# Simulated data structures

This section outlines in a unified fashion how we will simulate data without the specifications of each simulation context.

We will simulated $n = n_A + n_B$ images, where $n_A$ is the number of images from group $A$ and $n_B$ is the number of images from group $B$. Let $y_i$ be the $i$th image structured as an $s \times 1$ vector corresponding with the $s$ pixels/voxels of the image. Let $z_i$ be 1 if image $i$ is in group $A$ and 0 if image $i$ is in group $B$. We specify two covariance matrices $\Sigma_g = Q_g \Lambda^g Q_g^T$ for $g = A, B$ with matrix of eigenvectors $Q_g$ and eigenvalues $\{\Lambda_{ii}^g\}_{i=1}^s$, and a group effect $\beta \in \mathbb{R}$. Then we simulate images independently from the model

$$y_i = \beta z_i + \epsilon_i \qquad\qquad \epsilon_i \sim \text{MVN}(0, \Sigma_{g_i}),$$

equivalently

$$y_i \sim \text{MVN}(\beta z_i, \Sigma).$$

A particular simulation context will be defined through specification of the following:

(a) An 'image space.' Unless otherwise stated, we'll be talking about images of the same dimension as the handwritten digit data. Thus, $s = 256$ pixels arranged in a $16 \times 16$ lattice.

(b) Desired number of observations per group, $n_A$ and $n_B$

(c) Desired covariances $\Sigma_g$, either by providing a model (e.g. exponential) or by providing eigenvectors $Q_g$ and eigenvalues $\Lambda_g$. Unless otherwise specified, we will assume $\Sigma_A = \Sigma_B$.

(d) The group effect $\beta$. We might choose $\beta$ in a couple of ways, such as by a function of pixel location (e.g. 1 for all pixels in the top half of the image and -1 for all pixels in the bottom half), as a linear combination of ESF/GSP eigenvectors, or empirically (using the empirical covariance matrix, maybe estimated with regularization).

# Models

(a) Predicting images $y_i$ from group $z_i$/Inferring group effect

  (1) VBM

  (2) spVBM

    i. Only positive eigenvalue eigenvectors?

    ii. Only a subset of eigenvectors?

    iii. Knots or exact computation?

(b) Predicting group $z_i$ from image $\tilde{y}_i$ using sparse logistic regression where $\tilde{y}_i$ is a transformation of the image $y_i$

  (1) Voxels as covariates $\tilde{y}_i = y_i$

  (2) functional PCs as covariates

(3) Frequency intensities as covariates (images after application of ESF or GSP transformation, e.g. $\tilde{y}_i = Q_{ESF} y_i$)

(4) One other method?

(c) Inferring network

(a) Do we have a specific methodology picked out to look at?

## Specific simulations of interest

1. First simulation

**Data** Let $n_A = n_B = 1000$ where $\beta$ is 1 for pixels in the center $8 \times 8$ pixel square and 0 elsewhere, $\Sigma = \Sigma_A = \Sigma_B$ is an exponential correlation matrix with rate 1.

**PredictImage** Fit VBM and spVBM models predicting images $y_i$ from group $z_i$ using all eigenvectors from 16 knots. Report the same performance metrics as in Sarah's paper. Use an exponential network.

**PredictGroup** Train sparse logistic regression models on 800 observations (400 per group) predicting group $z_i$ from image covariates $\tilde{y}_i$ for the following transformations:

i. Voxels as covariates (no transformation)

ii. ESF frequencies as covariates (transformation from exponential network)

iii. GSP frequencies as covariates (transformation from exponential network, unnormalized Laplacian)

iv. functional PCs as covariates (talk with Yue for details)

Report test AUC, sensitivity, and specificity using remaining 200 observations. Report also which covariates were selected in the voxel, ESF frequency, and GSP frequency models.