Data Generation

The dataset contains 2,000 simulated images, each with a 16×16 grid of 256 pixels, divided evenly into groups A and B. Images in group A have a β effect in the central 8×8 region. Figure 1 shows an image without the β effect. Figures 2 and 3 display the β effect at strengths of 5 and 4 , respectively. The β matrix values are zero outside the central region. The `group_ind` vector classifies images into groups A (1) and B (0). Noise, ϵ_i, is added to each image, drawn from a multivariate normal distribution with zero mean and an exponential correlation structure:

$$\exp\left(-\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right)$$

where x, y are pixel coordinates. The β effect at 5 is more noticeable than at 4 , which is why a strength of 5 is used for analyses.

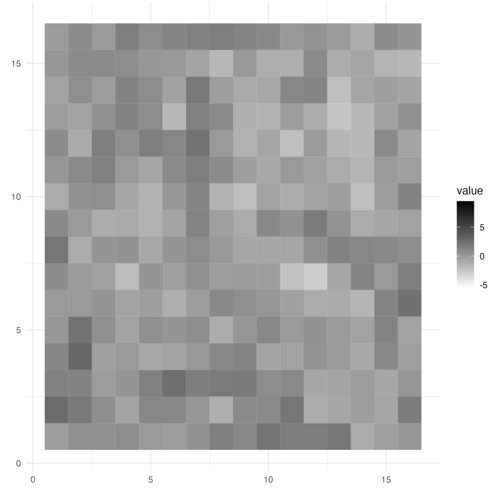

Figure 1: Example image without β effect.

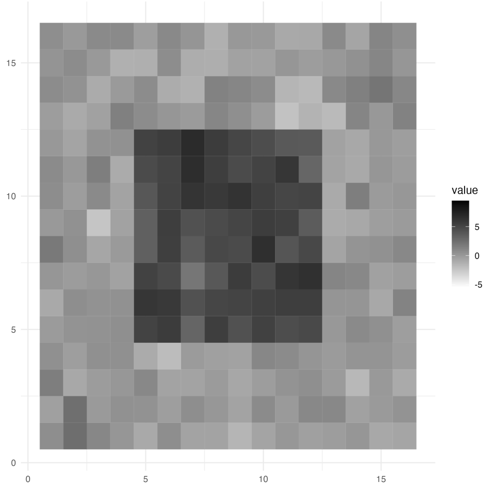

Figure 2: Example image with $\beta = 5$.

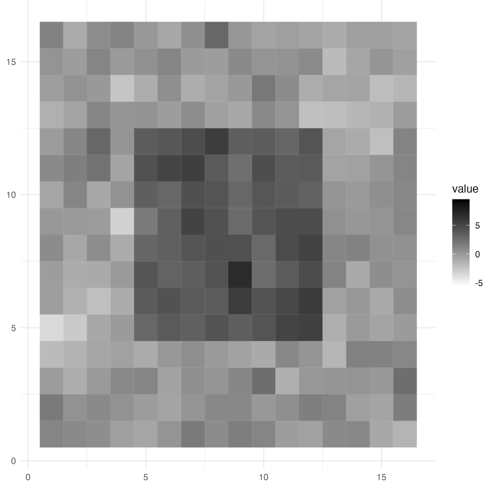

Figure 3: Example image with $\beta = 4$.

VBM

In the VBM analysis, a Generalized Linear Model (GLM) was applied pixel-wise to assess group effects on pixel intensities across 1000 iterations. For each iteration, the model generated effect size estimates and p-values for each pixel. These p-values were then corrected for multiple comparisons using the Bonferroni method. Figure 4 depicts the frequency of significant p-values in across pixels, with pixels showing significant β in all 100 iterations appearing in black, and those never showing significance in white.

In the VBM analysis, a Generalized Linear Model (GLM) was applied pixel-wise to assess group effects on pixel intensities across 1000 iterations. For each iteration, the model generated effect size estimates and p-values for each pixel. These p-values were then corrected for multiple comparisons using the Bonferroni method.

Figure 4 shows the frequency of significant p-values across pixels, with pixels showing significant β in all 1000 iterations appearing in black, and those never showing significance in white. Figure 5 displays the percentage of significant p-values after multiple correction. Figure 6 presents a boxplot of the percentage of significant p-values in the outer (non-central) areas, with the left boxplot showing values before adjustment and the right boxplot showing values after adjustment.

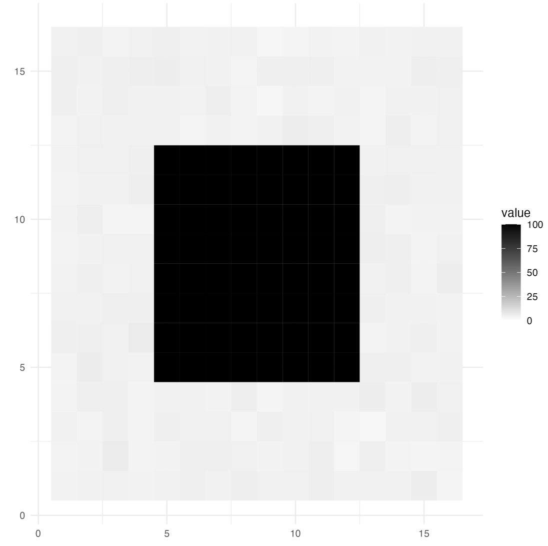

Figure 4: Percentage of significant p-values across pixels in VBM analysis.

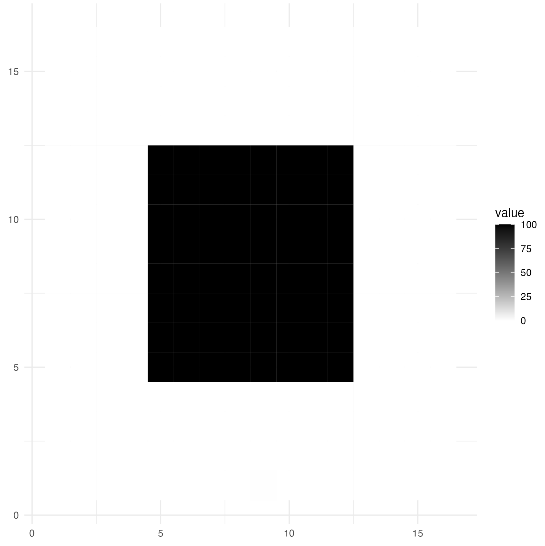

Figure 5: Percentage of significant p-values after correction across pixels in VBM analysis.

LASSO

A LASSO model was employed to predict group assignments using pixel values from images. In each iteration, 80% of the data was used for training and the remaining 20% for

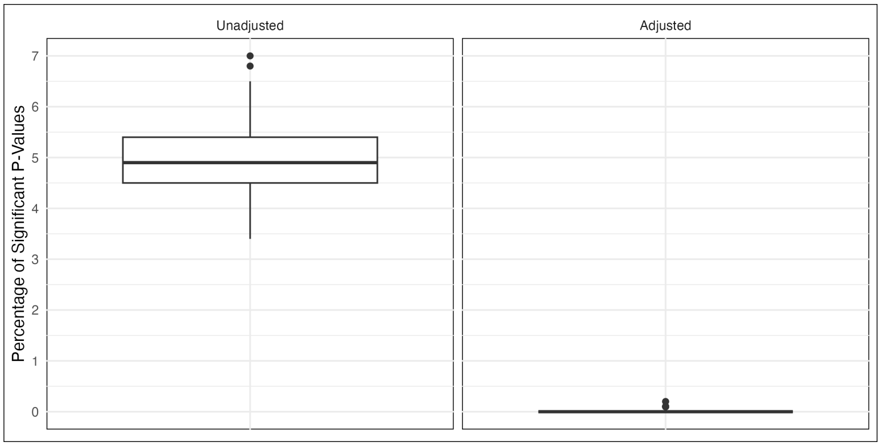

Figure 6: Percentage of significant p-values in outer area before (left) and after (right) adjustment from the VBM analysis.

testing. The optimal λ parameters, `lambda_min` and `lambda_1se`, were determined via cross-validation within the training group. The model's performance was assessed on the test group using accuracy and AUC metrics.

Initially, including all pixel values in the model led to perfect separation, indicating potential overfitting. To address this, the model construction began by incrementally adding one pixel from the image's edge and one from the center, evaluating if these additions achieved perfect accuracy. After integrating two pixels from each area, totaling four pixels, the model achieved perfect separation.

Additionally, a permutation test was conducted to estimate p-values. Using all the pixels, 500 iterations were performed. Within each iteration, the outcome was permuted 100 times. The original coefficients were compared with the permuted coefficient estimates to simulate the p-values of each covariate/pixel.

Figures 7, 8, and 9 illustrate the results of the analysis:

- Figure 7 shows the p-values before multiple adjustment.
- Figure 8 displays the p-values after multiple adjustment.
- Figure 9 presents a boxplot of the percentage of significant p-values for the outer area before and after adjustment.

Frequency

The exponential correlation matrix with rate 1 was used to calculate the eigenvectors and eigenvalues. The matrix appeared to be positive-definite, so all eigenvalues were positive. These eigenvectors were then used to transform the pixel values. A Lasso regression model was fitted on the transformed data to predict `group_ind`. To assess the significance of the model coefficients, 100 permutation tests were conducted within each iteration to obtain p-values. This process was replicated 100 times.

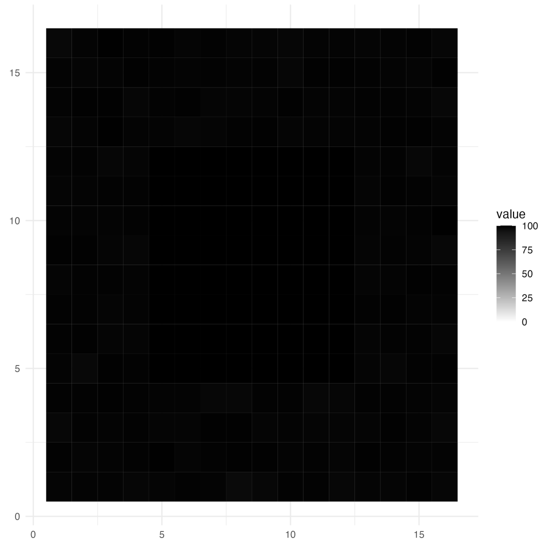

Figure 7: Percentage of significant p-values across pixels in LASSO.

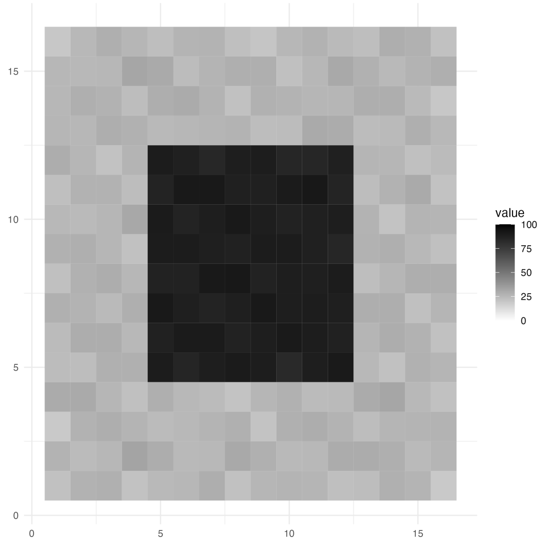

Figure 8: Percentage of significant p-values after correction across pixels in LASSO.

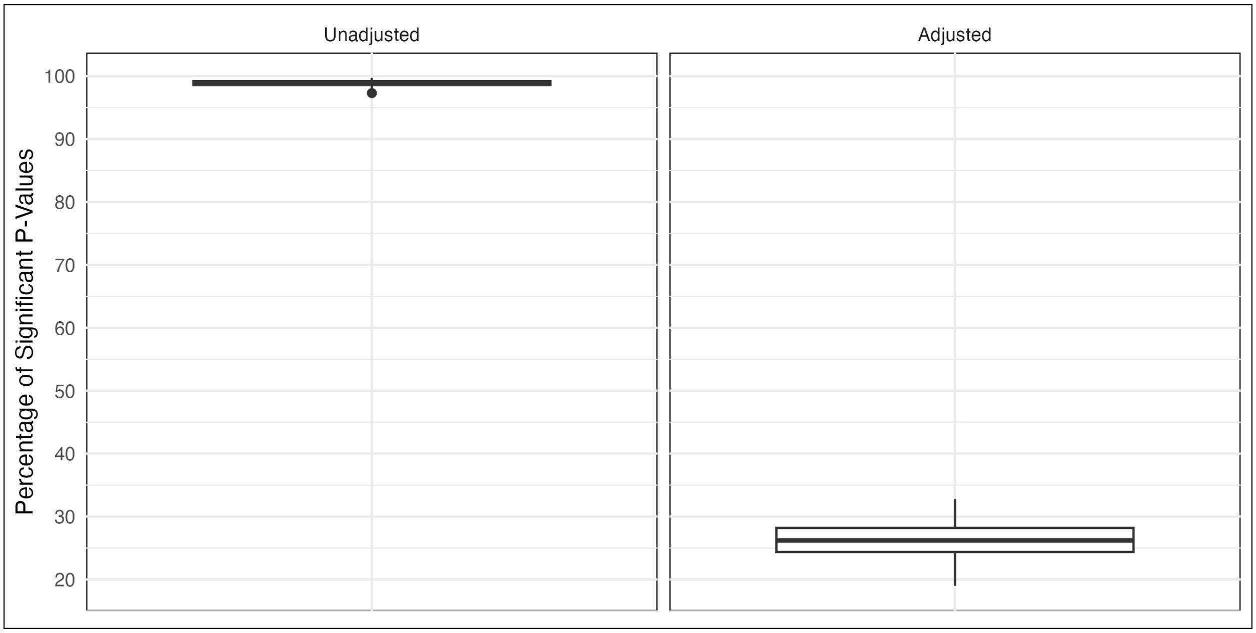

Figure 9: Percentage of significant p-values in outer area before (left) and after (right) adjustment from the LASSO models.

Simulation 1

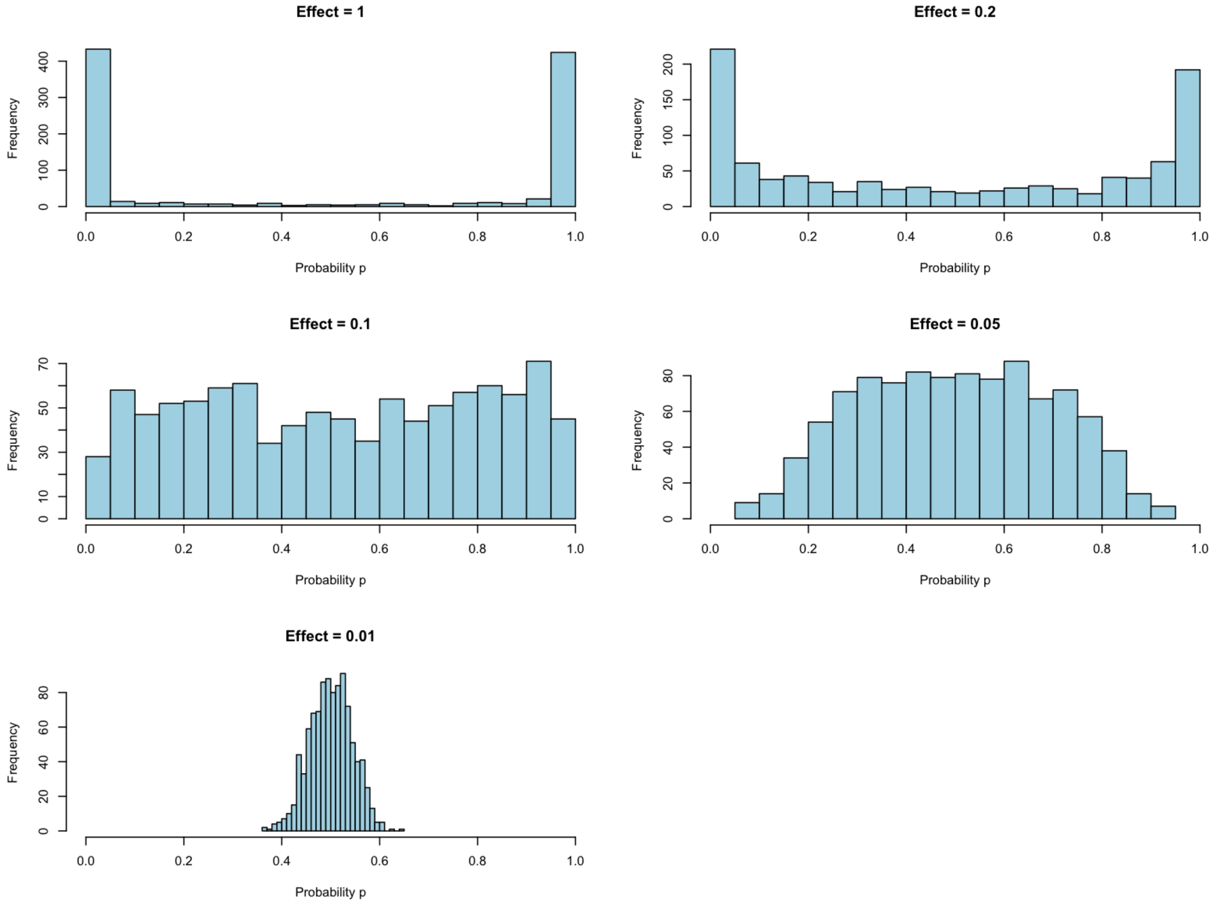

Figure 10: The distribution of p at different β values. $\beta = 0.1$ was chosen for model fitting as it gives the most evenly distributed values.

Figure 10 above shows the distribution of p for different values of β. The choice of $\beta = 0.1$ ensures an even spread of probabilities.

When using the λ given the lowest deviance and the lowest deviance plus 1 standard error, the average prediction accuracy on the test set is 72.5% (min-max: 61.0% - 82.0%) and 72.2% (59.0% - 82.0%). The corresponding AUC is 0.80 (SD=0.03) and 0.80 (SD=0.04). This indicates that the model is reasonably accurate and has a good discriminative ability.

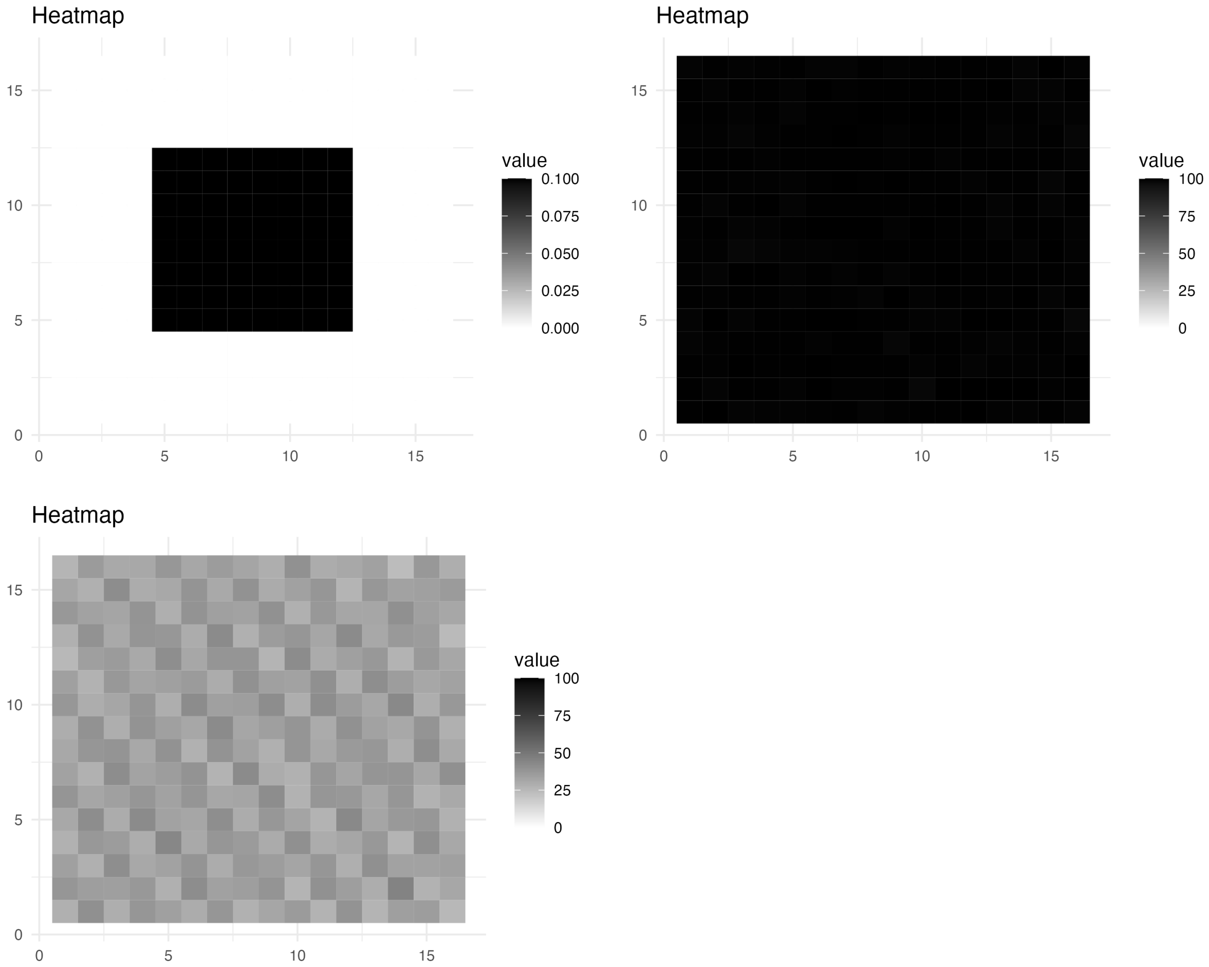

Figure 11: Heatmap of the actual β values, the percentage of getting a significant p-value for each pixel, and the percentage of getting a significant p-value after multiple comparisons adjustment.

Figure 11 illustrates the true β values, the percentage of significant p-values, and the adjusted significance post-multiple comparisons. The results show that due to the correlation structure in X, the LASSO model struggles to pinpoint the exact pixels with non-zero coefficients, resulting in diffuse significant p-values rather than precise identification.

Simulation 2

Simulation 2 differs from Simulation 1 by transforming the covariate matrix into the frequency space using eigenvectors and assuming a sparse coefficient vector in this transformed space. The covariate matrix in the frequency space is generated with a multivariate normal distribution with diagonal values equal to 1 and all other values equal to 0. The coefficient vector index with non-zero values was randomly decided and 10% elements are non-zero. The value of non-zero values was chosen similarly to Simulation 1.

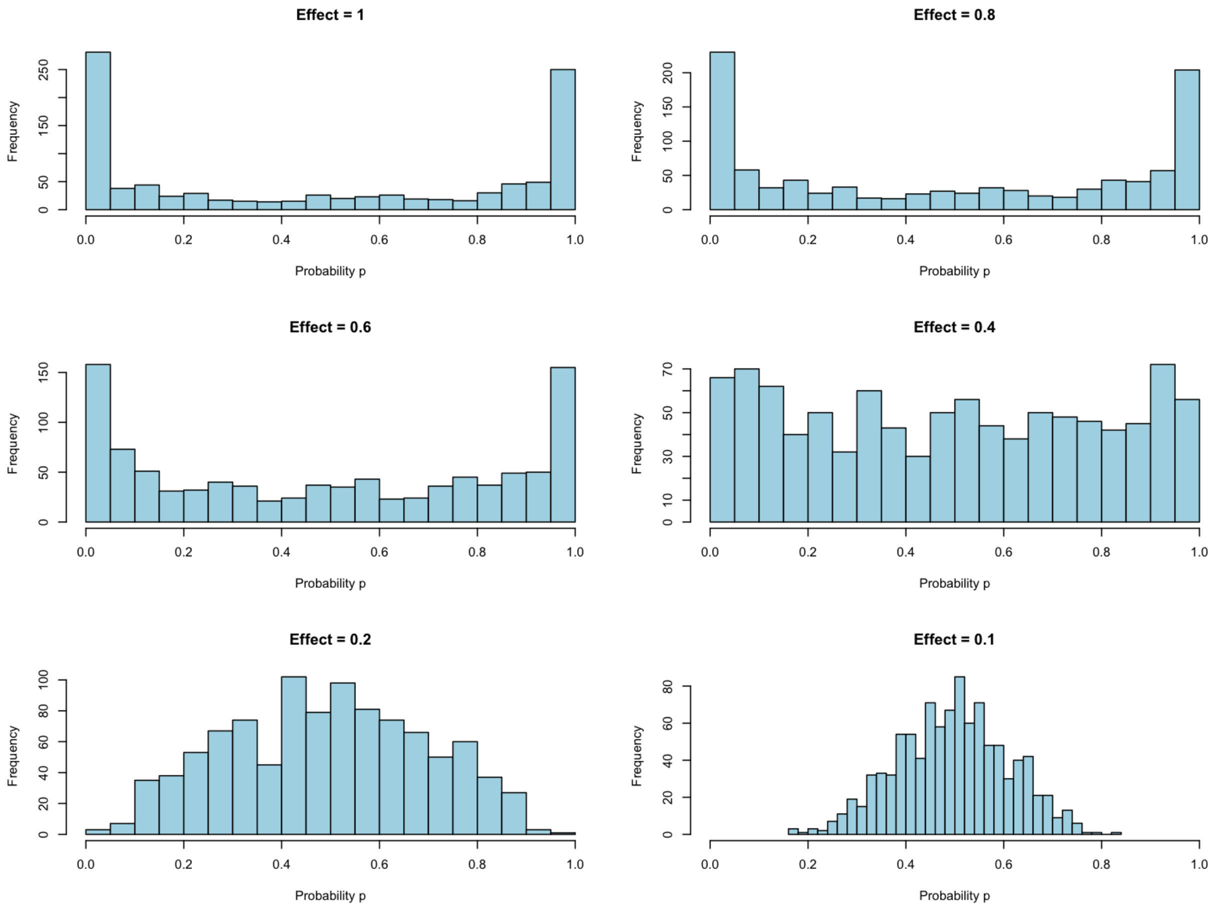

Figure 12: Distribution of p values in the frequency space when the non-zero value equals 0.4, showing a relatively balanced distribution.

Figure 12 shows that when the non-zero value equals 0.4, the distribution of p values is relatively balanced, making it a suitable choice for the model.

Since the correlation between covariates was resolved by eigen-transpose, both accuracy and AUC are better than Simulation 1. The accuracy using the lowest deviance λ is 74.9% (min-max, 66.0% - 83.5%), and 75.1% (min-max, 65.5% - 84.5%) for λ for lowest deviance plus 1 standard error. The corresponding average AUC is 0.83 (SD 0.03) and 0.83 (SD 0.03). These results demonstrate that transforming the data into the frequency space can enhance model performance by reducing collinearity among covariates.

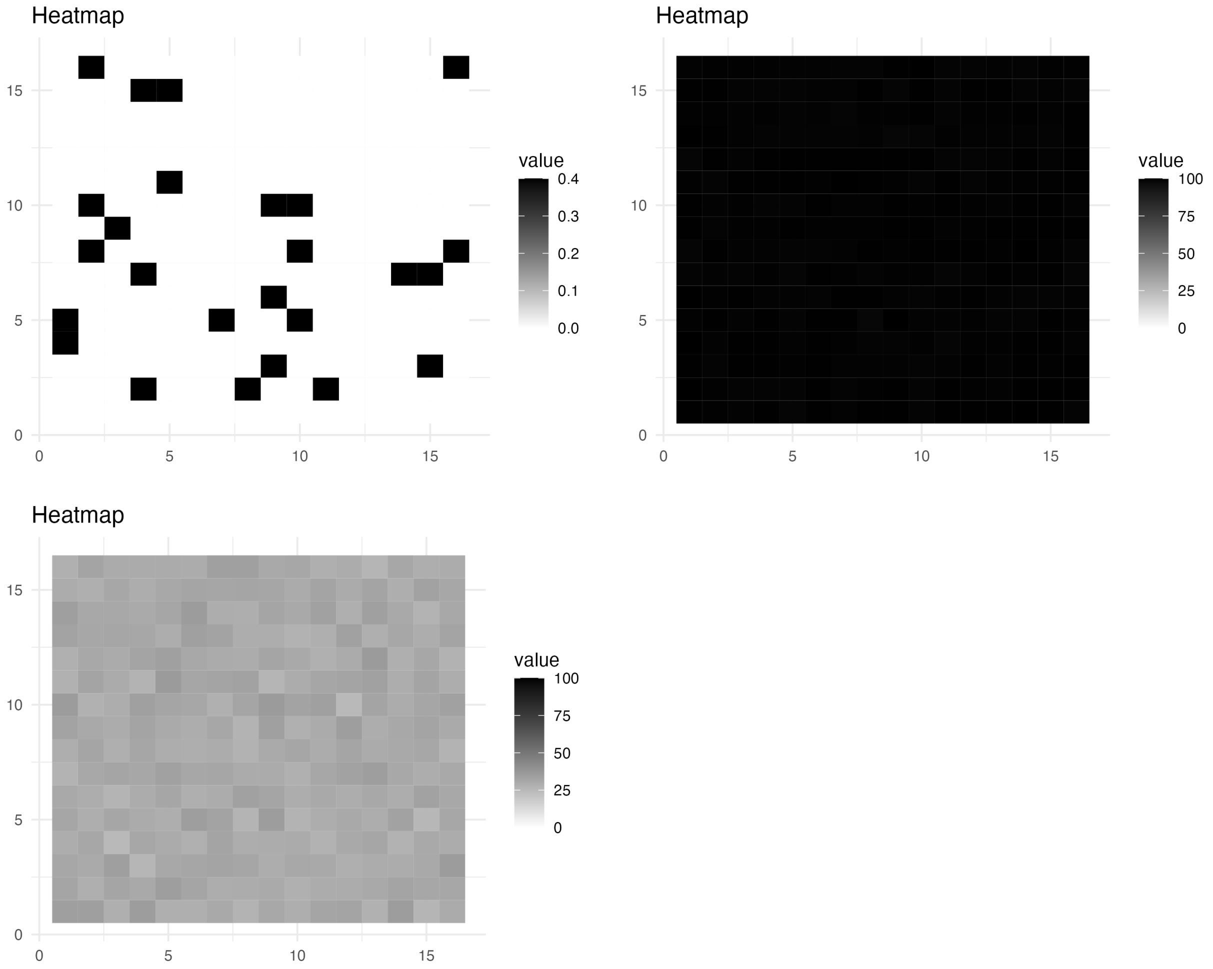

Figure 13: Heatmap of the coefficient vector in the frequency space, the percentage of getting a significant p-value for each pixel, and the percentage of getting a significant p-value after multiple comparisons adjustment.

The heatmap for Simulation 2 (Figure 13) shows the coefficient vector in the frequency space, significant p-values, and adjusted p-values. Despite the improvement in model performance, the heatmap reveals that the model still faces challenges in precisely identifying all non-zero coefficients.

Methods

In this study, we conducted two simulations to evaluate the performace of LASSO models in identifying important features in high-dimensional data. The first simulation assumes sparsity of the features in the pixel space, while the second simulation assumes sparsity in the frequency space.

The pixel space refers to the original high-dimensional space where each dimension represents a pixel in an image. In this space, features are directly observed and may have inherent correlations. Conversely, the frequency space is a transformed version of the pixel space, obtained through techniques like eigen decomposition.

Suppose X is a column vector representing 256 pixels. Its covariance matrix, Σ, is defined to have an exponential correlation structure, where $\Sigma_{ij} = -\exp(\text{dist}(i, j))$. Here, $\text{dist}(i, j)$ is the distance between the pixels i and j in a 16×16 matrix.

Let V be the matrix of eigenvectors of Σ, with each column representing an eigenvector. We can transform the random vector X into the frequency space by $X_{\text{freq}} = V^T X$. The covariance matrix of X_{freq} is given by $\text{cov}\left(X_{\text{freq}}\right) = V^T \Sigma V$, which is a diagonal matrix.

For the simulations, in each iteration, we randomly generate X_{freq} from a multivariate normal distribution with the covariance matrix $operatornamecov\left(X_{\text{freq}}\right)$. We repeat this process 1000 times. Then, we calculate X as $X = V X_{\text{freq}}$.

In the first simulation, we assume sparsity in the coefficient vectors in the pixel space. The coefficient vector β was specified to have non-zero values exclusively within a central 8×8 region. The response variable y was drawn from a binomial distribution with success probabilities determined by $\eta = X\beta$. The non-zero coefficients in β were chosen such that the probability $p = \frac{1}{1+\exp(-\eta)}$ was uniformly distributed across interval $[0, 1]$.

In the second simulation, we assume sparsity in the coefficient vectors in the frequency space. We defined a sparse coefficient vector b in the frequency space, where most of the 256 entries were zero and a randomly 10% were non-zero. The response variable y was generated similarly to the first simulation, ensuring $p = \frac{1}{1+\exp(-\eta)}$ was evenly distributed.

For both simulations, we fit two models: one using the covariates in the pixel space and another using the covariates in the frequency space. Each dataset, generated in size 1000×256 and representing images of 16×16 pixels, was split into training (80%) and test (20%) sets. The regularization parameter λ was tuned using cross-validation with the default binomial deviance metric. The dataset was divided into 10 folds, with the model trained and validated iteratively across these folds, varying λ. The optimal λ was chosen based on the lowest average binomial deviance.

After selecting the optimal λ, model performance was evaluated using accuracy and AUC metrics. Additionally, a permutation test was conducted 100 times to calculate p-values for each covariate. Across all iterations, we calculated the mean and standard deviation of the metrics, as well as the percentage of significant p-values for each covariate.

Results