Here, we apply the model X framework (Barber and Candes, 2015) to identifying transcription factors associated with disease, based on the expression of their target genes. The method assumes we have a ranked list of target genes for each transcription factor 1, …, *p,* gene expression count data, $D \in \mathbb{R}^{N \times q}$, and a response variable, Y, such as disease status ($Y \in \{0,1\}^N$). This ranking can come from the output of a machine learning model, such as LINGER (Duren et al, 2024).

First, we perform principal component analysis (PCA) on the top 1000 target genes of each TF. Specifically, for each TF j, we subset from D a scaled and centered count matrix D_j, containing the target genes of TF j, and perform singular value decomposition (SVD):

$$D_j = U^{(j)}\Sigma^{(j)}V^{(j)T}$$

We define the predictor matrix X by taking the first principal component (PC) from each decomposition, representing the overall activity of the j-th TF:

$$X = \left[U_{.1}^{(1)}, \dots, U_{.1}^{(p)}\right]$$

Model X knockoff framework

Letting X_j denote the j-th column of X, we test for conditional independence of X_j and Y, given the other predictor variables, X_{-j}:

$$H_{0,j} : X_j \perp Y \mid X_{-j}$$

The knockoff procedure first characterizes the conditional distribution of X_j given the other predictors. We start by generating a conditional distribution $P\left(X_j \middle| X_{-j}\right)$, such that the swap property holds:

$$\left(X, \tilde{X}\right)_{Swap(S)} = \left(X, \tilde{X}\right): \ \tilde{X} \sim P(X_j | X_{-j})$$

Where $\tilde{X}$ denotes the knockoff (generated) data, and X denotes the original data matrix. Swap(S) implies that if we swap columns $S \in \{1, \dots, j\}$ of X and $\tilde{X}$, they are equal in distribution.

As $\tilde{X}$ is constructed irrespective of Y, the predictor-response relationships between Y and the knockoffs are conditionally independent, given the real data. A formal proof is provided in Barber and Candes 2016. Once this is established, we can run a model to predict Y from the concatenated matrix $X_{KO} = [X, \tilde{X}]$:

$$Y = f(X_{KO}) + \epsilon$$

$$\epsilon \sim P(\theta)$$

Where f denotes some arbitrary model (LASSO, ridge regression, neural network, etc), ϵ denotes the error term belonging to some distribution parametrized by θ. We derive an importance metric, $W_j, \widetilde{W_j}$, for the j-th feature, corresponding to the real variable and the knockoff, respectively. In our case, we use the SHAP value of each feature for W. Under the model X framework, the test statistic, $V_j = W_j - \widetilde{W_j}$ should be symmetric about 0 for null features, and large and positive for true predictors.

Given this information, the distribution of $V_j's$ should be a mixture of null features (symmetric, mean 0, low variance) and true features (large, positive). The FDR at threshold t can be calculated as:

$$\text{FDR}(t) = \frac{\sum_j 1\{V_j \leq -t\}}{\sum_j 1\{V_j \geq t\}}$$

Here, we take advantage of the fact that null features are symmetric about 0, while alternative features are strictly positive. We can estimate the null proportion by the number of features with V_j less than $-t$, because only null features have negative values.

Constructing the model X knockoff

The latter procedure is reliant on the fact that we can construct a reliable joint distribution for X. To do this, we apply the sequential conditional independent pairs procedure, outlined in Barber and Candes 2015.

Step 1 – sampling of the first variable

For $j = 1$, we sample conditional $\widetilde{X_1}$

$$\widetilde{X_1}|X_{-1} \sim N\big(f_1(X_{-1}), \exp\big(g_1(X_{-1})\big)\big)$$

Where f and g are arbitrary models describing the conditional mean and variance, respectively. In this repository, we provide model X knockoff frameworks where f and g are gradient boosted machines. We estimate f and g by optimizing:

$$f_1 = argmin_{f_1}\left\{\sum_{i=1}^{N}\left(X_{i,1} - f_1\left(X_{i,-1}\right)\right)^2\right\}$$

$$g_1 = argmin_{g_1}\left\{\sum_{i=1}^{N}\left(\log\left(X_{i,1} - f_1\left(X_{i,-1}\right)\right)^2 - g_1\left(X_{i,-1}\right)\right)^2\right\}$$

Step 2 – sequential sampling of remaining variables

For $j = 2, \ldots, p$, we sample conditional $\tilde{X}_j$'s :

$$\tilde{X}_j | X_{-j}, \tilde{X}_{1:(j-1)} \sim N\left(\ f_j\left(\left[X_{-j}, X_{1:(j-1)}\right]\right), g_j\left(\left[X_{-j}, X_{1:(j-1)}\right]\right)\right)$$

Where f_j and g_j are estimated as

$$f_j = argmin_{f_j}\left\{\sum_{i=1}^{N}\left(X_{i,j} - f_j\left(X_{i,-j}, \tilde{X}_{i,1:(1-j)}\right)\right)^2\right\}$$

$$g_j = argmin_{g_j}\left\{\sum_{i=1}^{N}\left(\log\left(X_{i,j} - f_j\left(\left[X_{i,-j}, \tilde{X}_{i,1:(1-j)}\right]\right)\right)^2 - g_1\left(\left[X_{i,-j}, \tilde{X}_{i,1:(1-j)}\right]\right)\right)^2\right\}$$

FDR control of features

We now have a model X knockoff, $X' = [X, \tilde{X}]$, satisfying the swap property. Next, we train a predictive model and calculate a measure of feature importance for each variable of X'. In our case, we use the Shapley value for each feature.

$$W_j = SHAP(X', h(.|\theta), j)$$

$$\widetilde{W}_j = SHAP(X', h(.|\theta), j + p)$$

$$\theta = argmin_\theta\{\mathcal{L}(Y, h(X'|\theta))\}$$

Where $SHAP(.,.,.)$ is the mean absolute Shapley value calculated from the data X' and the model, $h(.|\theta)$, for the j-th feature. θ denotes the model parameters, Y is the response variable,

and $\mathcal{L}(.)$ is the loss function. In this repository, we use binary cross entropy (as Y is categorical disease status), but this can be anything, depending on the data type.

We calculate the FDR at threshold $t \geq 0$ as

$$FDR(t) = \frac{\#\,(null\ features \geq t)}{\#\,(features \geq t)} = \frac{\#\left(W_j \leq -t\right)}{\#\left(W_j \geq t\right)}$$

Where we use symmetry of the null distribution to equate:

$$\#\,(null\ features > t) = \#\left(W_j \leq -t\right)$$

We identify disease-associated TFs by selecting those with $W_j \geq t$ for a threshold t corresponding to an estimated $FDR \leq 0.2$.