Cooperative learning for multiview analysis. Ding, Li, Narasimhan, and Tibshirani. 2022, PNAS

Aloise Corbaz, Collection de l'Art Brut, Lausanne

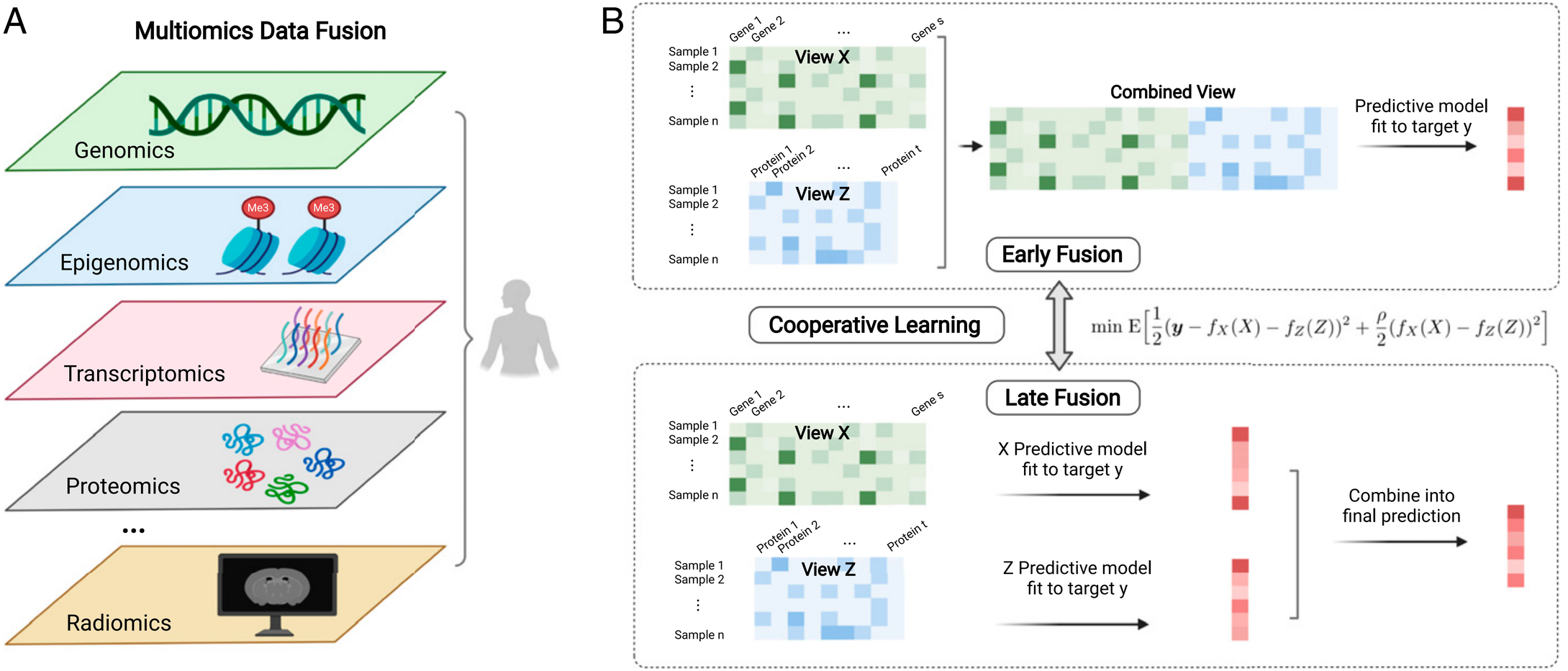
Laura Symul

Oct 10th, 2022

Holmes group meeting

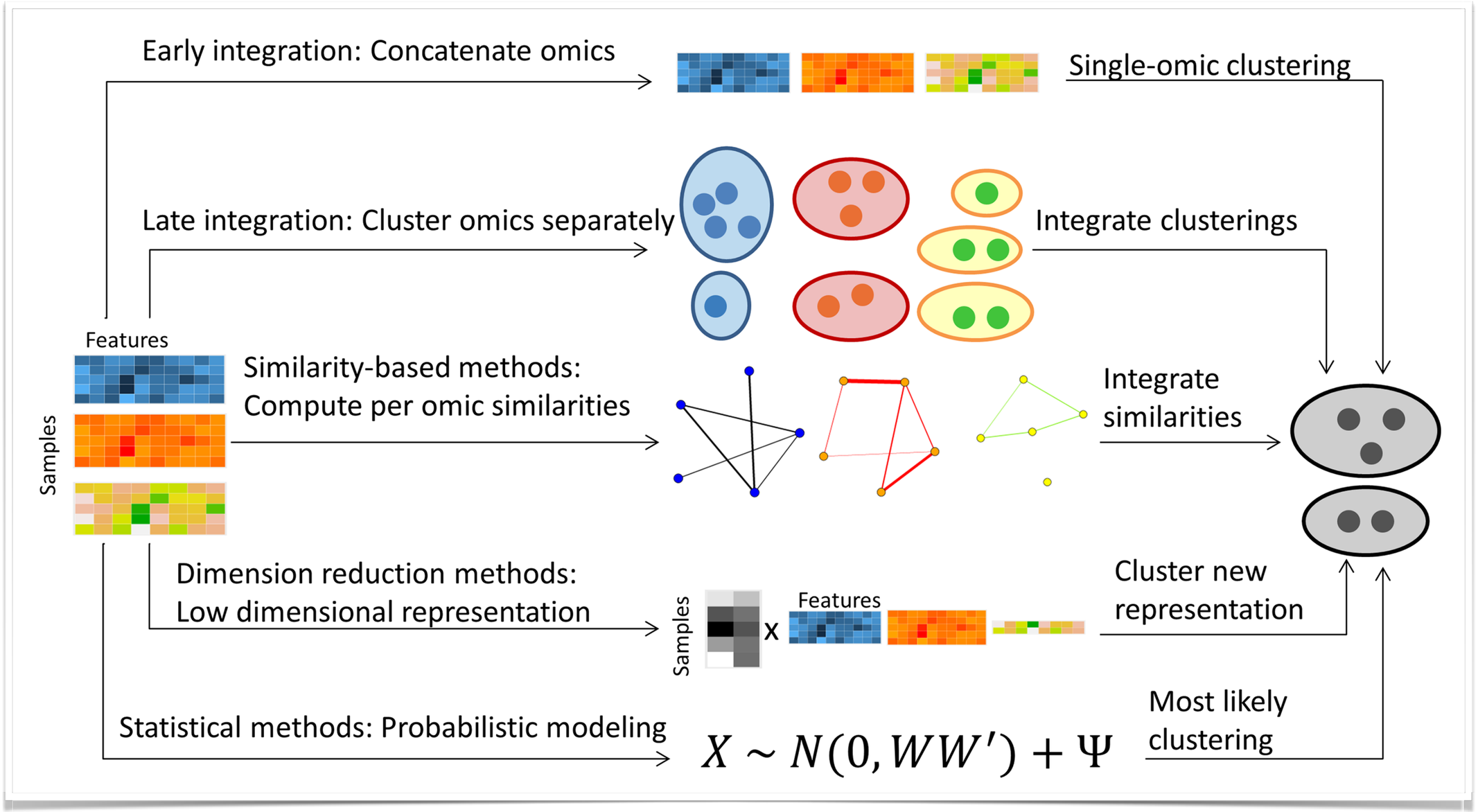
Cooperative learning for multiview analysis

Supervised learning context: the goal is to predict y, a variable of interest



In the main text, they only refer to the two-view case (X and Z), but they generalize in the Appendix

Early *vs.* late integration



Multi-omic and multi-view clustering algorithms: review and cancer benchmark, Rappoport and Shamir, Nucleic Acids Research, 2019

Early *vs.* late integration

In the supervised learning context (for 2 views):

In general early integration

$$\min \mathsf{E}\left[(y - f(X, Z))^2\right]$$

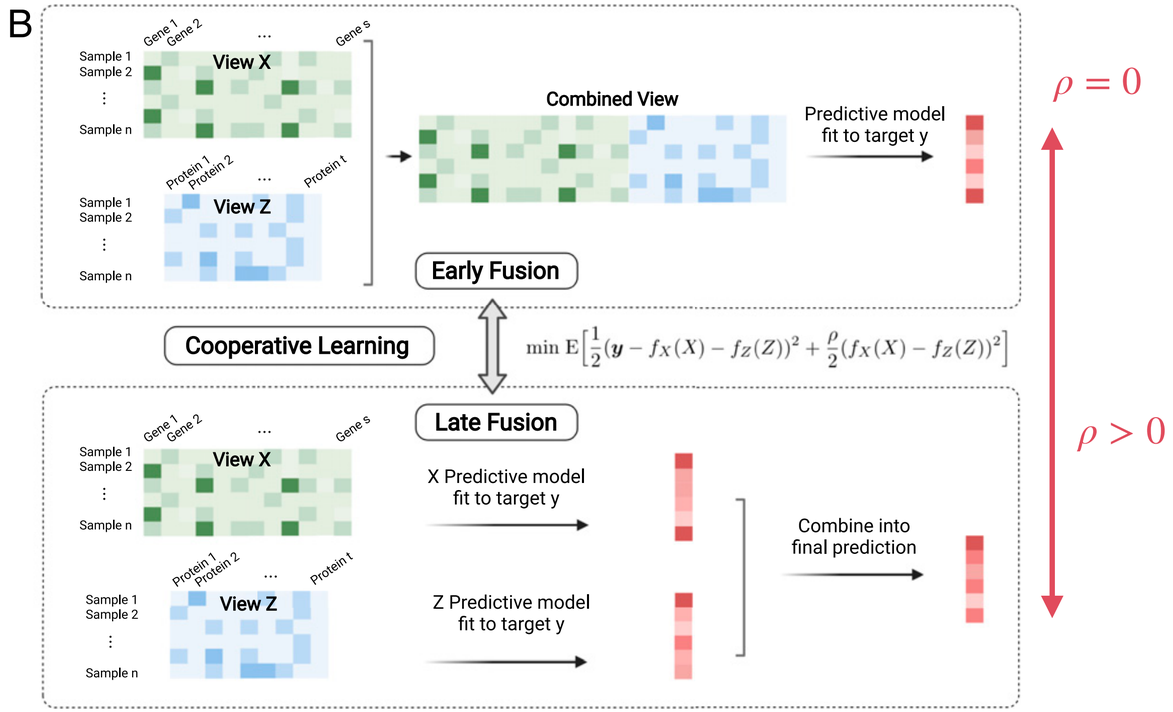
If no interactions between features of each view:

$$\min \mathsf{E}\left[\left(y - f_X(X) - f_Z(Z)\right)^2\right]$$

In general late integration

1. $\min \mathsf{E}\left[(y - f_X(X))^2\right]$
2. $\min \mathsf{E}\left[(y - f_Z(Z))^2\right]$
3. $\min \mathsf{E}\left[\left(y - f\left(f_X(X), f_Z(Z)\right)\right)^2\right]$

Cooperative learning for multiview analysis



Cooperative learning:

$$\min \mathrm{E}\left[\frac{1}{2}(y - f_X(X) - f_Z(Z))^2 + \frac{\rho}{2}(f(X) - f(Z))^2\right]$$

Prediction error

Agreement penalty

Cooperative learning for multiview analysis

$$\min \mathrm{E}\Big[\frac{1}{2}(\boldsymbol{y} - f_X(X) - f_Z(Z))^2 + \frac{\rho}{2}(f_X(X) - f_Z(Z))^2\Big]. \quad \textbf{[1]}$$

The solution to Eq. **1** has fixed points:

$$f_X(X) = \mathrm{E}\Big[\frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)f_Z(Z)}{(1+\rho)}|X\Big],$$

$$f_Z(Z) = \mathrm{E}\Big[\frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)f_X(X)}{(1+\rho)}|Z\Big]. \quad \textbf{[2]}$$

- If $\rho = 0$, from Eq. **1**, we see that cooperative learning chooses a functional form for f_X and f_Z and fits them together. If these functions are additive (for example, linear), then it yields a simple form of early fusion, where we simply use the combined set of features in a supervised learning procedure.
- If $\rho = 1$, then from Eq. **2**, we see that the solutions are the average of the marginal fits for X and Z. This is a simple form of late fusion.

General fitting algorithm:

Update the fit for each data view, in turn, holding the other view fixed, until a minimum is reached.

When updating a function, this approach allows to apply the fitting method for that data view to a penalty-adjusted "partial residual."

Note that this "one-at-a-time" fitting procedure is modular, so that we can choose a fitting mechanism appropriate for each data view. Specifically:

- For quantitative features like gene expression, copy number variation, or methylation: regularized regression (lasso or elastic net), a generalized additive model, boosting, random forests, or neural networks.
- For images: a convolutional neural network (CNN).
- For time-series data: an autoregressive model or a recurrent neural network.

Cooperative Regularized Linear Regression

Regularized Linear Regression (lasso) for multiview data

Consider feature matrices $X \in R^{n \times p_x}$, $Z \in R^{n \times p_z}$, and our target $y \in R^n$.
We assume that the columns of X and Z have been standardized, and y has mean zero.

For a fixed value of the hyperparameter $\rho \geq 0$, we want to find $\boldsymbol{\theta_x} \in \mathcal{R}^{p_x}$ and $\boldsymbol{\theta_z} \in \mathcal{R}^{p_z}$ that minimize:

$$J(\boldsymbol{\theta_x}, \boldsymbol{\theta_z}) = \frac{1}{2}||\boldsymbol{y} - X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z}||^2 + \frac{\rho}{2}||(X\boldsymbol{\theta_x} - Z\boldsymbol{\theta_z})||^2 + \underbrace{\lambda_x P^x(\boldsymbol{\theta_x}) + \lambda_z P^z(\boldsymbol{\theta_z})}_{\text{Lasso penalties (often } l_1 \text{ norm)}}, \quad \textbf{[3]}$$

If $\lambda_x = \lambda_z = \lambda$, the authors call this "**cooperative** regularized linear regression".
If $\lambda_x \neq \lambda_z$, they propose an adaptative strategy, and call this "**adaptative cooperative** regularized linear regression"

If $\rho = 0$ and $\lambda_x = \lambda_z = \lambda$, this is simply applying the lasso on the concatenated views.

Cooperative Regularized Linear Regression

With a common λ, the objective becomes

$$J(\boldsymbol{\theta}_x, \boldsymbol{\theta}_z) = \frac{1}{2}||\boldsymbol{y} - X\boldsymbol{\theta}_x - Z\boldsymbol{\theta}_z||^2 + \frac{\rho}{2}||(X\boldsymbol{\theta}_x - Z\boldsymbol{\theta}_z)||^2 + \lambda(||\boldsymbol{\theta}_x||_1 + ||\boldsymbol{\theta}_z||_1), \quad \textbf{[5]}$$

and we can compute a regularization path of solutions indexed by λ.

Problem [**5**] is convex, and the solution can be computed as follows. Letting

$$\tilde{X} = \begin{pmatrix} X & Z \\ -\sqrt{\rho}X & \sqrt{\rho}Z \end{pmatrix}, \tilde{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix}, \tilde{\boldsymbol{\beta}} = \begin{pmatrix} \boldsymbol{\theta}_x \\ \boldsymbol{\theta}_z \end{pmatrix}, \quad \textbf{[6]}$$

then the equivalent problem to Eq. **5** is

$$\frac{1}{2}||\tilde{\boldsymbol{y}} - \tilde{X}\tilde{\boldsymbol{\beta}}||^2 + \lambda(||\boldsymbol{\theta}_x||_1 + ||\boldsymbol{\theta}_z||_1). \quad \textbf{[7]}$$

This is a form of the lasso and can be computed, for example, by the glmnet package (20). This problem has $2n$ observations and $p_x + p_z$ features.

Algorithm 1 *Direct Algorithm for Cooperative Regularized Regression:*

Input: $X \in \mathcal{R}^{n \times p_x}$ and $Z \in \mathcal{R}^{n \times p_z}$, the response $\boldsymbol{y} \in \mathcal{R}^n$, and a grid of hyperparameter values $(\rho_{\texttt{min}}, \ldots, \rho_{\texttt{max}})$.

for $\rho \leftarrow \rho_{\texttt{min}}, \ldots, \rho_{\texttt{max}}$ **do**

  Set

$$\tilde{X} = \begin{pmatrix} X & Z \\ -\sqrt{\rho}X & \sqrt{\rho}Z \end{pmatrix}, \tilde{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix}.$$

  Solve Lasso$(\tilde{X}, \tilde{\boldsymbol{y}}, \lambda)$ over a decreasing grid of λ values.

end

Select the optimal value of ρ^* based on the CV error and get the final fit.

Note that ρ can be larger than 1

Cooperative Regularized Linear Regression: Relation with late fusion

Problem [**5**] is convex, and the solution can be computed as follows. Letting

$$\tilde{X} = \begin{pmatrix} X & Z \\ -\sqrt{\rho}X & \sqrt{\rho}Z \end{pmatrix}, \tilde{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{y} \\ \mathbf{0} \end{pmatrix}, \tilde{\boldsymbol{\beta}} = \begin{pmatrix} \boldsymbol{\theta_x} \\ \boldsymbol{\theta_z} \end{pmatrix}, \quad \textbf{[6]}$$

then the equivalent problem to Eq. **5** is

$$\frac{1}{2}||\tilde{\boldsymbol{y}} - \tilde{X}\tilde{\boldsymbol{\beta}}||^2 + \lambda(||\boldsymbol{\theta_x}||_1 + ||\boldsymbol{\theta_z}||_1). \quad \textbf{[7]}$$

Next, we discuss the relation of cooperative learning to late fusion. Let X and Z have centered columns and y centered; from Eq. **6**, we obtain

$$\tilde{X}^T\tilde{X} = \begin{pmatrix} X^TX(\mathbf{1}+\rho) & X^TZ(\mathbf{1}-\rho) \\ Z^TX(\mathbf{1}-\rho) & Z^TZ(\mathbf{1}+\rho) \end{pmatrix}. \quad \textbf{[11]}$$

Assuming X and Z have full rank, and omitting the ℓ_1 penalties, we obtain the least-squares estimates

$$\begin{pmatrix} \hat{\boldsymbol{\theta}}_{\boldsymbol{x}} \\ \hat{\boldsymbol{\theta}}_{\boldsymbol{z}} \end{pmatrix} = \begin{pmatrix} X^TX(1+\rho) & X^TZ(1-\rho) \\ Z^TX(1-\rho) & Z^TZ(1+\rho) \end{pmatrix}^{-1} \begin{pmatrix} X^T\boldsymbol{y} \\ Z^T\boldsymbol{y} \end{pmatrix}. \quad \textbf{[12]}$$

If $X^TZ = 0$ (uncorrelated features between the views), this reduces to a linear combination of the least squares estimates for each block; when $\rho = 1$, it is simply the average of the least squares estimates for each block. The above relation also holds when we include the ℓ_1 penalties.

Cooperative Regularized Linear Regression: Simulation experiment

Comparison of cooperative learning accuracy vs several alternatives (prediction from X or Z only, early, and late fusion) in 3 cases:

1. **Correlated X and Z**
2. **Correlated X and Z; X contains more signal than Z** ⟶ This means $t_x > t_z$
3. **X and Z uncorrelated; only X contains signal** ⟶ This means $t_z = 0$

Data generation

We generated Gaussian data with n = 200 and p = 500 in each of two views X and Z and created correlation between them using latent factors.

The response y was generated as a linear combination of the latent factors, corrupted by Gaussian noise.

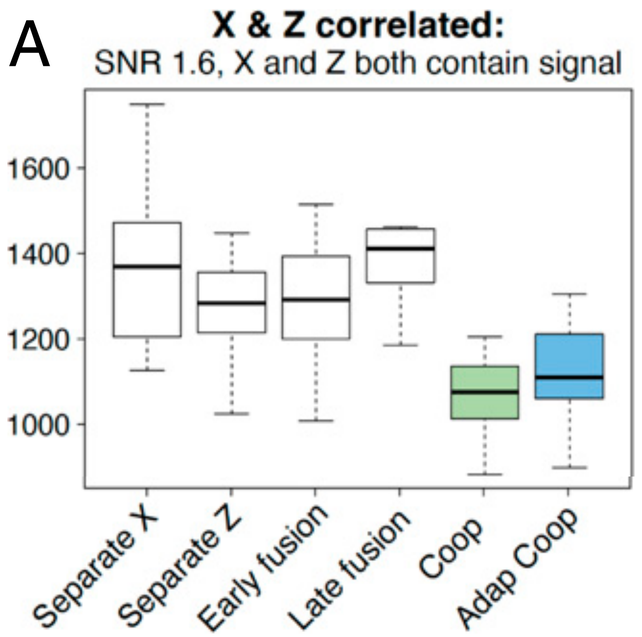
We introduced sparsity by letting some columns of X and Z have no effect on y.

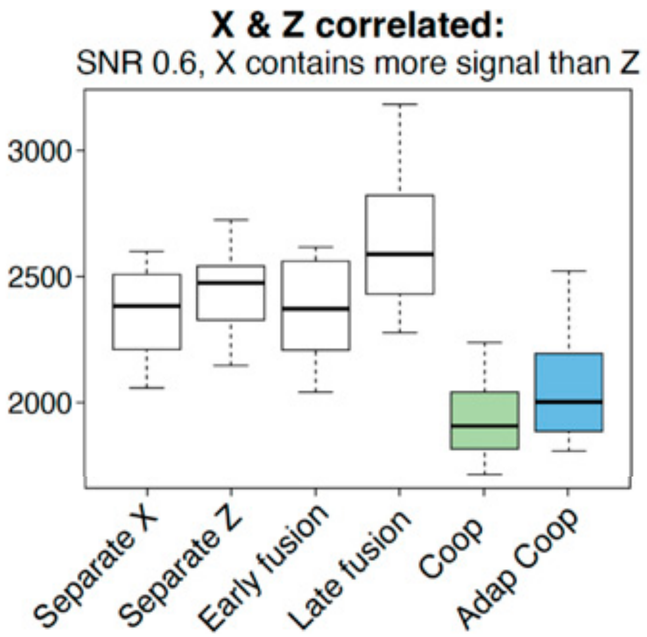
Simulation Procedure for Cooperative Regularized Linear Regression. The simulation is set up as follows. Given values for parameters $n, p_x, p_z, p_u, s_u, t_x, t_z, \boldsymbol{\beta}_u, \sigma$, we generate data according to the following procedure:

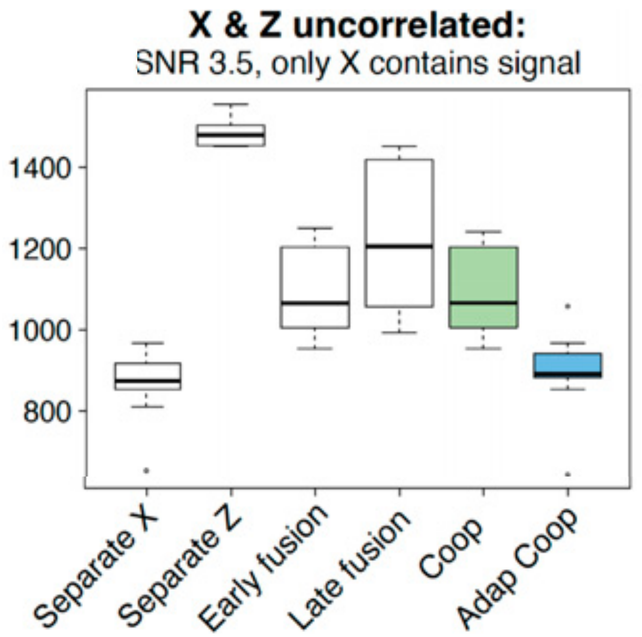
1. $x_j \in \mathcal{R}^n$ distributed i.i.d. MVN$(0, I_n)$ for $j = 1, 2, \ldots, p_x$.
2. $z_j \in \mathcal{R}^n$ distributed i.i.d. MVN$(0, I_n)$ for $j = 1, 2, \ldots, p_z$.
3. For $i = 1, 2, \ldots, p_u$ (p_u corresponds to the number of latent factors, $p_u < p_x$ and $p_u < p_z$):
   a) $u_i \in \mathcal{R}^n$ distributed i.i.d. MVN$(0, s_u^2 I_n)$;
   b) $x_i = x_i + t_x * u_i$;
   c) $z_i = z_i + t_z * u_i$.
4. $X = [x_1, x_2, \ldots, x_{p_x}], Z = [z_1, z_2, \ldots, z_{p_z}]$.
5. $U = [u_1, u_2, \ldots, u_{p_u}]$, $\boldsymbol{y} = U\boldsymbol{\beta}_u + \epsilon$ where $\epsilon \in \mathcal{R}^n$ distributed i.i.d. MVN$(0, \sigma^2 I_n)$.

Cooperative Regularized Linear Regression: Simulation experiment

A

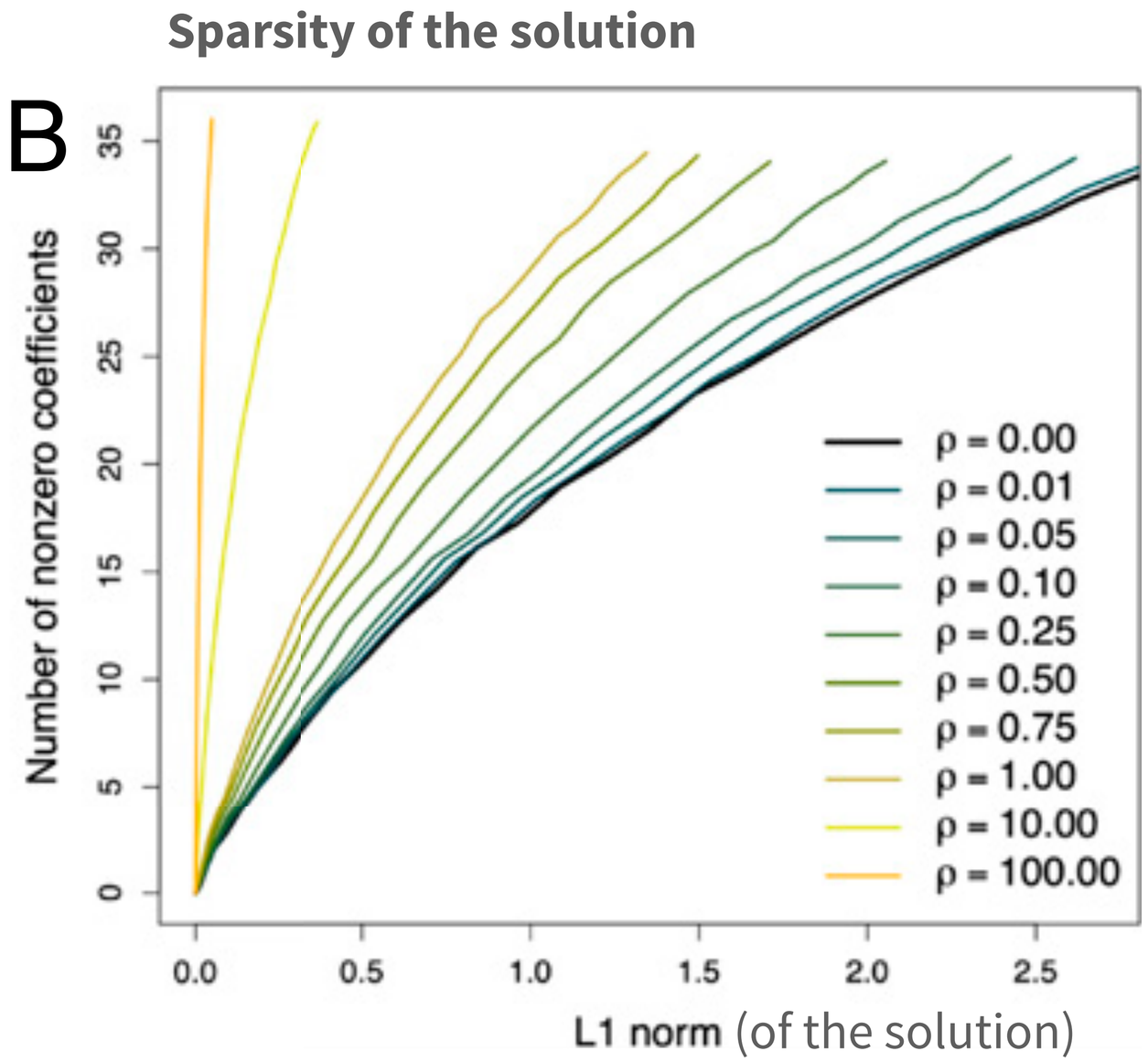






Early fusion is lasso y ~ [X,Z].
Late fusion is
- lasso l_x : y ~ X,
- lasso l_z : y ~ Z,
- then y ~ l_x + l_z

Cooperative Regularized Linear Regression: Simulation experiment



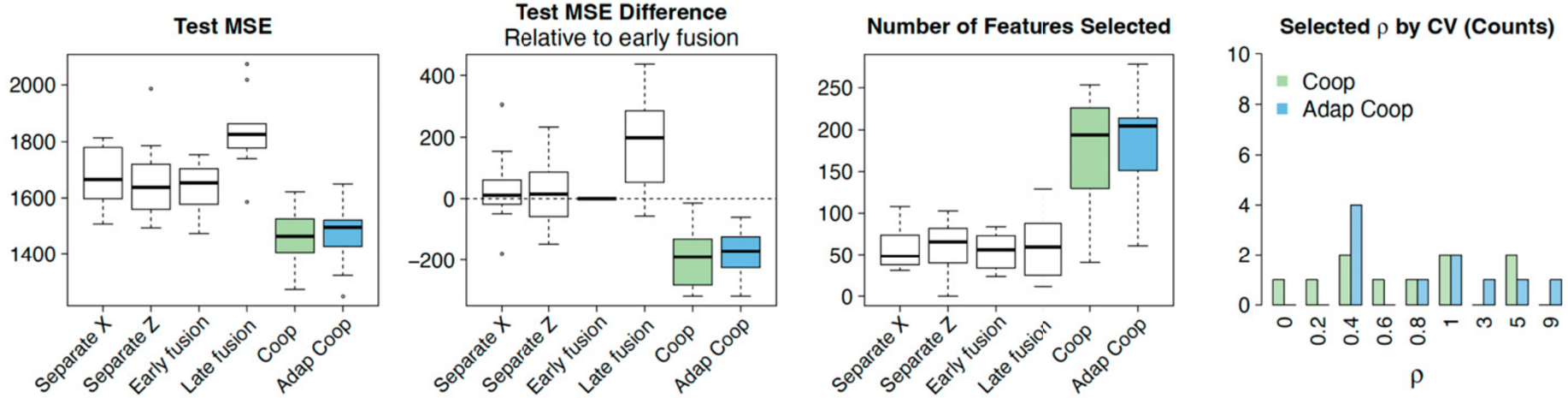
Varies with λ

"Later fusion" discourages sparsity

Cooperative Regularized Linear Regression: Simulation experiment

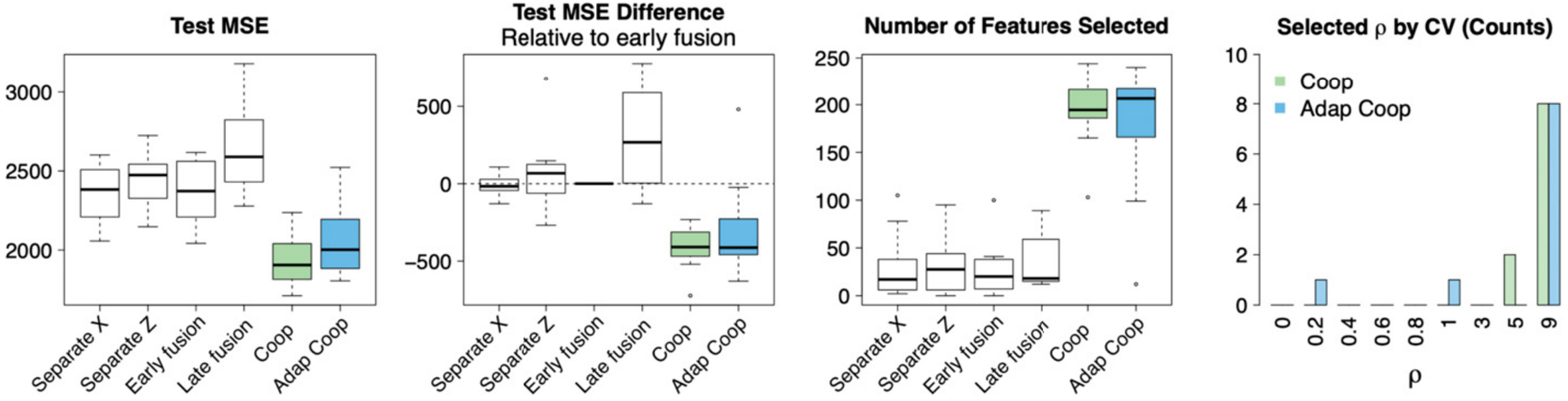
A

X and Z correlated (medium correlation) - Both contain signal - SNR 1.8

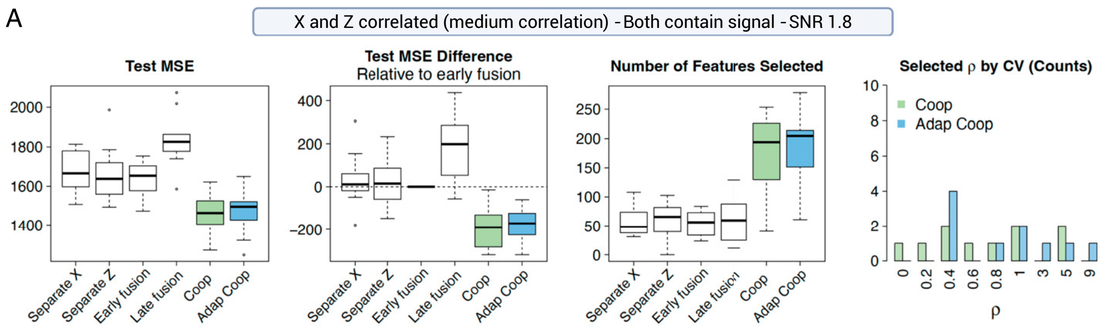


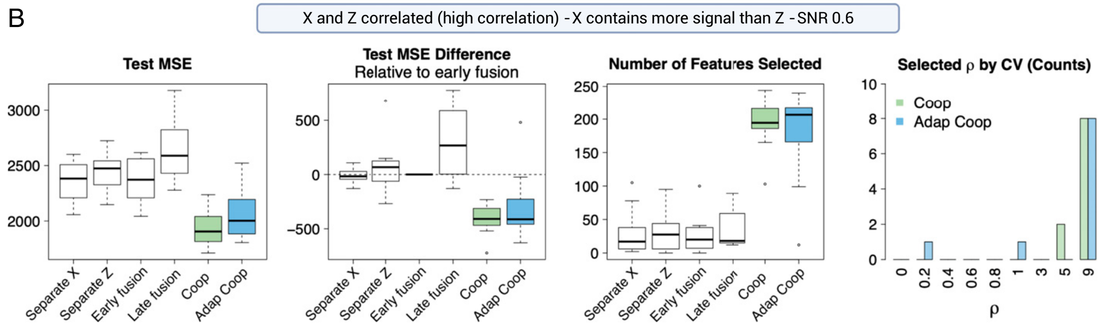
B

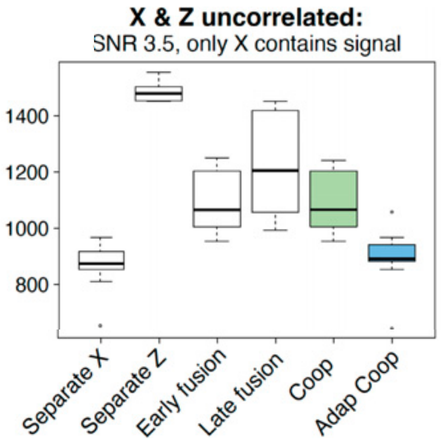
X and Z correlated (high correlation) - X contains more signal than Z - SNR 0.6



Cooperative Regularized Linear Regression: Simulation experiment







- Cooperative learning is **most helpful** when the data **views are correlated and both contain signal.**
- When the correlation between data views is higher, **higher values of ρ are more likely to be selected**.
- **When only one view contains signal** and the views are not correlated, cooperative learning is outperformed by the separate model fit on the view containing the signal.
- But **adaptive cooperative** learning is able to perform on par with the separate model.
- Cooperative learning tends to yield a **less sparse** model

Cooperative Learning: Simulation experiment #2

Imaging + “omics" data

Back to the “general” case (as opposed to the lasso), where the marginal have to be optimized in turns.

Generative model:

Factor model to generate the signal

- in X (the omics data),
- in Z (the “imaging” data), and
- in y (the response)

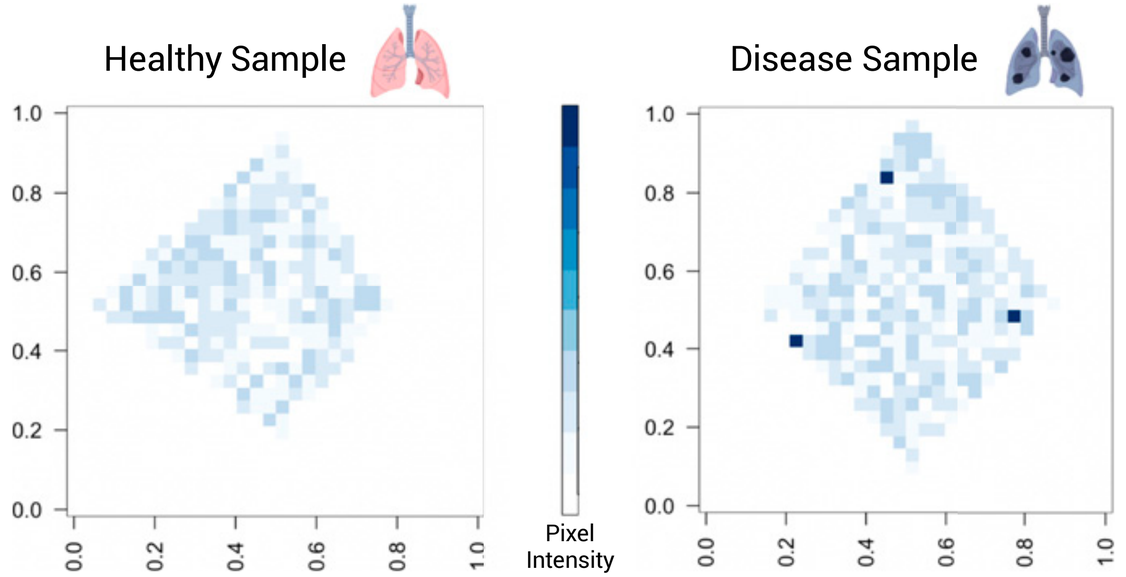


Fig. 4. Generated images for “healthy” and “disease” samples. One can think of the image as an abstract form of a patient’s lung, with the darker spots corresponding to the tumor sites. The intensity of the dark spots on the disease samples is generated to correlate with the omics data and the signal in the outcome.

Cooperative Learning: Simulation experiment #2

Algorithm S3 *Simulation procedure for generating the imaging and "omics" data.*

Input: Parameters $n, p_x, p_u, s_u, t, \sigma, \beta_u, I_{\max}, \text{ndim}, \text{threshold}$.
Output: $X \in \mathcal{R}^{n \times p_x}$ (omics), $Z \in \mathcal{R}^{n \times \text{ndim} \times \text{ndim} \times 1}$ (images assuming one color channel), $\boldsymbol{y} \in \mathcal{R}^n$.

1. $x_j \in \mathcal{R}^n$ distributed i.i.d. MVN$(0, I_n)$ for $j = 1, 2, \ldots, p_x$
2. For $i = 1, 2, \ldots, p_u$ ($p_u < p_x$, where p_u corresponds to the number of factors):
   (a) $u_i \in \mathcal{R}^n$ distributed i.i.d. MVN$(0, s_u^2 I_n)$
   (b) $x_i = x_i + t * u_i$
3. $U = [u_1, u_2, \ldots, u_{p_u}]$, $X = [x_1, x_2, \ldots, x_{p_x}]$
4. $\boldsymbol{y}_u = U\beta_u + \epsilon$ where $\epsilon \in \mathcal{R}^n$ distributed i.i.d. MVN$(0, \sigma^2 I_n)$
5. For $i = 1, 2, \ldots, n$:
   (a) $P_i = \frac{1}{1+\exp(\boldsymbol{y}_{u_i})}$, $\boldsymbol{y}_i \sim$ Bernoulli(P_i)
   (b) Generate a 2D pixel matrix of image $Z_i \in \mathcal{R}^{\text{dim} \times \text{dim} \times 1}$
   (c) Generate a polygon PG_i inside Z_i, defined by 4 vertices $[v_1, v_2, v_3, v_4]$ on the axes, i.e. $v_1 = [0, a_1], v_2 = [0, a_2], v_3 = [a_3, 0], v_4 = [a_4, 0]$, where $a_1 \sim \text{Unif}(\frac{\text{ndim}}{2}, \text{ndim}), a_2 \sim \text{Unif}(-\text{ndim}, -\frac{\text{ndim}}{2}), a_3 \sim \text{Unif}(\frac{\text{ndim}}{2}, \text{ndim}), a_4 \sim \text{Unif}(-\text{ndim}, -\frac{\text{ndim}}{2})$
   (d) Randomly sample points from Z_i: if the point $[x', y']$ falls inside the polygon PG_i, i.e. $[x', y'] \in \text{PG}_i$, then $Z_i[x', y'] \sim \text{Unif}(0, 1)$
   (e) If $\boldsymbol{y}_i = 1$, $I_{\text{disease}} = I_{\max} \times \boldsymbol{y}_{u_i}$, where $I_{\max}$ is the maximum intensity of pixel values for images,
   - For $x' = 1, 2, \ldots, \text{ndim}$:
     - For $y' = 1, 2, \ldots, \text{ndim}$:
       * $P(x', y') \sim \text{Unif}(0, 1)$
       * If $[x', y'] \in \text{PG}_i$ and $P(x', y') < \text{threshold}$, $Z_i[x', y'] = I_{\text{disease}}$

Cooperative Learning: Simulation experiment #2 - Results

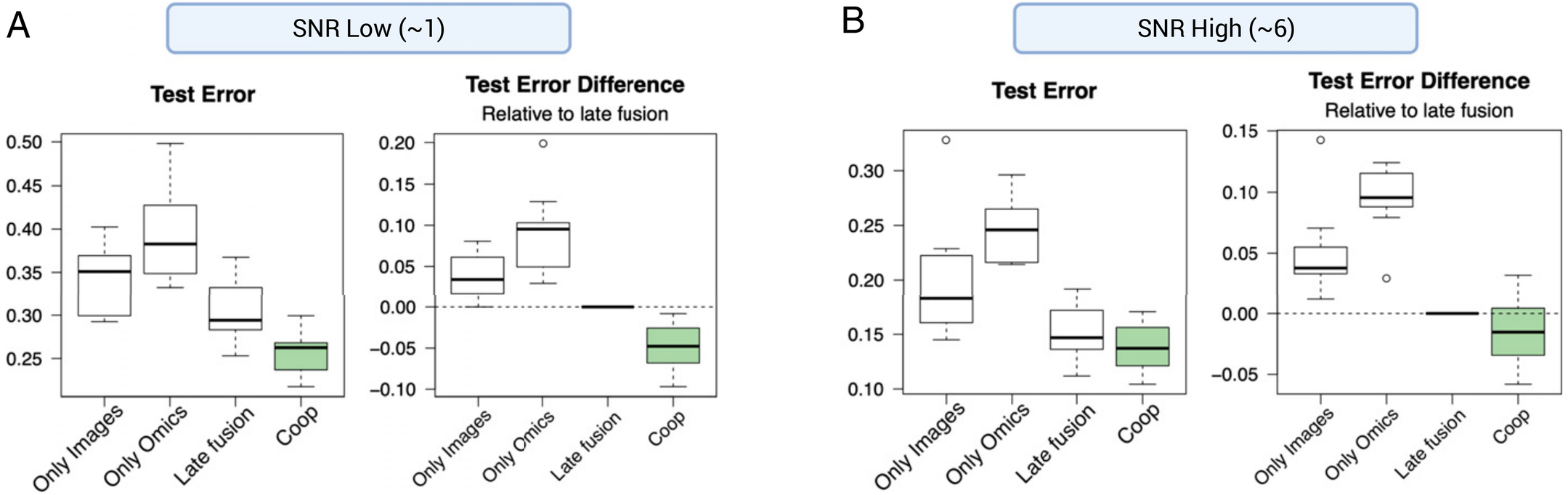


Fig. 5. Simulation studies on cooperative learning with imaging and omics data. *A* corresponds to the relatively low SNR setting (SNR $= 1$) and *B* to the higher SNR setting (SNR $= 6$). For each setting, the left panel shows the misclassification error on the test set for CNN on only images, lasso on only omics, late fusion, and cooperative learning; the right panel shows the difference in misclassification error relative to late fusion. Here, Coop refers to cooperative learning. For both settings, the range of ρ values for cooperative learning to select from is (0, 20). The average ρ selected in the low SNR setting is 6.8 and in the high SNR setting is 8.0.

1) Late fusion achieves a lower misclassification error on the test set than the separate models;

2) Cooperative learning outperforms late fusion and achieves the lowest test error by encouraging the predictions from the two views to agree.

3) cooperative learning is especially helpful when the SNR is low. When the SNR is lower, the marginal benefit of leveraging the other view(s) in strengthening signal becomes larger.

Cooperative Learning on real data: labor onset predictions

Cooperative learning (regression) to a dataset of labor onset, collected from a cohort of women who went into labor spontaneously, as described in I. A. Stelzer *et al.*, 2021.

Proteome and **metabolome** were measured from blood samples collected from the patients during the last 120 days of pregnancy.

The goal of the analysis is to predict time to spontaneous labor.

Training set
40 women

Test set
13 women

1,322 proteins and 3,529 metabolites were quantified.

X: 40 x 1,322
Z: 40 x 3,529

Fits + predictions were repeated 10 times across 10 different splits for the training vs test sets.

Cooperative Learning on real data: labor onset predictions

Table 1. Multiomics studies on labor-onset prediction

Methods	Test MSE		Relative to early fusion		Number of features selected
	Mean	SD	Mean	SD	Mean
Separate proteomics	475.51	80.89	69.14	81.44	26
Separate metabolomics	381.13	36.88	−25.24	30.91	11
Early fusion	406.37	44.77	0	0	15
Late fusion	493.34	63.44	86.97	68.13	21
Cooperative learning	**335.84**	**38.51**	−**70.53**	**32.60**	52

The first two columns in the table show the mean and SD of MSE on the test set across different splits of the training and test sets; the third and fourth columns show the MSE difference relative to early fusion; the last column shows the average number of features selected. The methods include 1) separate proteomics: the standard lasso is applied on the proteomics data only; 2) separate metabolomics: the standard lasso is applied on the metabolomics data only; 3) early fusion: the standard lasso is applied on the concatenated data of proteomics and metabolomics data; 4) late fusion: separate lasso models are first fit on proteomics and metabolomics independently and the predictors are then combined through linear LS; and 5) cooperative learning (Algorithm 1). The average of the selected ρ values is 0.9 for cooperative learning. Cooperative learning achieves the lowest test MSE.

Possibly identified new relevant proteins for predicting labor onset.

Extension: modelling interactions between views

$$\min \mathrm{E}\left[\frac{1}{2}(\boldsymbol{y} - f_X(X) - f_Z(Z))^2 + \frac{\rho}{2}(f_X(X) - f_Z(Z))^2\right]. \quad \textbf{[1]}$$

Prediction error

If $\rho = 0$ (early fusion case), we could also have interactions between the features of the two views.

$$\min \mathrm{E}\left[(y - f(X, Z))^2\right]$$

Where $f(X, Z) \neq f_X(X) + f_Z(Z)$

For example: $f(X, Z) = f_X(X) + f_Z(Z) + f_{XZ}(X, Z)$

They propose an extension to account for these interactions in the case where there are 2 views.

Extension: modelling interactions between views

Modeling Interactions between Views. In our general objective function Eq. **1**, we can capture interactions between features in the same view, by using methods such as random forests or boosting for the learners f_X and f_Z. However, this framework does not allow for interactions between features in different views. Here is an objective function to facilitate such interactions:

$$\min \mathrm{E}\Big[\frac{1}{2}(\boldsymbol{y} - f_X(X) - f_Z(Z) - f_{XZ}(X,Z))^2 + \frac{\rho}{2}(f_X(X) - f_Z(Z))^2 + \frac{\rho}{2(1-\rho)} f_{XZ}^2(X,Z)\Big], \quad \textbf{[18]}$$

where $f_{XZ}(X,Z)$ is a joint function of X and Z, including, for example, interactions between the features in each view.

The solution to Eq. **18** has fixed points:

$$f_X(X) = \mathrm{E}\Big[\frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)f_Z(Z)}{(1+\rho)} - \frac{f_{XZ}(X,Z)}{1+\rho}|X\Big],$$

$$f_Z(Z) = \mathrm{E}\Big[\frac{\boldsymbol{y}}{1+\rho} - \frac{(1-\rho)f_X(X)}{(1+\rho)} - \frac{f_{XZ}(X,Z)}{1+\rho}|Z\Big],$$

$$f_{XZ}(X,Z) = \mathrm{E}\Big[(1-\rho)(\boldsymbol{y} - f_X(X) - f_Z(Z))|X,Z\Big]. \quad \textbf{[19]}$$

When $\rho = 0$, from Eq. **18**, the solution reduces to the additive model $f_X(X) + f_Z(Z) + f_{XZ}(X,Z)$. As $\rho \to 1$, the joint term $f_{XY} \to 0$, and we again get the late fusion estimate as the average of the marginal predictions $\hat{f}_X(X)$ and $\hat{f}_Z(Z)$. To implement this in practice, we simply insert learners such as random forest or boosting for f_X, f_Z and f_{XZ}. The first two use only features from X and Z, while the last uses features from both.

The `multiview` R package

Implements the **cooperative regularized linear regression** algorithm (not the adaptive one, so the $\lambda_X = \lambda_Z = \lambda$)

Perform cooperative learning using the direct algorithm for two or more views.

Description

`multiview` uses `glmnet::glmnet()` to do most of its work and therefore takes many of the same parameters, but an intercept is always included, standardization is always done and several other parameters do not apply. Therefore they are always overridden and warnings issued.

Usage

```
multiview(
  x_list,
  y,
  rho = 0,
  family = gaussian(),
  exclude = NULL,
  mvlambda = NULL,
  ...
)
```

Thank you

Aloise Corbaz, Collection de l’Art Brut, Lausanne