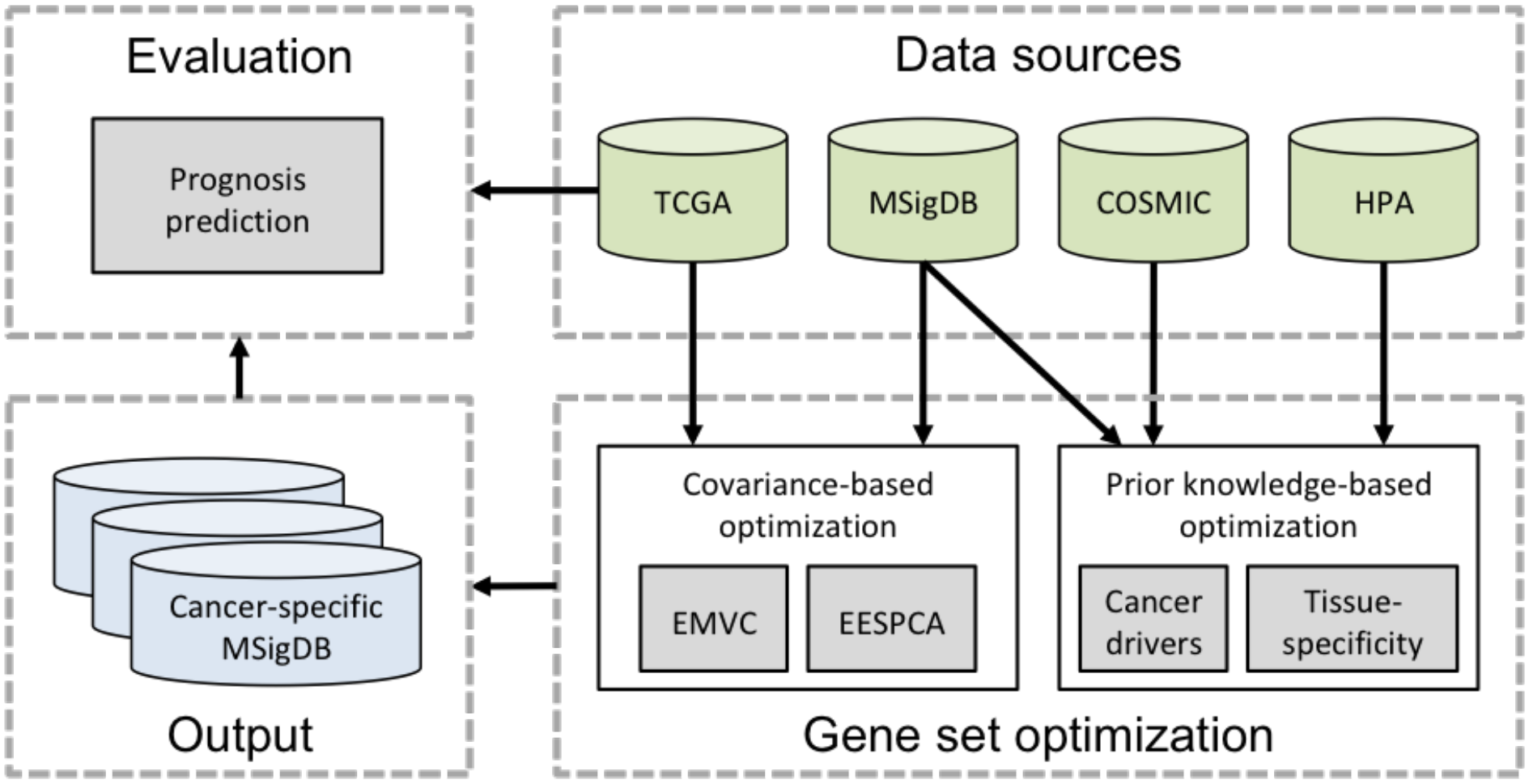
NCI R21: Cancer-specific gene set testing

Rob Frost

Department of Biomedical Data Science
Geisel School of Medicine at Dartmouth

Aim 1 approach

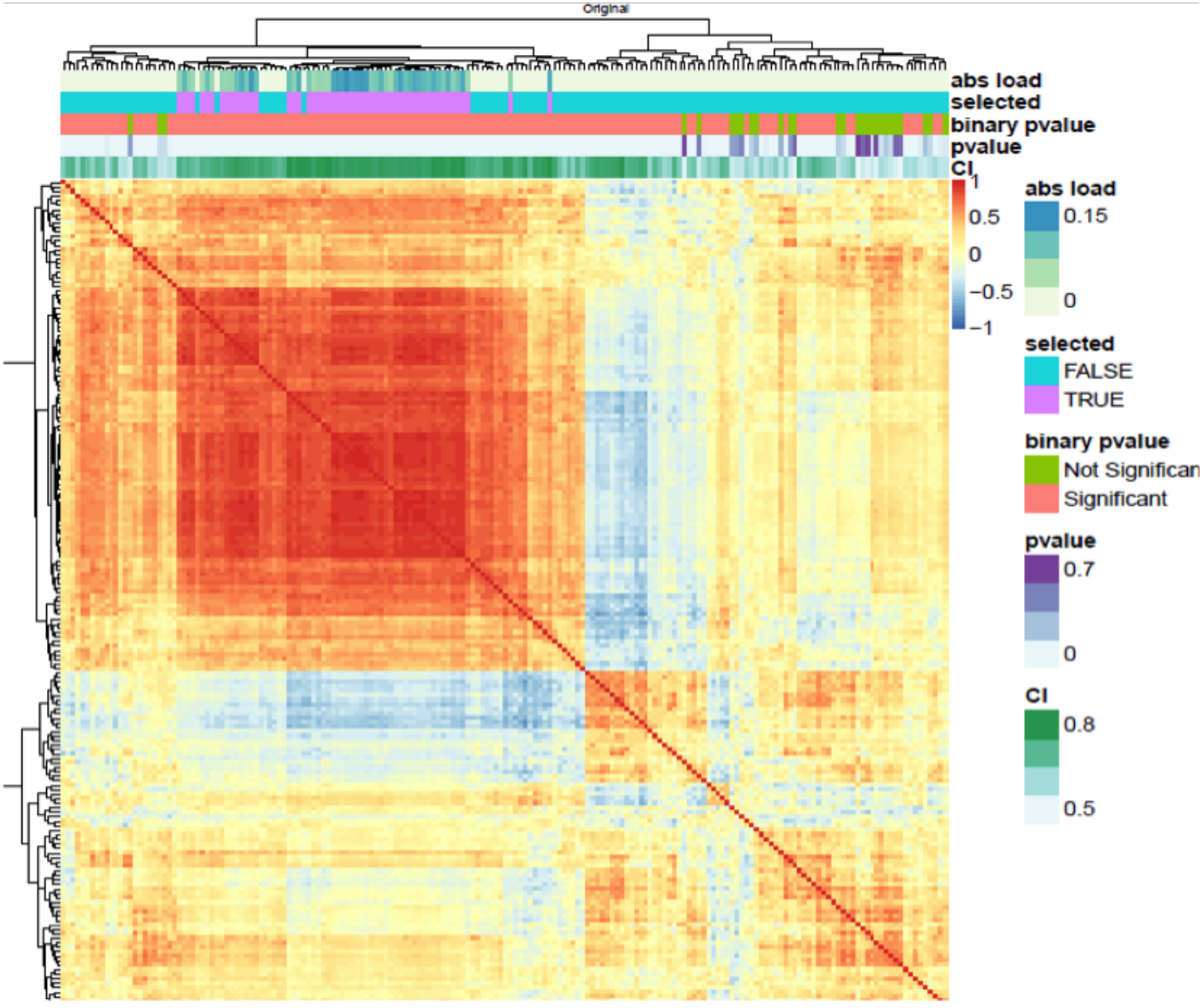


Covariance-based optimization

Hypothesis: Genes whose expression is predictive of cancer survival are more likely to be co-expressed in that cancer than non-predictive genes.

Application to gene set optimization: Estimate the cancer-specific covariance structure for genes in a specific pathway. Retain/weight genes within the primary co-expressed block(s).

GBMLGG covariance for Hallmark G2M Checkpoint



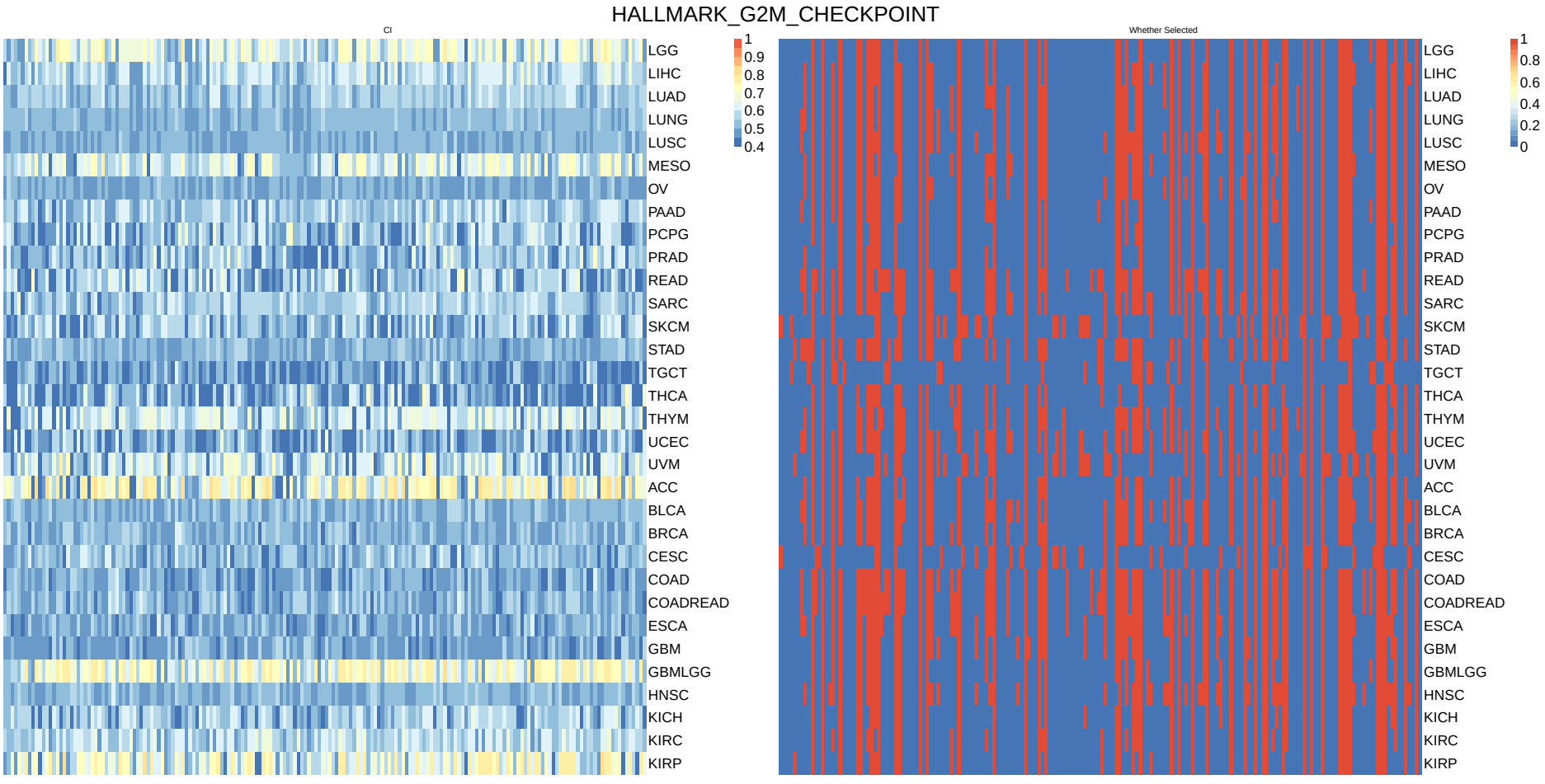
EESPCA

Eigenvectors-from-eigenvalues sparse principal component analysis (EESPCA): Sparse-PCA method based on an approximation of the eigenvalue-eigenvector identity that is specifically optimized for sparse PCA analysis of high-dimensional data (e.g., transcriptomics).

How to apply?

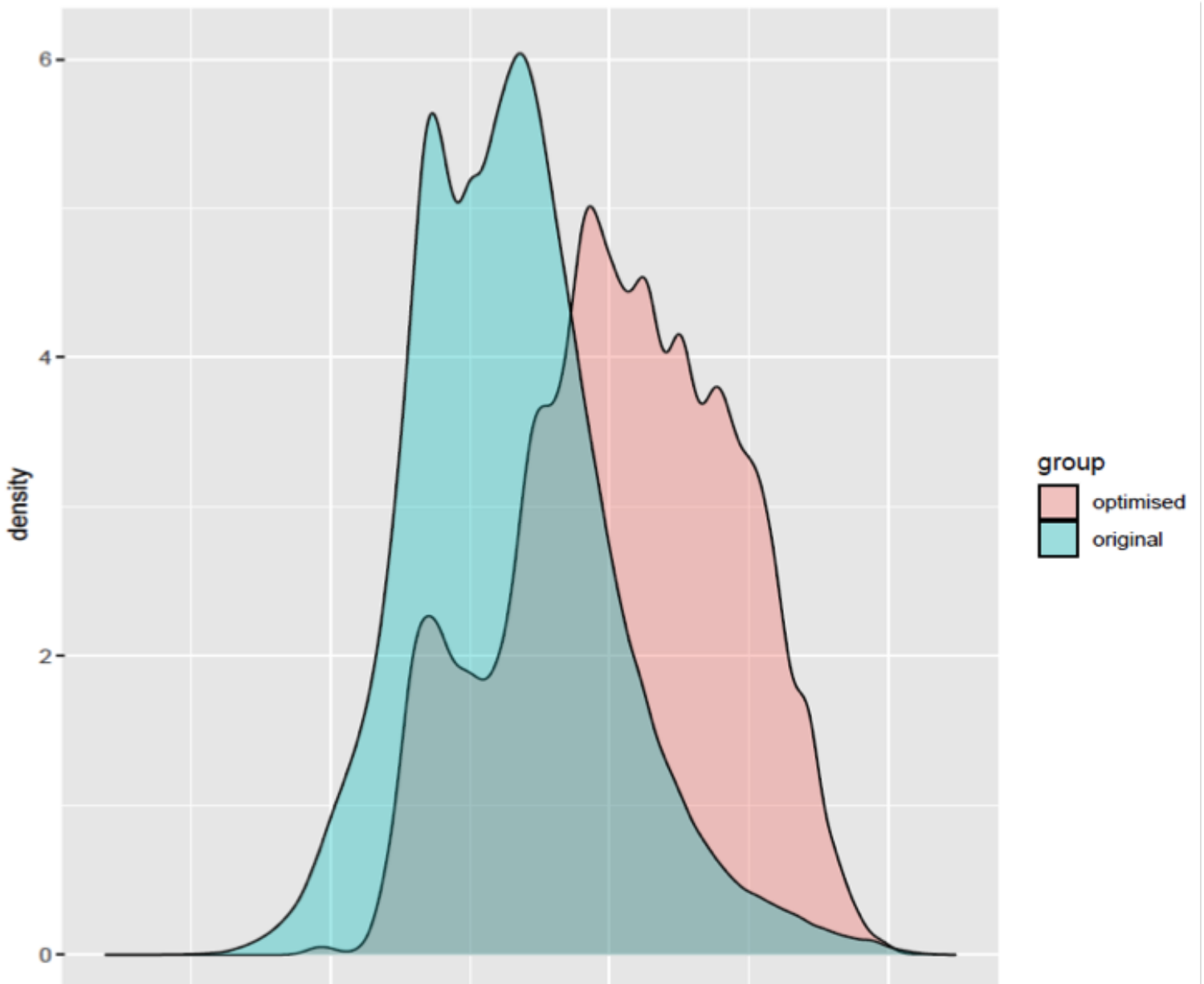
- Compute 1st sparse PCA of cancer gene expression data for genes in set.
- Retain genes with non-zero loading.

EESPCA filtering for Hallmark G2M Checkpoint pathway

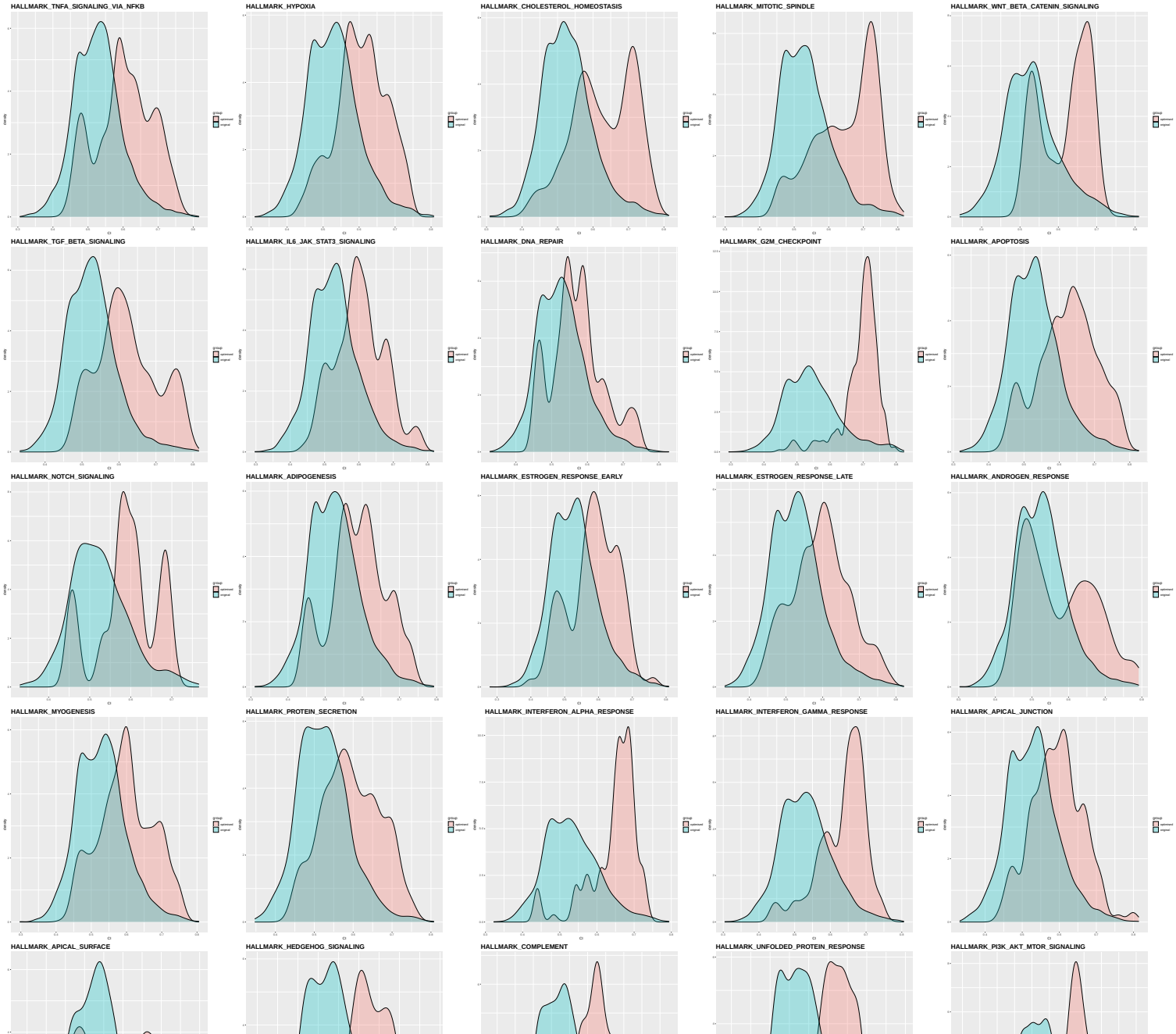


Pan-cancer or cancer-specific?

Impact of EESPCA-based filtering on prognosis prediction



Per-pathway results



Eigenvectors-from-eigenvalues sparse PCA (EESPCA)

Eigenvector-eigenvalue identity

Theorem 1 from Denton et al.

$$|v_{i,j}|^2 = \frac{\prod_{k=1}^{p-1}(\lambda_i(\mathbf{A}) - \lambda_k(\mathbf{M}_j))}{\prod_{k=1,k\neq i}^{p}(\lambda_i(\mathbf{A}) - \lambda_k(\mathbf{A}))} \tag{1}$$

where

- $\mathbf{A}$ be a $p \times p$ Hermitian matrix with eigenvalues $\lambda_i(\mathbf{A})$ and
- $|v_{i,j}|^2$ is the squared value of element j of eigenvector v_i.
- $\mathbf{M}_j$ be the $p-1 \times p-1$ submatrix of $\mathbf{A}$ that results from removing the the jth column and the jth row from $\mathbf{A}$

Applications?

Sample covariance matrix $\hat{\mathbf{\Sigma}}$ is Hermitian, which implies that squared normed PC loadings can be computed as a function of PC variances for the full data set and all of the leave-one-out variable subsets.

How to leverage?

- Null distribution for eigenvector loadings based on eigenvalue RMT distribution?
- Information regarding true eigenvector support?

Challenge: Identity only generates squared normed values (i.e., it fails to capture the sign of the loading) and requires computation of the eigenvalues of all of the covariance sub-matrices.

Eigenvector-eigenvalue identity approximation

$\lambda_i(\hat{\mathbf{\Sigma}})$ for $i = 2, ..., p-1$ are equal to the corresponding sub-matrix eigenvalues $\lambda_i(\hat{\mathbf{\Sigma}}_j)$, many of the eigenvalue difference terms cancel and we can simplify as follows:

$$|v_{1,j}|^2 = \frac{\prod_{k=1}^{p-1}(\lambda_1(\hat{\mathbf{\Sigma}}) - \lambda_k(\hat{\mathbf{\Sigma}}_j))}{\prod_{k=2}^{p}(\lambda_1(\hat{\mathbf{\Sigma}}) - \lambda_k(\hat{\mathbf{\Sigma}}))}$$

$$|v_{1,j}|^2 = \frac{(\lambda_1(\hat{\mathbf{\Sigma}}) - \lambda_1(\hat{\mathbf{\Sigma}}_j))(\lambda_1(\hat{\mathbf{\Sigma}}) - \lambda_2(\hat{\mathbf{\Sigma}}_j))...(\lambda_1(\hat{\mathbf{\Sigma}}) - \lambda_{p-1}(\hat{\mathbf{\Sigma}}_j))}{(\lambda_1(\hat{\mathbf{\Sigma}}) - \lambda_2(\hat{\mathbf{\Sigma}}))...(\lambda_1(\hat{\mathbf{\Sigma}}) - \lambda_{p-1}(\hat{\mathbf{\Sigma}}))(\lambda_1(\hat{\mathbf{\Sigma}}) - \lambda_p(\hat{\mathbf{\Sigma}}))}$$

$$|\tilde{v}_{1,j}|^2 = \frac{\lambda_1(\hat{\mathbf{\Sigma}}) - \lambda_1(\hat{\mathbf{\Sigma}}_j)}{\lambda_1(\hat{\mathbf{\Sigma}}) - \lambda_p(\hat{\mathbf{\Sigma}})}$$

If we also assume that $\lambda_p(\hat{\mathbf{\Sigma}}) \approx 0$, we can further simplify the approximate squared loadings $|\tilde{v}_{1,j}|^2$ to:

$$|\tilde{v}_{1,j}|^2 = 1 - \frac{\lambda_1(\hat{\mathbf{\Sigma}}_j)}{\lambda_1(\hat{\mathbf{\Sigma}})} \tag{2}$$

Properties of approximation

- Greatly reduces the computational cost
- Ratio of approximate-to-true loadings:

$$r_{1,j} = \sqrt{|\tilde{v}_{1,j}|^2 / |v_{1,j}|^2} \tag{3}$$

will tend to be larger for variables that have a non-zero loading on the population PC than for variables with a zero population loading.

Eigenvectors from Eigenvalues Sparse PCA

Input: $n \times p$ mean-centered matrix $\mathbf{X}$ and threshold α (default $1/\sqrt{p}$)
Output: sparse first PC loadings vector $\mathbf{v}_1^s$ and associated eigenvalue λ_1^s

1: $\hat{\mathbf{\Sigma}} = 1/(n-1)\mathbf{X}^T\mathbf{X}$ ▷ Compute sample covariance matrix
2: $\{\mathbf{v}_1, \lambda_1\} = \text{powerIteration}(\hat{\mathbf{\Sigma}})$ ▷ Compute first eigenvector/eigenvalue
3: $\forall_{j \in 1 \ldots p} \lambda_1(\hat{\mathbf{\Sigma}}_j) = \text{powerIteration}(\hat{\mathbf{\Sigma}}_j)$ ▷ Compute first submatrix eigenvalues
4: $\forall_{j \in 1 \ldots p} |\tilde{v}_{1,j}|^2 = 1 - \lambda_1(\hat{\mathbf{\Sigma}}_j)/\lambda_1(\hat{\mathbf{\Sigma}})$ ▷ Approximate first eigenvector
5: $\forall_{j \in 1 \ldots p} r_{1,j} = \sqrt{|\tilde{v}_{1,j}|^2/|v_{1,j}|^2}$ ▷ Compute ratio of eigenvector elements

Eigenvectors from Eigenvalues Sparse PCA

1: Compute sparse $\mathbf{v}_1^s$ using $\mathbf{r}_1$:
   a: $\mathbf{v}_1^s = \mathbf{r}_1 \odot \mathbf{v}_1$ ▷ Scale $\mathbf{v}_1$ by $\mathbf{r}_1$
   b: $\mathbf{v}_1^s = \mathbf{v}_1^s / ||\mathbf{v}_1^s||_2$ ▷ Normalize $\mathbf{v}_1^s$ to unit length
   c: Set $v_{1,j}^s = 0$ for all $v_{1,j}^s < \alpha$ ▷ Truncate $\mathbf{v}_1^s$ using threshold α
   d: $\mathbf{v}_1^s = \mathbf{v}_1^s / ||\mathbf{v}_1^s||_2$ ▷ Normalize $\mathbf{v}_1^s$ to unit length
2: $\lambda_1^s = (\mathbf{v}_1^s)^T \hat{\mathbf{\Sigma}} \mathbf{v}_1^s$ ▷ Compute sparse first eigenvalue

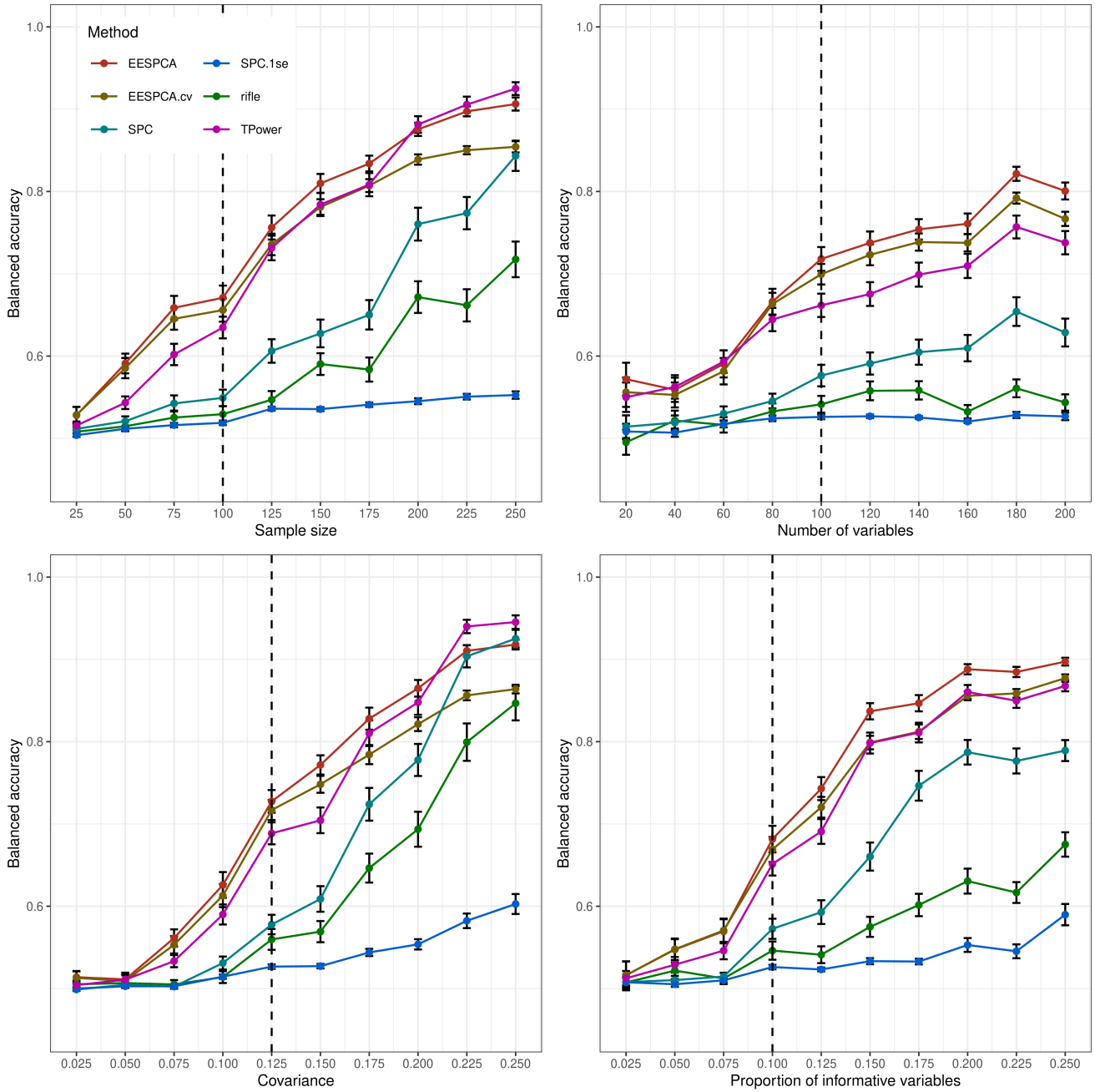
EESPCA performance

Relative to current state-of-the-art sparse PCA methods SPC, TPower and Rifle, EESPCA provides superior:

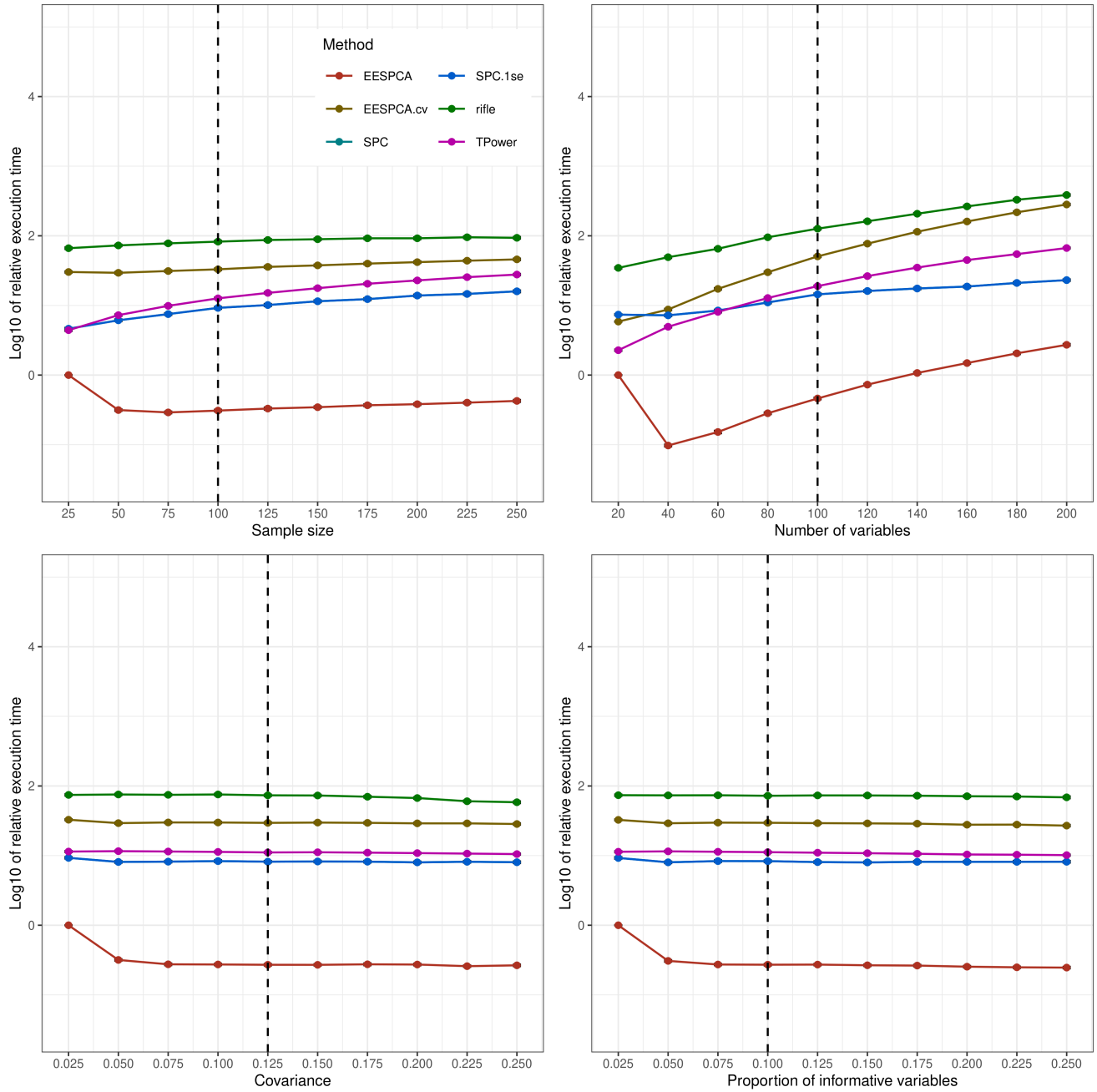
- Classification accuracy
- Out-of-sample reconstruction error
- Computational cost

across a range of covariance structures, data dimensionality and noise levels.

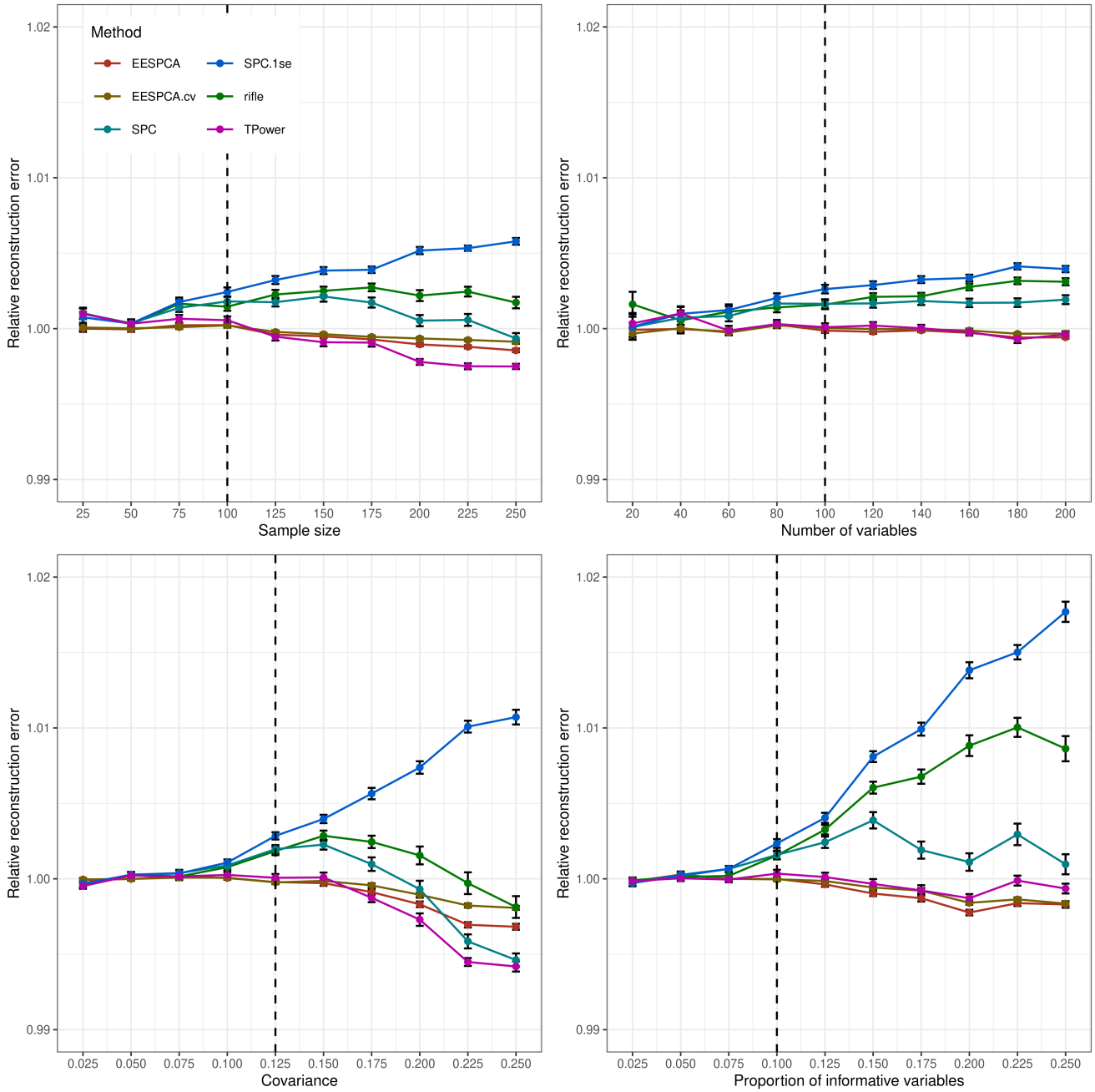
Classification performance



Computational speed



Out-of-sample reconstruction error



Acknowledgments

- NIH grants R21CA253408, P20GM130454, P30CA023108
- Department of Biomedical Data Science at Dartmouth
- Norris Cotton Cancer Center