda_2}\sqrt{2(1-\rho)} + \frac{\lambda_1\sqrt{p_i\tau}}{\lambda_2||Y||_1}.$$

Debiased Lasso

The KKT conditions for lasso estimator $\hat{\beta}^n$, which describe necessary conditions for finding an optimal solution $\hat{\beta}^n$ without violating our constraints, give

$$\frac{X^\top (Y - X\hat{\beta}^n)}{n} = \lambda_1 s$$

where s is a subgradient of the ℓ_1 penalty at $\hat{\beta}_n$.

We add a term proportional to the subgradient to compensate for the downward bias introduced by the ℓ_1 penalty.

$$\hat{\beta}^u = \hat{\beta}^n + \frac{MX^\top (Y - X\hat{\beta}^n)}{n},$$

Here, M is a matrix designed to 'decorrelate' the columns of X, and controls both the bias and variance for our debiased estimator.

Debiased Lasso

The KKT conditions for lasso estimator $\hat{\beta}^n$, which describe necessary conditions for finding an optimal solution $\hat{\beta}^n$ without violating our constraints, give

$$\frac{X^\top(Y - X\hat{\beta}^n)}{n} = \lambda_1 s$$

where s is a subgradient of the ℓ_1 penalty at $\hat{\beta}_n$.

We add a term proportional to the subgradient to compensate for the downward bias introduced by the ℓ_1 penalty.

$$\hat{\beta}^u = \hat{\beta}^n + \frac{MX^\top(Y - X\hat{\beta}^n)}{n},$$

Here, M is a matrix designed to 'decorrelate' the columns of X, and controls both the bias and variance for our debiased estimator.

Debiased Lasso

The KKT conditions for lasso estimator $\hat{\beta}^n$, which describe necessary conditions for finding an optimal solution $\hat{\beta}^n$ without violating our constraints, give

$$\frac{X^\top(Y - X\hat{\beta}^n)}{n} = \lambda_1 s$$

where s is a subgradient of the ℓ_1 penalty at $\hat{\beta}_n$.

We add a term proportional to the subgradient to compensate for the downward bias introduced by the ℓ_1 penalty.

$$\hat{\beta}^u = \hat{\beta}^n + \frac{MX^\top(Y - X\hat{\beta}^n)}{n},$$

Here, M is a matrix designed to 'decorrelate' the columns of X, and controls both the bias and variance for our debiased estimator.

Obtaining M

Algorithm 1 Relaxed Precision Matrix Estimation

Input: Matrix $\mathbf{X}$, scalar μ

Set $\widehat{\Sigma} \equiv \left(\mathbf{X}^\top \mathbf{X}\right)/n$.

for $i = 1, 2, \ldots, p$ **do**

Let m_i be a solution of the convex program:

$$\text{minimize} \quad m^\top \widehat{\Sigma} m \quad \text{subject to} \left\| \widehat{\Sigma} m - e_i \right\|_\infty \leq \mu$$

where $e_i \in \mathbb{R}^p$ is the vector with one at the i-th position and zero everywhere else. ▷ Set $M = \mathrm{I}_{p \times p}$ if not feasible

Purpose of M

- Controlling $|M\hat{\Sigma} - I|_\infty$, the maximum absolute entry-wise difference, controls the bias of $\hat{\beta}^u$
- Minimizing the diagonal elements of $M\hat{\Sigma}M$ controls the variance of $\hat{\beta}^u$

Purpose of M

- Controlling $|M\hat{\Sigma} - I|_\infty$, the maximum absolute entry-wise difference, controls the bias of $\hat{\beta}^u$
- Minimizing the diagonal elements of $M\hat{\Sigma}M$ controls the variance of $\hat{\beta}^u$

Debiased Sparse Group Lasso

We use the same correction as in the debiased lasso,

$$\hat{\beta}^u = \hat{\beta}^n + \frac{MX^\top(Y - X\hat{\beta}^n)}{n},$$

obtaining an estimator $\hat{\beta}^u$ that is approximately Gaussian with covariance $\sigma^2 M\hat{\Sigma}M/n$.

Obtaining $\hat{\sigma}^2$

Consider the optimization problem

$$\underset{\beta}{\text{argmin}} \left\{ ||Y - X\beta||_2^2 + \beta^\top (\lambda_2 L + \lambda_R I)\beta \right\},$$

where $\lambda_R I$ ensures that $\lambda_2 L + \lambda_R I$ is invertible.

Maximizing the respective log-likelihood $l(\sigma^2, \lambda_2, \lambda_r)$ gives

$$\sigma^2 = \frac{1}{n} y^T \Sigma_L^{-1} y.$$

where

$$\Sigma_L = X(\lambda_2 L + \lambda_R I)^{-1} X^\top + I$$

Wald Test

For treatment groups $a = 1, \ldots, A$, we seek to test $|\hat{\beta}_{g,1}^{u\top}, \quad \ldots, \quad \hat{\beta}_{g,A}^{u\top}|^\top = \mathbf{0}$.

We apply a Wald test with $H_0 : \beta_g = \mathbf{0}; H_A : \beta_g \neq \mathbf{0} \quad \forall g$.

Our covariance matrix is block diagonal, with one block for each treatment group a:

$$\left[\begin{array}{c|c|c} \hat{\sigma}_1^2 Q_1 / n_1^* & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \hat{\sigma}_A^2 Q_A / n_A^* \end{array}\right],$$

where $Q_a = M_a \hat{\Sigma}_a M_a$, and M_a is our relaxed inverse of $\hat{\Sigma}_a$.

Debiasing Algorithm

Algorithm 2 PROLONG Debiasing

Input: Vector $y \in \mathbb{R}^{n(T-1)}$, matrix $X \in \mathbb{R}^{n(T-1)\times pT(T-1)/2}$, PROLONG estimator $\hat{\beta}^{n^*}$.

Output: Unbiased estimator $\hat{\beta}^u$, covariance matrix Q.

procedure PROLONG DEBIASING

$$\hat{\Sigma} = \frac{X^\top X}{n}$$

$$M = \text{RELAXED INVERSE}(\hat{\Sigma})$$

$$\hat{\beta}^u = \hat{\beta}^{n^*} + \frac{MX^\top (Y - X\hat{\beta}^{n^*})}{n^*}$$

$$Q = M\Sigma M$$

end procedure=0

Inference

Algorithm 3 PROLONG Inference

Input: Matrices $Y_a \in \mathbb{R}^{n_a \times T}$, tensors $\mathcal{X}_a \in \mathbb{R}^{n_a \times p \times T}$ for $a = 1, \ldots, A$, FDR threshold α.

procedure PROLONG INFERENCE

for $a = 1, \ldots, A$ **do**

$X_a = \text{GetDeltaX}(\mathcal{X}_a)$ $\quad \triangleright\ X_a \in \mathbb{R}^{n_a(T-1) \times pT(T-1)/2}$

$y_a = \text{GetDeltaY}(Y_a)$ $\quad \triangleright\ y_a \in \mathbb{R}^{n_a(T-1)}$

$\hat{\beta}_a^{n^*} = \text{PROLONG}(y_a, X_a)$ $\quad \triangleright\ n_a^* = n_a(T-1)$

$(\hat{\beta}_a^u, Q_a) = \text{PROLONG DEBIASING}(Y_a, X_a, \hat{\beta}_a^{n_a^*})$

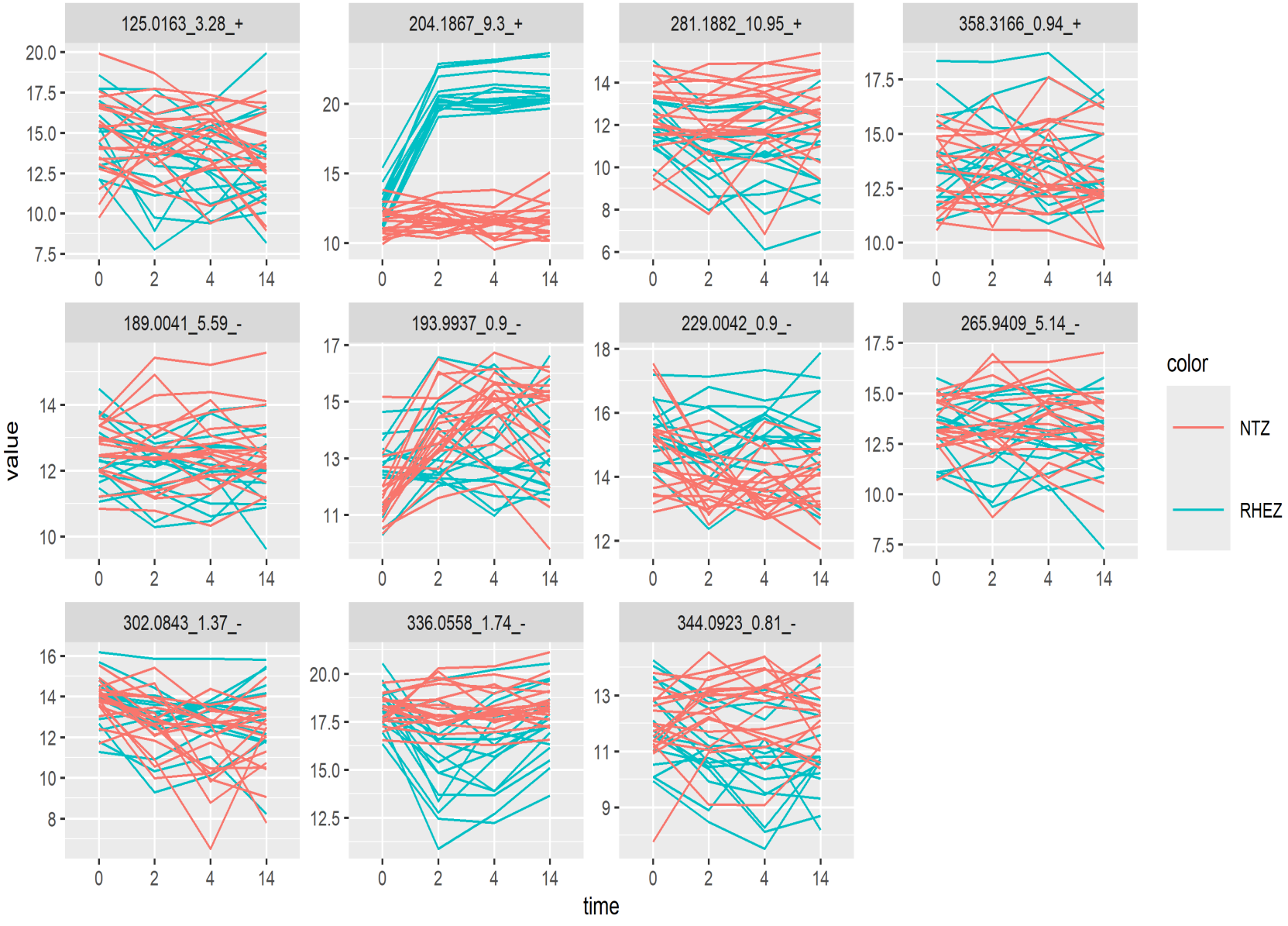
$\hat{\sigma}_a^2 = \hat{\sigma}_{aMLE}^2$

end for

$\forall g$ Wald test with FDR correction at specified α

end procedure

Inference Procedure Selected Variable Trajectories



Software

- The sparse group lasso update, debiasing step, and inference step will be added to R package ‘prolong‘ into a second main function
- All tuning parameters will be selected automatically via MLE or cross-validation with the exception of an optional FDR threshold

Software

- The sparse group lasso update, debiasing step, and inference step will be added to R package ‘prolong‘ into a second main function
- All tuning parameters will be selected automatically via MLE or cross-validation with the exception of an optional FDR threshold

Section 4 - Binary Outcome

1. First-differencing longitudinal omics data and incorporating observed covariate dependence via weighted graph Laplacian
2. PROLONG - **P**enalized **r**egression on **o**utcome-guided **l**ongitudinal **o**mics data with **n**etwork and **g**roup constraints
3. Uncertainty quantification in a network-constrained sparse group lasso model for outcome-guided high-dimensional omics data with multiple treatment groups
4. Penalized logistic regression on binary clinical outcomes with potential covariate-related misclassification, with application to longitudinal omics data

General Idea

- We have some latent binary Y, but can only measure some Y^* that may have some misclassification dependent on variables Z.
- For cross-sectional demographic variables, we want to obtain accurate $\hat{\beta}$'s and uncertainty quantification.
- For large sets of longitudinal omics variables, we want to perform variable selection.
- Use information from the observed correlation and adjust for any covariate-related misclassification.

General Idea

- We have some latent binary Y, but can only measure some Y^* that may have some misclassification dependent on variables Z.
- For cross-sectional demographic variables, we want to obtain accurate $\hat{\beta}$'s and uncertainty quantification.
- For large sets of longitudinal omics variables, we want to perform variable selection.
- Use information from the observed correlation and adjust for any covariate-related misclassification.

General Idea

- We have some latent binary Y, but can only measure some Y^* that may have some misclassification dependent on variables Z.
- For cross-sectional demographic variables, we want to obtain accurate $\hat{\beta}$'s and uncertainty quantification.
- For large sets of longitudinal omics variables, we want to perform variable selection.
- Use information from the observed correlation and adjust for any covariate-related misclassification.

General Idea

- We have some latent binary Y, but can only measure some Y^* that may have some misclassification dependent on variables Z.
- For cross-sectional demographic variables, we want to obtain accurate $\hat{\beta}$'s and uncertainty quantification.
- For large sets of longitudinal omics variables, we want to perform variable selection.
- Use information from the observed correlation and adjust for any covariate-related misclassification.

Model for Latent Y

We are interested in the relationship between some cross-sectional, binary Y and one or more variables X.

We can write our conditional probabilities of Y taking value 1 as

$$P\left(Y_i = 1 \mid X_i; \beta\right) = \pi_{i1} = \frac{\exp\left\{\beta_{1,0} + \beta_{1,X} X_i\right\}}{1 + \exp\left\{\beta_{1,0} + \beta_{1,X} X_i\right\}}$$

We use value 2 as our reference category instead of 0 for ease of indexing.

Model for Latent Y

We are interested in the relationship between some cross-sectional, binary Y and one or more variables X.

We can write our conditional probabilities of Y taking value 1 as

$$P\left(Y_i = 1 \mid X_i; \beta\right) = \pi_{i1} = \frac{\exp\left\{\beta_{1,0} + \beta_{1,X} X_i\right\}}{1 + \exp\left\{\beta_{1,0} + \beta_{1,X} X_i\right\}}$$

We use value 2 as our reference category instead of 0 for ease of indexing.

Model for Latent Y

We are interested in the relationship between some cross-sectional, binary Y and one or more variables X.

We can write our conditional probabilities of Y taking value 1 as

$$P\left(Y_i = 1 \mid X_i; \beta\right) = \pi_{i1} = \frac{\exp\left\{\beta_{1,0} + \beta_{1,X} X_i\right\}}{1 + \exp\left\{\beta_{1,0} + \beta_{1,X} X_i\right\}}$$

We use value 2 as our reference category instead of 0 for ease of indexing.

Structure of X

We focus our application to two general types of X

- Cross-sectional - no transformations applied
- Longitudinal omics - we recommend first-differencing, but do not strictly require doing so

$$\begin{bmatrix} \tilde{X}_{11}^{[j]} & \cdots & \tilde{X}_{1T}^{[j]} \\ & \vdots & \\ \tilde{X}_{n1}^{[j]} & \cdots & \tilde{X}_{nT}^{[j]} \end{bmatrix}_{n \times T} \rightarrow \begin{bmatrix} \Delta\tilde{X}_{11}^{[j]} & \cdots & \Delta\tilde{X}_{1(T-1)}^{[j]} \\ & \vdots & \\ \Delta\tilde{X}_{n1}^{[j]} & \cdots & \Delta\tilde{X}_{n(T-1)}^{[j]} \end{bmatrix}_{n \times (T-1)}$$

Structure of X

We focus our application to two general types of X

- Cross-sectional - no transformations applied
- Longitudinal omics - we recommend first-differencing, but do not strictly require doing so

$$\begin{bmatrix} \tilde{X}_{11}^{[j]} & \cdots & \tilde{X}_{1T}^{[j]} \\ & \vdots & \\ \tilde{X}_{n1}^{[j]} & \cdots & \tilde{X}_{nT}^{[j]} \end{bmatrix}_{n \times T} \rightarrow \begin{bmatrix} \Delta\tilde{X}_{11}^{[j]} & \cdots & \Delta\tilde{X}_{1(T-1)}^{[j]} \\ & \vdots & \\ \Delta\tilde{X}_{n1}^{[j]} & \cdots & \Delta\tilde{X}_{n(T-1)}^{[j]} \end{bmatrix}_{n \times (T-1)}$$

Misclassification Mechanism

Instead of directly observing Y we observe Y^*, which also takes value 1 or 2, that is a potentially misclassified version of Y. We can write Y^*, conditional on Y, as

$$P(Y_i^* = k | Y_i = l, Z_i; \gamma) = \pi_{ikl}^* = \frac{\exp\{\gamma_{kj0} + \gamma_{klZ} Z_i\}}{1 + \exp\{\gamma_{kl0} + \gamma_{klZ} Z_i\}},$$

where Z is a matrix of covariates that may be related to the misclassification.

Complete Data Log-Likelihood

We can estimate (β, γ) via observed log-likelihood

$$\ell_{\text{obs}}(\beta, \gamma; X, Z) = \sum_{i=1}^{n} \sum_{k=1}^{2} y_{ik}^{*} \log \left\{ \sum_{j=1}^{2} \underbrace{\pi_{ikl}^{*}}_{\gamma} \underbrace{\pi_{il}}_{\beta} \right\}.$$

However, jointly maxing β, γ is numerically challenging, so we use the complete log-likelihood with the latent Y, separating the π^{*} and π components

$$\ell_{\text{complete}}(\beta, \gamma; X, Z) = \sum_{i=1}^{n} \left[\sum_{l=1}^{2} y_{il} \log \left\{ \underbrace{\pi_{il}}_{\beta} \right\} + \sum_{l=1}^{2} \sum_{k=1}^{2} y_{il} y_{ik}^{*} \log \left\{ \underbrace{\pi_{ikl}^{*}}_{\gamma} \right\} \right].$$

Network Penalized Log-Likelihood

To incorporate the observed dependence from the covariates in X, we again use penalty $\lambda\beta^T L\beta$. We modify the complete data log-likelihood as follows:

$$\begin{aligned}
&\ell_{\text{penalized}}(\beta, \gamma, \lambda, X, Z) \\
&= \ell_{\text{complete}}(\beta, \gamma; X, Z) - \lambda\beta^\top L\beta \\
&= \sum_{i=1}^{n}\left[\sum_{l=1}^{2} y_{il}\log\left\{\underbrace{\pi_{il}}_{\beta}\right\} + \sum_{l=1}^{2}\sum_{k=1}^{2} y_{il}y^*_{ik}\log\left\{\underbrace{\pi^*_{ikl}}_{\gamma}\right\}\right] - \lambda\beta^\top L\beta
\end{aligned}$$

We use this log-likelihood as the starting point for the penalized EM algorithm.

Penalized EM Algorithm - E-step

In the expectation (E) step, we replace the latent y_{il} with our "best guess" of the probability that $y_{il} = 1$, w_{il}

$$w_{il} = P(Y_i = l | Y_i^*, X, Z) = \sum_{k=1}^{2} \frac{y_{ik}^* \pi_{kl}^* \pi_{il}}{\sum_{\ell=1}^{2} \pi_{ik\ell}^* \pi_{i\ell}}$$

Penalized EM Algorithm - M-step

In the maximization (M) step, we separate our expected log-likelihood into functions of β, γ_{k1}, and γ_{k2}

$$Q = \sum_{i=1}^{n} \left[\sum_{l=1}^{2} w_{il} \log \left\{ \underbrace{\pi_{il}}_{\beta} \right\} + \sum_{l=1}^{2} \sum_{k=1}^{2} w_{il} y_{ik}^{*} \log \left\{ \underbrace{\pi_{ikl}^{*}}_{\gamma} \right\} \right]$$

$$\Rightarrow Q_{\beta}^{L} = \sum_{i=1}^{n} \left[\sum_{l=1}^{2} w_{il} \log \left\{ \underbrace{\pi_{il}}_{\beta} \right\} \right] - \lambda \beta^{\top} L \beta,$$

$$Q_{\gamma_{k1}} = \sum_{i=1}^{n} \left[\sum_{k=1}^{2} w_{i1} y_{ik}^{*} \log \left\{ \underbrace{\pi_{ik1}^{*}}_{\gamma} \right\} \right] \quad \text{(sensitivity component)},$$

$$Q_{\gamma_{k2}} = \sum_{i=1}^{n} \left[\sum_{k=1}^{2} w_{i2} y_{ik}^{*} \log \left\{ \underbrace{\pi_{ik2}^{*}}_{\gamma} \right\} \right] \quad \text{(specificity component)}.$$

Penalized EM Algorithm - M-step

In the maximization (M) step, we separate our expected log-likelihood into functions of β, γ_{k1}, and γ_{k2}

$$Q = \sum_{i=1}^{n} \left[\sum_{l=1}^{2} w_{il} \log \left\{ \underbrace{\pi_{il}}_{\beta} \right\} + \sum_{l=1}^{2} \sum_{k=1}^{2} w_{il} y_{ik}^{*} \log \left\{ \underbrace{\pi_{ikl}^{*}}_{\gamma} \right\} \right]$$

$$\Rightarrow Q_{\beta}^{L} = \sum_{i=1}^{n} \left[\sum_{l=1}^{2} w_{il} \log \left\{ \underbrace{\pi_{il}}_{\beta} \right\} \right] - \lambda \beta^{\top} L \beta,$$

$$Q_{\gamma_{k1}} = \sum_{i=1}^{n} \left[\sum_{k=1}^{2} w_{i1} y_{ik}^{*} \log \left\{ \underbrace{\pi_{ik1}^{*}}_{\gamma} \right\} \right] \quad \text{(sensitivity component)},$$

$$Q_{\gamma_{k2}} = \sum_{i=1}^{n} \left[\sum_{k=1}^{2} w_{i2} y_{ik}^{*} \log \left\{ \underbrace{\pi_{ik2}^{*}}_{\gamma} \right\} \right] \quad \text{(specificity component)}.$$

Penalized EM Algorithm - M-step

In the maximization (M) step, we separate our expected log-likelihood into functions of β, γ_{k1}, and γ_{k2}

$$Q = \sum_{i=1}^{n} \left[\sum_{l=1}^{2} w_{il} \log \left\{ \underbrace{\pi_{il}}_{\beta} \right\} + \sum_{l=1}^{2} \sum_{k=1}^{2} w_{il} y_{ik}^{*} \log \left\{ \underbrace{\pi_{ikl}^{*}}_{\gamma} \right\} \right]$$

$$\Rightarrow Q_{\beta}^{L} = \sum_{i=1}^{n} \left[\sum_{l=1}^{2} w_{il} \log \left\{ \underbrace{\pi_{il}}_{\beta} \right\} \right] - \lambda \beta^{\top} L \beta,$$

$$Q_{\gamma_{k1}} = \sum_{i=1}^{n} \left[\sum_{k=1}^{2} w_{i1} y_{ik}^{*} \log \left\{ \underbrace{\pi_{ik1}^{*}}_{\gamma} \right\} \right] \quad \text{(sensitivity component)},$$

$$Q_{\gamma_{k2}} = \sum_{i=1}^{n} \left[\sum_{k=1}^{2} w_{i2} y_{ik}^{*} \log \left\{ \underbrace{\pi_{ik2}^{*}}_{\gamma} \right\} \right] \quad \text{(specificity component)}.$$

Penalized EM Algorithm - M-step

We fit the Q_γ components via weighted logistic regression where the outcome is Y^*. We can rewrite Q_β^L as

$$\begin{aligned} Q_\beta^L &= \sum_{i=1}^{n} \left[\sum_{l=1}^{2} w_{il} \log \{\pi_{il}\} \right] - \lambda \beta^\top L \beta, \\ &= \sum_{i=1}^{n} \left[\sum_{l=1}^{2} w_{il}^L \log \{\pi_{il}^L\} \right] \quad \text{where} \end{aligned}$$

$$w_{il}^L = [w_{il}; \mathbf{2}]; \quad X^L = (1+\lambda)^{-1/2}[X_i; \lambda^{1/2} S^\top]$$

$$\pi_{il}^L = \frac{\exp\{\beta_{l0} + \beta_{lX} X^L\}}{1 + \exp\{\beta_{j0} + \beta_{lX} X^L\}}$$

$$L = SS^\top$$

We fit Q_β^L via quasi-binomial logistic regression with augmented w_{il}^L as the outcome and X^L as the covariates.

Penalized EM Algorithm - M-step

We fit the Q_γ components via weighted logistic regression where the outcome is Y^*. We can rewrite Q_β^L as

$$
\begin{aligned}
Q_\beta^L &= \sum_{i=1}^{n} \left[\sum_{l=1}^{2} w_{il} \log \{\pi_{il}\} \right] - \lambda \beta^\top L \beta, \\
&= \sum_{i=1}^{n} \left[\sum_{l=1}^{2} w_{il}^L \log \{\pi_{il}^L\} \right] \quad \text{where} \\
w_{il}^L &= [w_{il}; \mathbf{2}]; \quad X^L = (1+\lambda)^{-1/2} [X_i; \lambda^{1/2} S^\top] \\
\pi_{il}^L &= \frac{\exp\{\beta_{l0} + \beta_{lX} X^L\}}{1 + \exp\{\beta_{j0} + \beta_{lX} X^L\}} \\
L &= SS^\top
\end{aligned}
$$

We fit Q_β^L via quasi-binomial logistic regression with augmented w_{il}^L as the outcome and X^L as the covariates.

Penalized EM Algorithm - M-step

We fit the Q_γ components via weighted logistic regression where the outcome is Y^*. We can rewrite Q_β^L as

$$
\begin{aligned}
Q_\beta^L &= \sum_{i=1}^{n} \left[\sum_{l=1}^{2} w_{il} \log \{\pi_{il}\} \right] - \lambda \beta^\top L \beta, \\
&= \sum_{i=1}^{n} \left[\sum_{l=1}^{2} w_{il}^L \log \{\pi_{il}^L\} \right] \quad \text{where} \\
w_{il}^L &= [w_{il}; \mathbf{2}]; \quad X^L = (1+\lambda)^{-1/2} [X_i; \lambda^{1/2} S^\top] \\
\pi_{il}^L &= \frac{\exp\{\beta_{l0} + \beta_{lX} X^L\}}{1 + \exp\{\beta_{j0} + \beta_{lX} X^L\}} \\
L &= SS^\top
\end{aligned}
$$

We fit Q_β^L via quasi-binomial logistic regression with augmented w_{il}^L as the outcome and X^L as the covariates.

Grouping Effect

Theorem
(Grouping Effect) *Given data (Y, X) and fixed scalar λ, let $\hat{\beta}(\lambda)$ be the estimator obtained via the proposed EM algorithm. Suppose $\hat{\beta}_i \hat{\beta}_j > 0$ and the two vertices i and j are only linked to each other on the network, $d_i = d_j = w(i,j)$. Define*

$$D_\lambda(i,j) = \frac{1}{||Y||_1} \left| \hat{\beta}_i(\lambda) - \hat{\beta}_j(\lambda) \right|.$$

then

$$D_\lambda(i,j) \leq \frac{\sqrt{2(1-\rho)}}{2\lambda}.$$

Data Application: MEPS

Data used in original unpenalized method 'COMBO':

- Outcome is self-reported history of MI
- Variables of interest are Age along with Smoking and Exercise Status
- Covariates for Misclassification are Age and Sex

Correlation Table:

	Smoking	Exercise	Age
Smoking	1.0000	0.0284	-0.0593
Exercise	0.0284	1.0000	0.0109
Age	-0.0593	0.0109	1.0000

Sanity Checking with Low Correlation Data

Parameter	Estimates	SE	Estimates	SE
Intercept	-4.3741	0.0655	-4.3874	0.0670
Smoking	1.5437	0.1066	1.5496	0.1039
Exercise	0.3034	0.1257	0.3066	0.1262
Age	0.0939	0.0097	0.0938	0.0091
gamma11	2.9692	0.0997	2.9973	0.0988
gamma21	-1.7656	0.0363	-1.7759	0.0363
gamma31	-0.1984	0.0047	-0.1995	0.0046
gamma12	-3.5796	0.1124	-3.5749	0.1128
gamma22	-0.8183	0.1084	-0.8184	0.1083
gamma32	0.0835	0.0050	0.0833	0.0050

Unpenalized *Laplacian*

Group Lasso Modification

In some applications, we may prioritize variable selection over unbiased dense coefficient estimation, e.g. biomarker discovery with large p omic data. We can supplement our network penalty with a group lasso penalty, replacing Q_β^L with

$$Q_\beta^g = \sum_{i=1}^{n} \left[\sum_{l=1}^{2} w_{il}^L \log \left\{ \pi_{il}^L \right\} \right] - \lambda_g \sum_{k=1}^{K} p_k \beta^{(k)},$$

where p_k is the group size. We maximize Q_β^g using group lasso, selecting λ_g via cross-validation after we select λ and augment w_{il}^L, X^L.

Future Data Work

This project is particularly motivated by very similar data to the first two, but with binary TB Recurrence as our outcome.

- Consider the same data structure, but now our outcome indicates whether the TB comes back in the next few years after treatment
- This indicator may be misclassified, as some patients can be clinically, but not microbiologically, confirmed
- Can we identify biomarkers for later recurrence of TB?

Another motivating data application is finding early metabolomic biomarkers for drug-resistant TB. Drug resistance in TB can lead to delays in adequate treatment and result in worse disease progression - identifying drug-resistance early is critical.

Acknowledgements

My Committee

My Family

My Cohort

Thank You!