

Supplement to “Explainable Machine Learning for Functional Data”

Katherine Goode, J. Derek Tucker, Daniel Ries, and Heike Hofmann

This document contains additional applications of the VEESA pipeline corresponding to the article “Explainable Machine Learning for Functional Data”. Section 1 compares the VEESA pipeline to the cross-sectional approach with the shifted peaks data. Section 2 contains additional details about the H-CT material classification example. Lastly, Section 3 contains additional details about the inkjet dataset example. All code associated with the analyses in the main paper and supplement are available at github.com/sandialabs/veesa/tree/master/demos/goode-et-al-paper.

1 Comparing VEESA Pipeline to Cross-Sectional Approach with the Shifted Peaks Data

The cross-sectional approach treats the observations across functions at each time point as the predictor variables. A post-hoc explainability method such as PFI may then be applied to try to identify the times that are important to the model for prediction. However, this approach presents a disadvantage. Due to the nature of functional data, the cross-sectional predictor variables are likely to be correlated. For example, consider the shifted peaks data described in Section 2 of the main text. Figure S1 shows a heatmap of all pairwise Spearman correlations between the cross-sectional shifted peaks data predictor variables. There are strong positive and negative correlations for almost all variables. As mentioned in the main text, correlation between predictor variables has been shown to lead to biased PFI results. Here, we apply the cross-sectional modeling approach to the shifted peaks data and compare the results to the VEESA pipeline. We highlight the difficulties with gaining insight to the model produced by the cross-sectional approach.

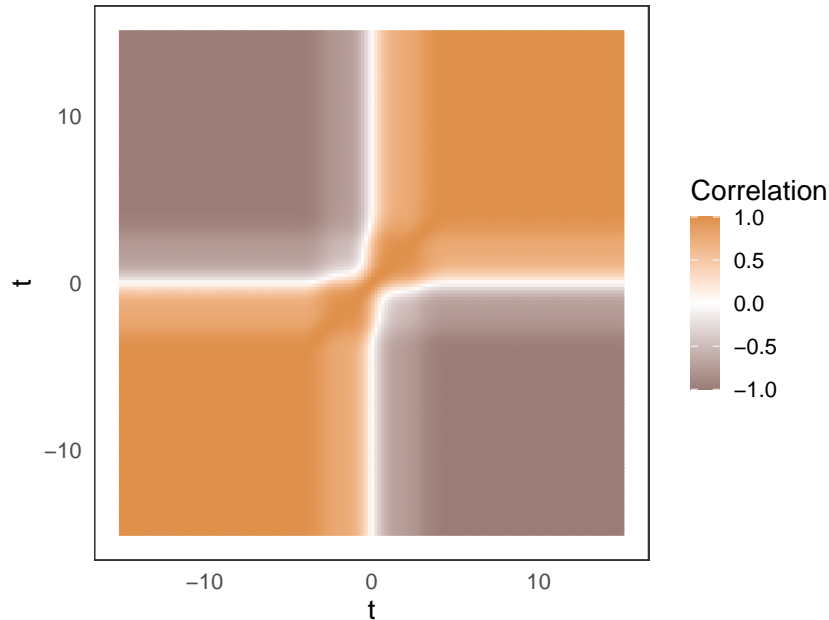


Figure S1: Spearman correlations between all pairs of cross-sectional variables from the shifted peaks data.

The shifted peaks data are separated in the same training and testing datasets with 400 and 100 observations, respectively. A random forest is fit to the training data (*randomForest* R package). The cross-sectional variables (one associated with each of the 150 times where the functions are observed) are treated as the model inputs, and the group associated with a function is treated as the model output. The default tuning parameters options are used to mimic the VEESA pipeline random forest from the main text. The random forest accuracy on the test data is 1, which agrees with the the result from the VEESA pipeline random forest. PFI is applied to the test data using a metric of accuracy with 10 replications to mimic the computation of PFI in the VEESA pipeline from the main text.

Figure S2 (top left) includes the true group means depicted as solid lines and the cross-sectional group means depicted by points. The error bars represent plus/minus one cross-sectional standard deviation. As seen in Section 2, the cross-sectional means do not capture the shape of the true means. Figure S2 (top right) depicts the PFI values computed using the cross-sectional approach. Unlike the PFI results from the VEESA pipeline method (Figure 3), the PFI values for all variables (time points in this instance) are approximately 0 with no variability (i.e., all times have standard deviations across PFI replicates of 0). In this instance, the PFI results do not appear to be inflated due to bias. Instead, this result suggests that none of the individual time points are important in regard to the accuracy of the model. These PFI results indicate that permuting one time point, while leaving all other time points as observed, does not affect the model enough to alter the accuracy of the model on the test data. This information is unhelpful in identifying the aspects of the data that are important to the model for prediction. The application of the VEESA pipeline to this data, described in Section 3, produces non-zero PFI values, and the interpretation of the most important fPC (jPC 1) provides a clear explanation as to the aspect of the data that is important to the model for prediction.

Since the predictor variables in the cross-sectional approach are highly correlated, we suspect that the non-zero PFI values are the result of accuracy being used as the metric. Consider the definition of accuracy. For a set of $i = 1, \dots, n$ observations, let y_i be the observed value and \hat{y}_i be the predicted value for observation i . Accuracy is computed as

$$Accuracy(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n I(\hat{y}_i = y_i).$$

With the shifted peaks data, the predicted value is whether a function belongs to group 1 or 2. To compute the PFI for a time point, the observations across the simulated functions at that time are permuted. With the remaining 149 time points being highly correlated with the permuted variable, enough information is likely provided to the model to produce the same prediction as to which group the function belongs to. However, a metric for a binary response variable based on the model probability that a function belongs to a group is more likely to be affected by the permutation of a single time point. One example of such a metric is log-loss. For $y \in \{0, 1\}$ and $p = P(y = 1)$ (estimated from a model), log-loss is defined as

$$LL(y, p) = y \log(p) + (1 - y) \log(1 - p).$$

To test this idea, we compute new PFI values using the log-loss metric for the random forest from the cross-sectional approach and the random forest in the VEESA pipeline. Both sets of PFI values are computed on the test data. Figure S2 (bottom) depict the PFI values computed for the cross-sectional (left) and VEESA pipeline (right) approaches, respectively, using log-loss. As suspected, the log-loss metric produces non-zero values for the cross-sectional approach. However, note the difference in y-axis between the cross-sectional and VEESA pipeline PFI values. The cross-sectional PFI values are much smaller than the VEESA pipeline PFI values (i.e., the model predicted probability is much less affected when a variable is permuted). The cross-sectional PFI values suggest that the times between, approximately, $t = -2$ and $t = 2$ have little importance. The time points outside of this region have non-zero values. We may expect the times between -2 and 2 to have little importance since there is a lot of overlap between the two group during this time. We may also expect the times in the ranges of $t \in (-7, -2)$ and $t \in (2, 7)$ to be important since the groups are distinguished by their differing peaks associated with these intervals. However, the PFI values continue to be non-zero in the below -7 and above 7, where the the functions all have values of approximately zero. This result either suggests that the model is doing a poor job of using the information from the predictor variables, or more likely, these PFI values are biased due to the correlation between predictor variables.

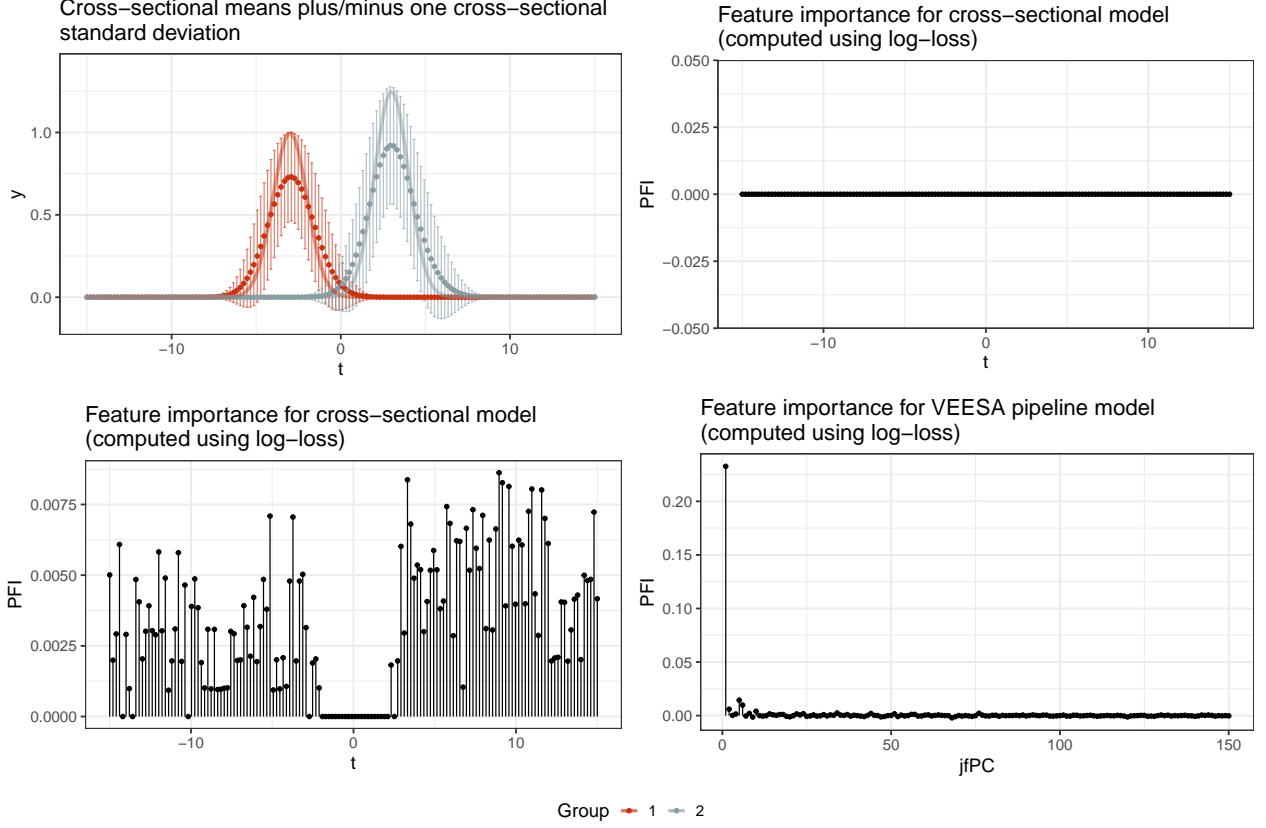


Figure S2: (Top Left) The solid lines represent the true group means. The dots and error bars represent the cross-sectional group means plus and minus one cross-sectional standard deviation. (Top Right) PFI values computed on the test data using the metric of accuracy for a random forest trained using the cross-sectional approach. (Bottom Left) PFI values computed on the test data using the metric of log-loss for a random forest trained using the cross-sectional approach. (Bottom Right) PFI values computed on the test data using the metric of log-loss for a random forest trained using the VEESA pipeline.

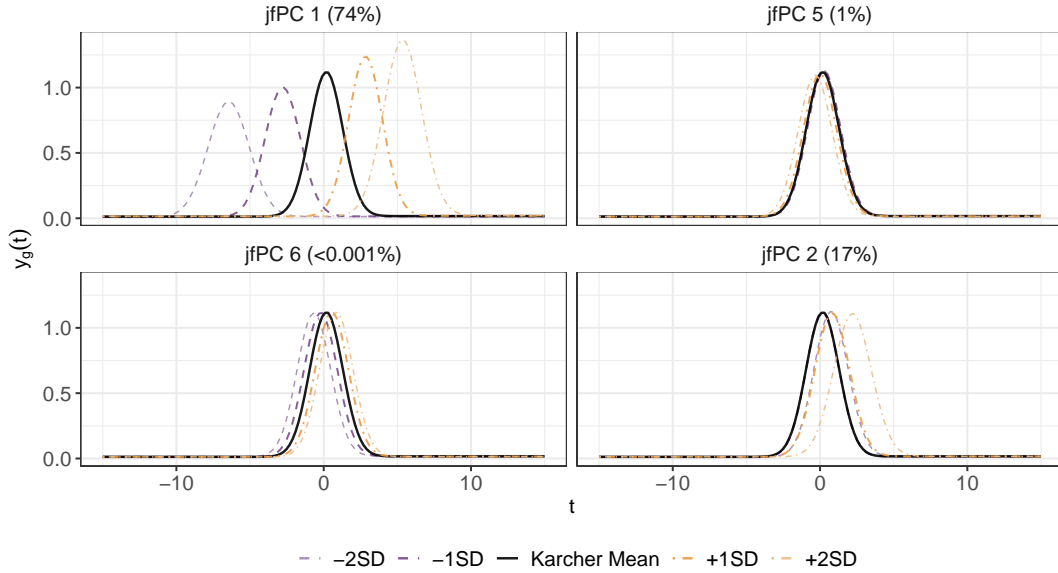


Figure S3: Principal direction plots of the VEESA pipeline jfPCs with log-loss based PFI values greater than 0.005 (excluding jfPC 1, which is shown in the main text).

The VEESA pipeline PFI values computed with accuracy (Figure 3) only found jfPC 1 to be important on the test data. The log-loss metric identifies additional principal components as important, but jfPC 1 remains the most important. Figure S3 includes the principal direction plots for the four principal components with PFI greater than 0.005 (computed using log-loss). Recall that jfPC 1 captures variability in both the horizontal and vertical directions. The other three principal components capture much more nuanced types of variability but mostly consider small types of horizontal variability. In this example, the test data accuracy is equivalent between the VEESA pipeline and cross-validation approaches, but the VEESA pipeline feature importance results are more meaningful.

To summarize, there are several concerns with the cross-sectional approach:

- The functional nature of the data is ignored. By treating each time point of samples as a predictor variable, the model is not aware of the relationship between samples within a function. By presenting the functions to the model in this form, information about the functional shape is lost. This may be a problem when applying a model to unseen data.
- The horizontal variability in the functions is ignored. In the shifted peaks data, the true functional means have different peak times, which leads to horizontal variability in the functions. The cross-sectional approach only uses the variability in the functions in the vertical direction to discriminate between the two classes. This effect is seen in the cross-sectional group means in Figure S2. The shapes of the cross-sectional means do not accurately reflect the true functional means, because only the variability in the vertical direction is accounted when computing cross-sectional means in this manner.
- Correlation between the predictor variables can lead to bias in PFI. In practice, it may not be clear which PFI values are biased, so it becomes difficult to trust the results.

All three of these concerns are addressed by the VEESA pipeline.

2 H-CT Material Classification Additional Analyses

Additional details about the analysis of the H-CT material classification data are provided here. In particular, we present the model accuracies from all methods (VEESA pipeline and cross-sectional approaches) and box-filter runs (Section 2.1), the clocked times of the steps in the VEESA pipeline from an implementation on the H-CT data (Section 2.2), the PFI variability from the implementations of the VEESA pipeline included in the main text (Section 2.3), additional results from the vfPCA analysis (Section 2.4), and results from the jfPCA analysis (Section 2.5).

2.1 Smoothing Selection and Cross-Sectional Comparison

We consider a range of box-filter runs and select the value to use in the main text based on the best test data accuracy. We implement the VEESA pipeline using jfPCA, vfPCA, and hfPCA and several variations of the cross-sectional approach for comparison. The details of the implementations for smoothing, the VEESA pipeline, and the cross-sectional approaches are as follows.

Smoothing The number of times the box-filter is run affects the smoothness of the functions (i.e., more runs lead to smoother functions). Here, we consider 1, 5, 10, 15, 20, and 25 runs of the box-filter.

VEESA Pipeline The VEESA pipeline is applied to all versions of the smoothed data using jfPCA, vfPCA, and hfPCA. A neural network is used as the model for each scenario. All models are trained with the default settings in *scikit-learn* (one layer with 100 neurons and a ReLU activation function).

Cross-Sectional Approach The cross-sectional approach is applied for comparison to the VEESA pipeline. Three variations in regard to smoothing are considered: applying the cross-sectional approach with (1) no smoothing, (2) after smoothing, and (3) after smoothing and ESA alignment. Again, neural networks are used as the model (trained using *scikit-learn* with default settings).

The model accuracies versus the number of box-filter runs are depicted in Figure S4 for all implementations. The colors represent the method used to process the data before training a neural network. The solid lines and circles indicate accuracy on the training data, and the dashed lines and triangles indicate accuracy on the testing data. The application of the cross-sectional method with no smoothing or alignment returns the highest accuracy. This is followed by the cross-sectional method with smoothing but no alignment. The accuracies returned from the VEESA pipeline approaches are always lower on the testing data than training data. hfPCA results in the lowest accuracies followed by jfPCA, and vfPCA returns the highest accuracy for all box-filter iterations until 25. This is expected since it is understood that the vertical variability in the signatures contains the most information for material classification. The results from the VEESA pipeline using vfPCA when 20 runs of the box-filter are those displayed in the main text.

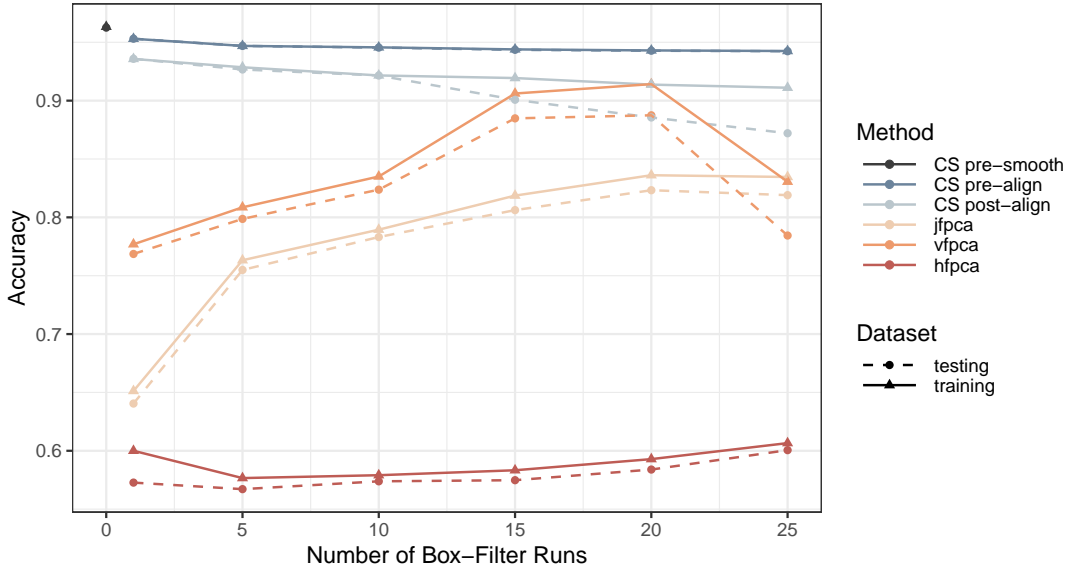


Figure S4: Model accuracies from neural networks applied using the VEESA pipeline and the cross-sectional approach.

2.2 Analysis Times

The VEESA pipeline may be computationally intensive, especially with larger datasets. As an example, here we consider the time it took to implement the VEESA pipeline with the H-CT data with 1 run of the box-filter. The analyses were conducted using the python VEESA code and run with parallelization on a computer with 44 cores and 252 gigabytes of memory. Figure S5 presents the number of hours it took to complete each step in the VEESA pipeline (separately by type of efPCA). The points are colored by training and testing data. Note that the alignment and computation of principal components are computed jointly on the test data, and thus, their time is reported together. The step that took the longest, by far, was the alignment of the training data functions. Figure S6 shows the hours of each step with the alignment process on the training data removed for easier viewing. Here points are colored by the efPCA method used. Note that the computation of the joint fPCs on the training data takes longer than the vfPCs and hfPCs. Finally, Figure S7 shows the hours with the alignment and PCA processes on both the training and testing data removed (again colored by efPCA method). This allows for a better visual of the amount of time it took to train the models and compute PFI. Note that while the proportion of time it takes to train the models and compute PFI is much smaller than the alignment process, some of the model training steps and one of the computations of PFI took over an hour.

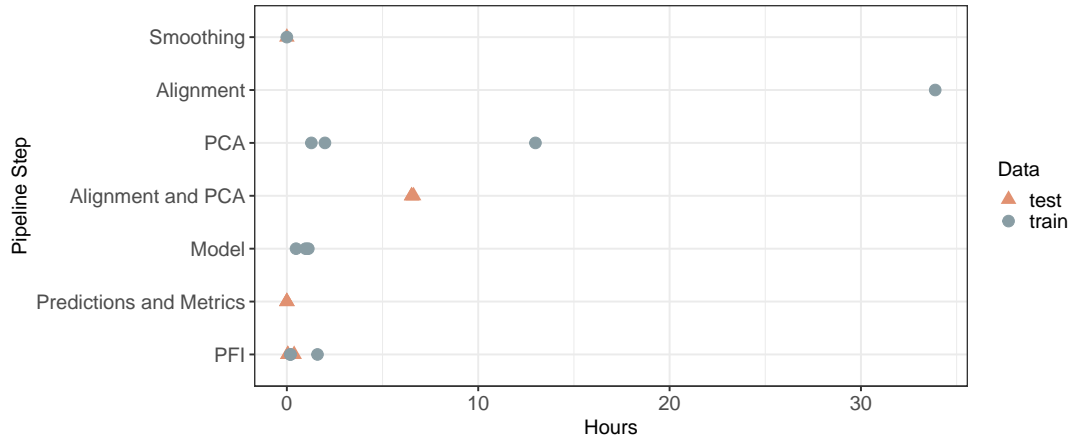


Figure S5: Analysis step times for the VEESA pipeline analyses of the H-CT data.

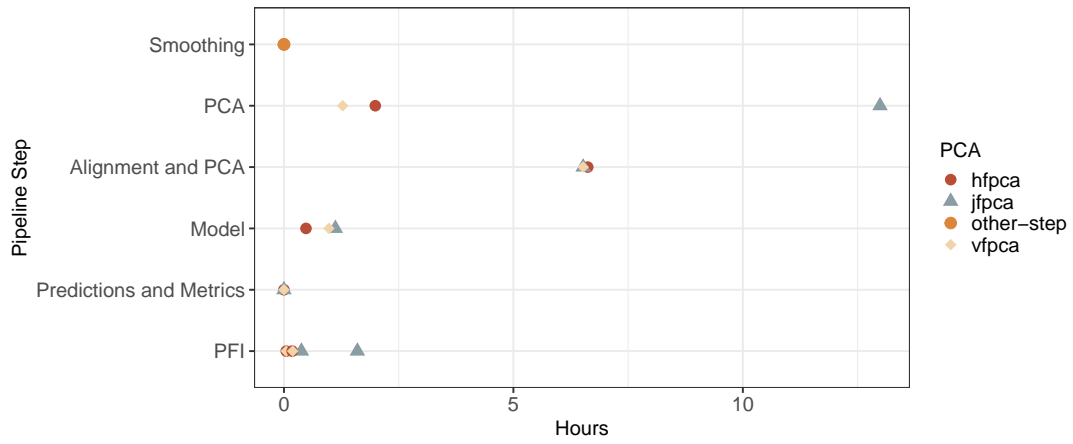


Figure S6: Analysis step times for the VEESA pipeline analyses of the H-CT data (with the alignment of the training data removed).

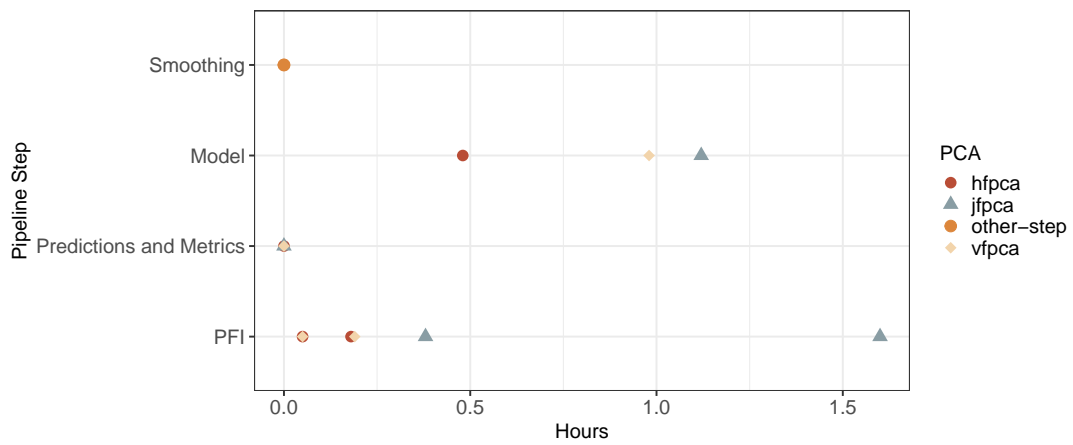


Figure S7: Analysis step times for the VEESA pipeline analyses of the H-CT data (with the alignment and PCA computations of the training and testing data removed).

2.3 PFI Variability

Figure S8 depicts the five feature importance replicates used to compute the PFI results included in the main text. The points are colored by the replicate with an alpha shading that provides transparency. However, all points appear to be the same color, because they are overlapping. The standard deviation between replicates is approximately 0 for all principal components.

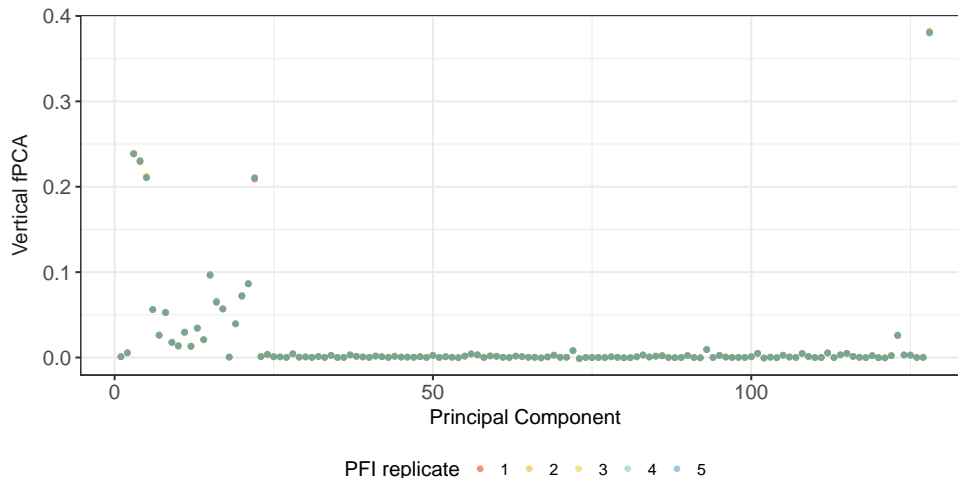


Figure S8: PFI replicate values for vfPCs associated with the H-CT material classification example.

2.4 vfPCA Principal Directions (Additional)

Figure S9 shows the proportion of variation and PFI values associated with the vfPCs described in the main text. Figures S10 and S11 depict the principal direction and principal differences plots of the nine PCs with the highest PFI values from the vfPCA model.

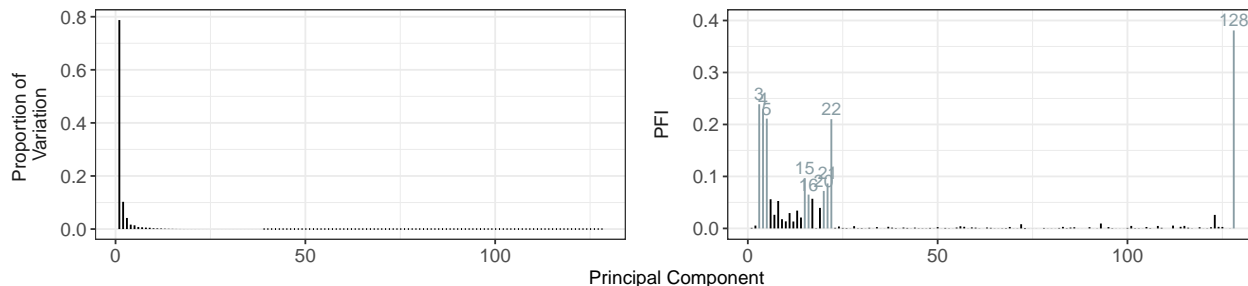


Figure S9: Proportion of variation and PFI values associated with vfPCs. The vfPCs with the nine highest PFI values are labeled and colored.

2.5 jfPCA Principal Directions

Figure S12 shows the proportion of variation and PFI values associated with the best performing model fit with jfPCs. Figures S13 and S14 depict the principal direction and principal differences plots of the nine PCs with the highest PFI values from the jfPCA model.

3 Inkjet Printer Additional Analyses

We include some additional results from the inkjet printer analyses here: details on the cross-validation procedure (Section 3.1), classification performance within each printer (Section 3.2), additional principal

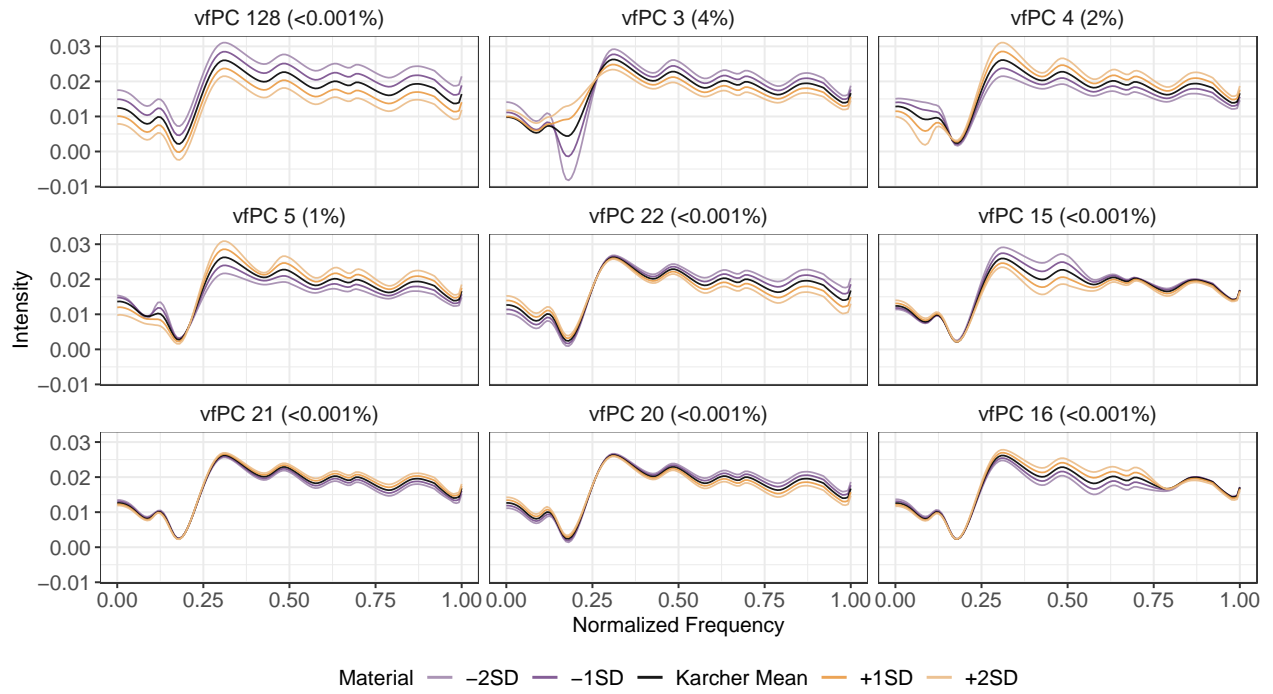


Figure S10: Principal directions from the nine vfPCs with the highest PFI from the H-CT material classification example.

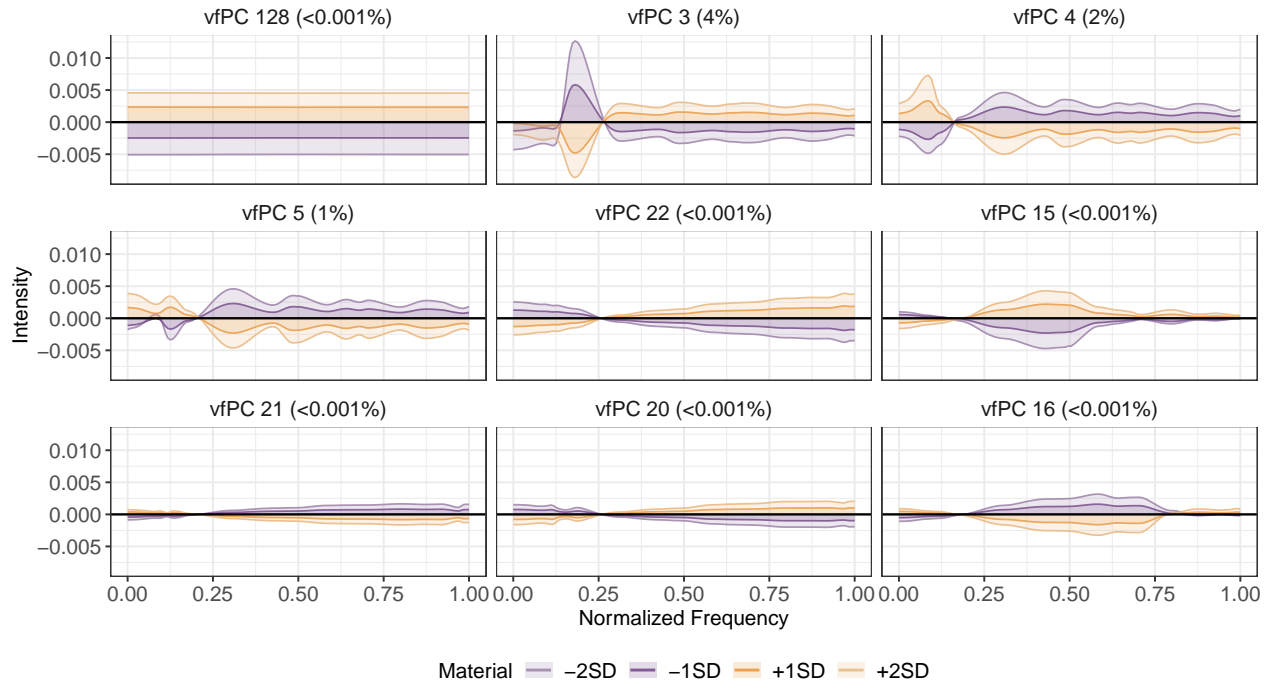


Figure S11: Principal differences from the nine vfPCs with the highest PFI from the H-CT material classification example.

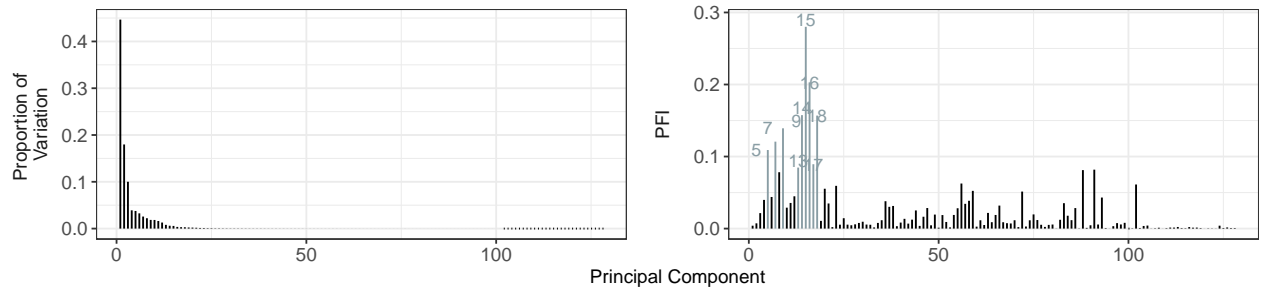


Figure S12: Proportion of variation and PFI values associated with jfPCs. The jfPCs with the nine highest PFI values are labeled and colored.

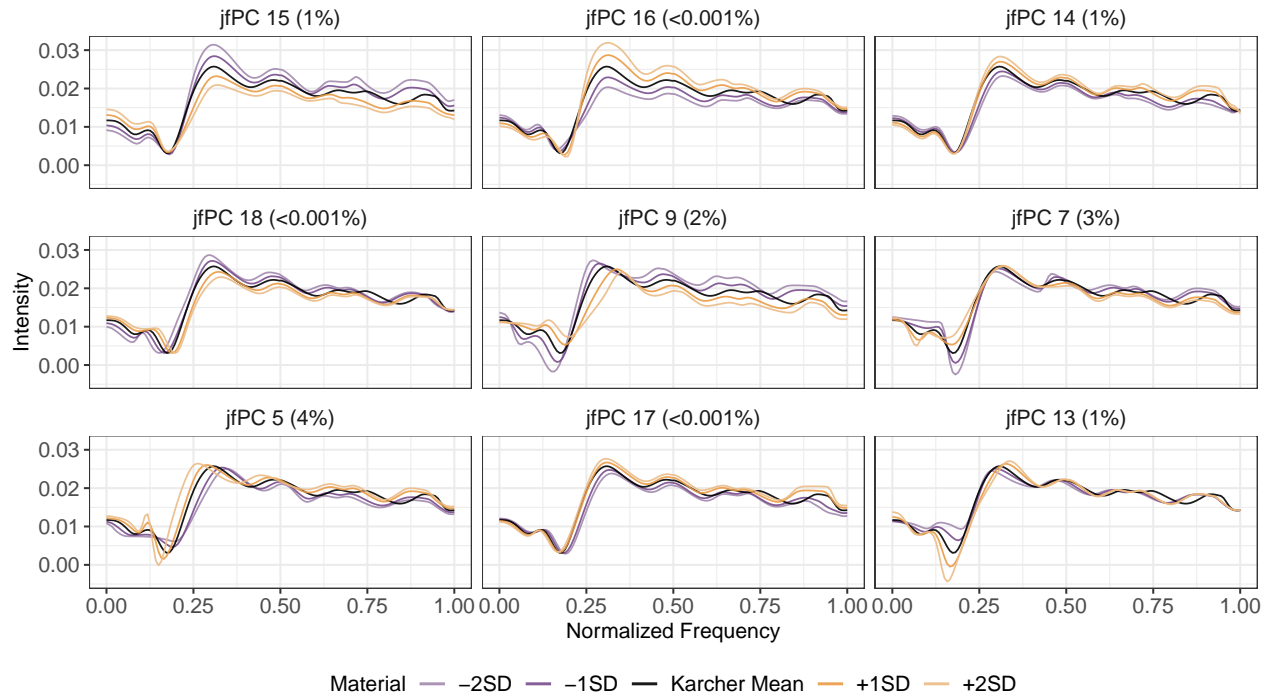


Figure S13: Principal directions from the nine jfPCs with the highest PFI from the H-CT material classification example.

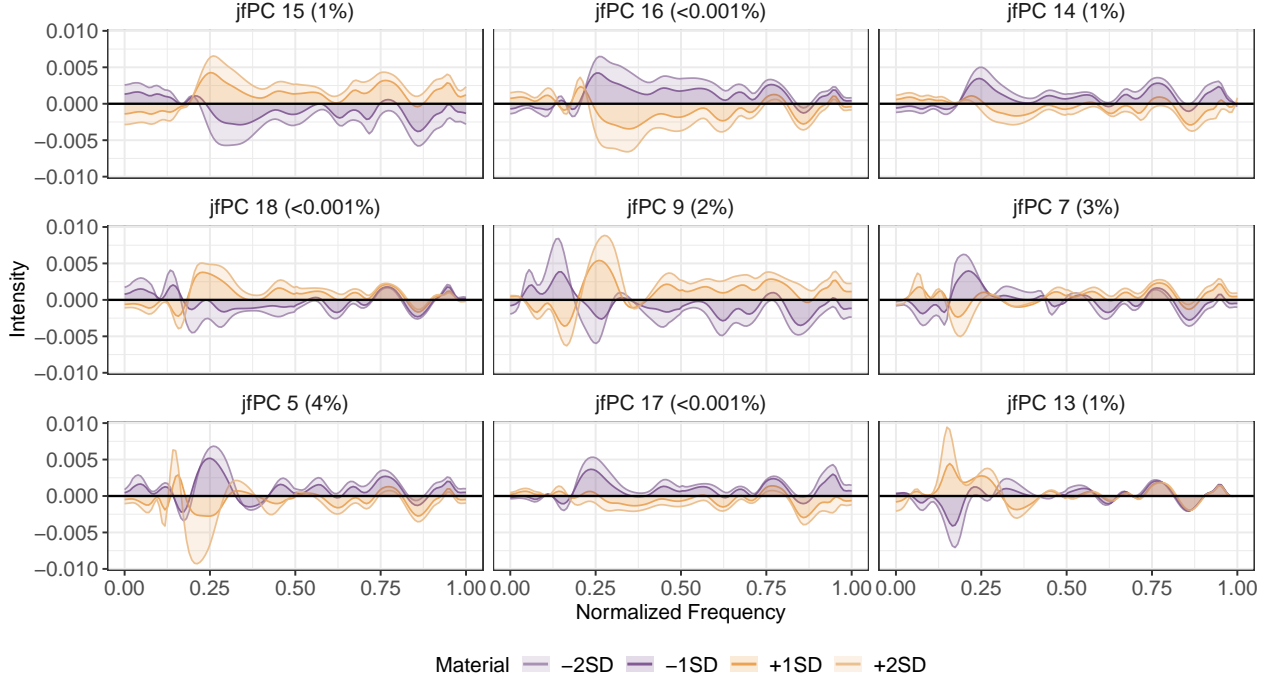


Figure S14: Principal differences from the nine jfPCs with the highest PFI from the H-CT material classification example.

directions for the important jfPCs from the best and worst performing models not included in the main text (Section 3.3), and an implementation of the VEESA pipeline for the worst performing scenarios with a subset of the principal components removed based on the feature importance in the main text (Section 3.4).

3.1 Cross-Validation Procedure Details

Here we describe the cross-validation procedure used to identify the best and worst models. The folds are created such that each replicate for a printer and color is randomly assigned to be in fold 1, 2, or 3 (with 3 functions in fold 1 and 2 functions in folds 2 and 3). Then two of the folds are used to train a model using Steps 1-4 of the VEESA pipeline described in Section 3.1, and the third fold is used for testing Steps 1-4 of the VEESA pipeline described in Section 3.2. The procedure is repeated such that the three possible fold pairs are used for training. Since the fold a signature is assigned to is random, this 3-fold cross-validation procedure is repeated a total of 10 times to account for random variability.

For each color, model accuracy is computed for each test fold and repetition, and the cross-validation predictive performance metric is computed as the average of the test-fold accuracies across the 10 replicates. That is, let y_i represent the true printer associated with observation $i = 1, \dots, 77$, and let $\hat{y}_{i,r}$ represent the predicted value of y_i when observation i is in the test fold for replicate $r = 1, \dots, 10$. The cross-validation average accuracy for one color ($color \in \{\text{cyan, magenta, yellow}\}$) is computed as

$$Acc_{color} = \frac{1}{10 \cdot 77} \sum_{r=1}^{10} \sum_{i=1}^{77} I[y_i = \hat{y}_{i,r}]. \quad (1)$$

There are a handful of values that must be specified when implementing the VEESA pipeline. We choose to vary the number of times the box-filter is run for smoothing (0, 5, 10, 15, 20, 25, 30, and 35 times), the number of PCs input to the model (10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 PCs), and the number of trees in the random forest (50, 100, 250, 500, and 1000 trees). The cross-validation process is applied 400 times for each color (once for all combinations of box-filter runs, input PCs, and random forest trees). The values considered are selected to show a range over which model accuracy increases and diminishes. The random

forests are fit using the *randomForest* R package with all tuning parameters set to the default value besides for the number of trees.

The cross-validation average accuracies are shown in Figure S15. The x-axis shows the number of PCs input to the model, and the y-axis depicts the cross-validation accuracy. The rows and columns separate the results by color and the number of random forest trees, respectively. The color of a line represents the number of times the box-filter is run, and the horizontal dashed black line represents the best cross-validation accuracy for a color obtained by Buzzini et al. (2021). There are some clear trends in the cross-validation results. For example, regardless of the number of random forest trees, the average accuracies tend to increase as the number of PCs increase to around 30-50 and then decrease as the number of PCs increase further. We also see that for all PCs, number of trees, and color, the more times the box-filter is run, the average accuracy tends to increase until approximately 25 times when the improvement is either minimal or the accuracy begins to decrease. The grey triangles pointing up and down in Figure S15 indicate the VEESA pipeline scenarios with the highest and lowest average cross-validation accuracy, respectively, within a color.

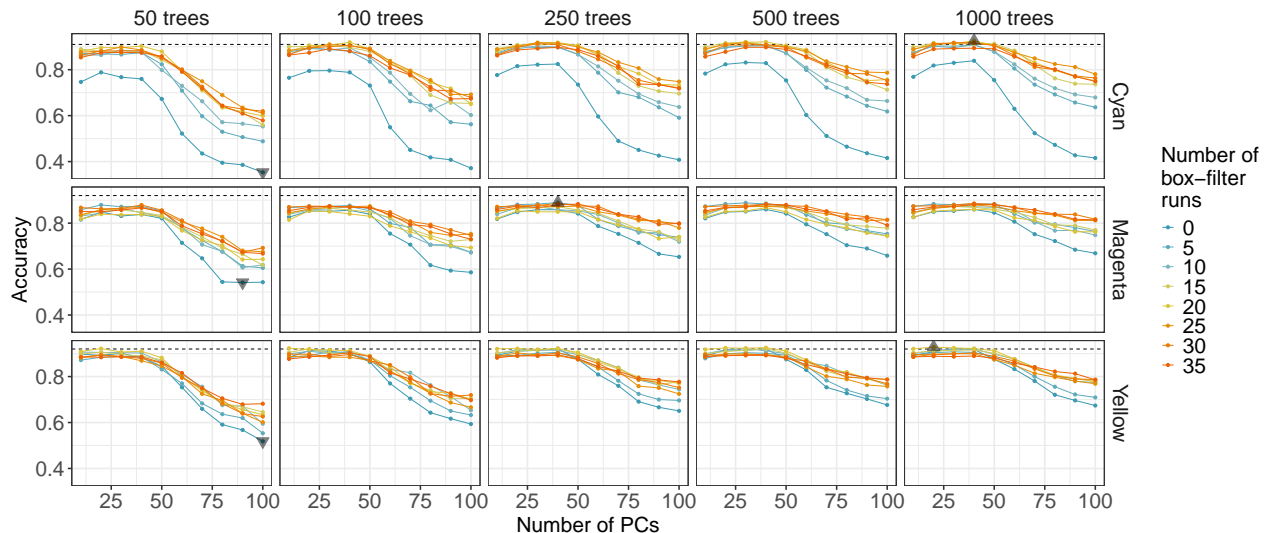


Figure S15: Inkjet Data Cross-Validation Average Accuracies. Triangles pointing up and down highlight the highest and lowest average cross-validation accuracies from the VEESA pipeline, respectively, for each color. Horizontal dashed lines represent best average cross-validation accuracies for each color from Buzzini et al. (2021).

3.2 Printer Classification Details

Figure S16 depicts confusion matrices for the three colors considered in the inkjet printer analysis in the main text. The true printer is listed on the x-axis, and the predicted printer is listed on the y-axis. The plots are generated using all test fold predictions for the best performing model for each color. These figures let us see which printers are challenging. For all colors, printers 2 and 3 and 7 and 8 often are predicted as the other printer. Printers 2 and 3 share the same manufacturer, and printers 7 and 8 have the same manufacturer and model. It is understandable why these printers are more challenging. The confusion between printers 7 and 8 is particularly true for magenta. Additionally, printer 11 is frequently mistaken for printer 10 with magenta. Again, these printers share a manufacturer. These results suggest that the models tend to struggle to distinguish printers that are from the same manufacturer, which would be expected.

3.3 Additional Principal Directions

Figures S17 through S21 show the principal direction (top) and principal difference (bottom) plots for the five most important PCs from the worst performing cyan model and best and worst performing magenta and yellow models, respectively.

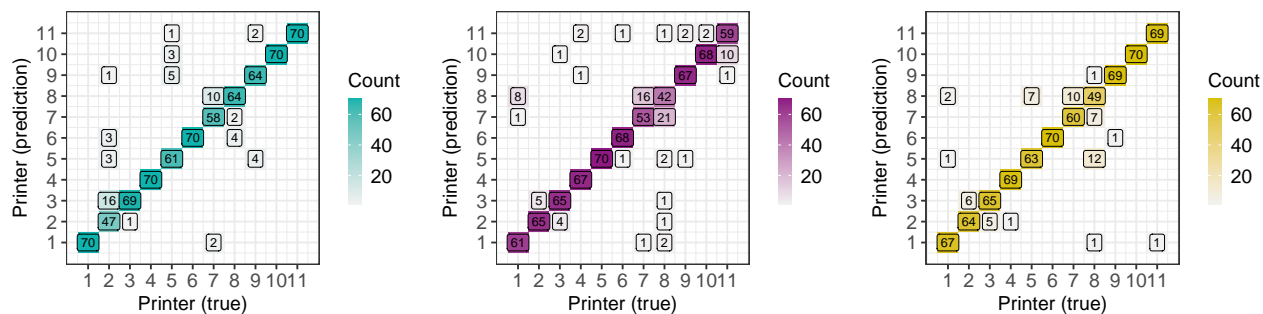


Figure S16: Confusion matrices for the inkjet printer random forests.

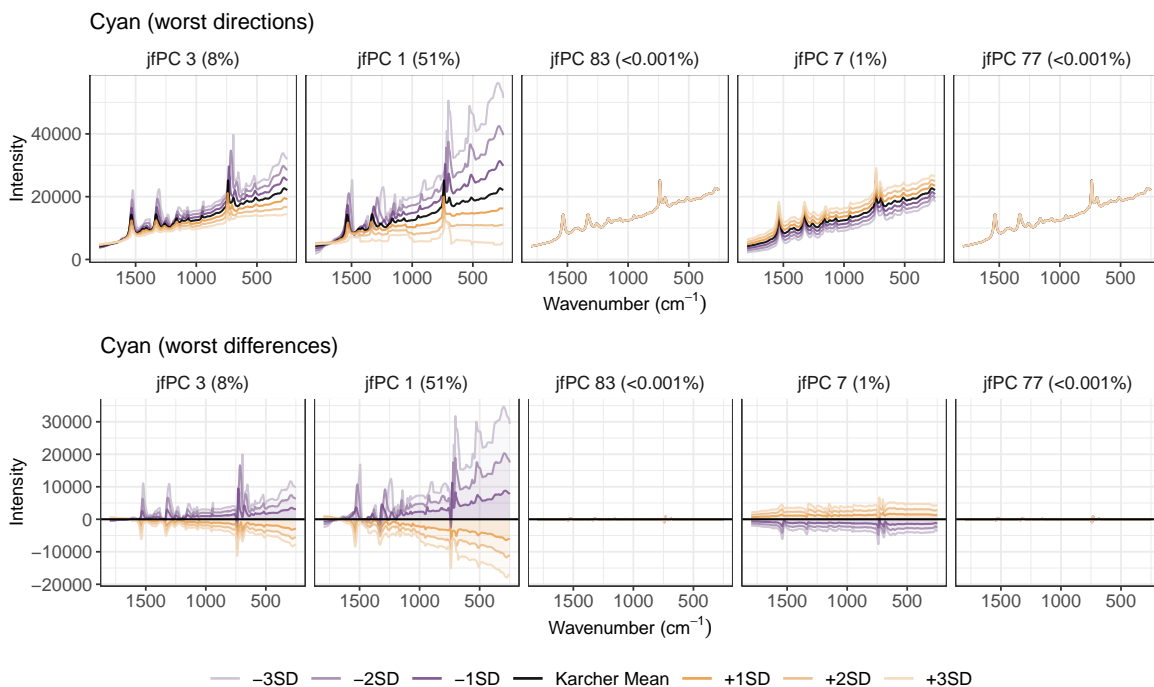


Figure S17: The principal direction and principal differences plots for the five most important jfPCs from the worst model for cyan inkjet signatures.

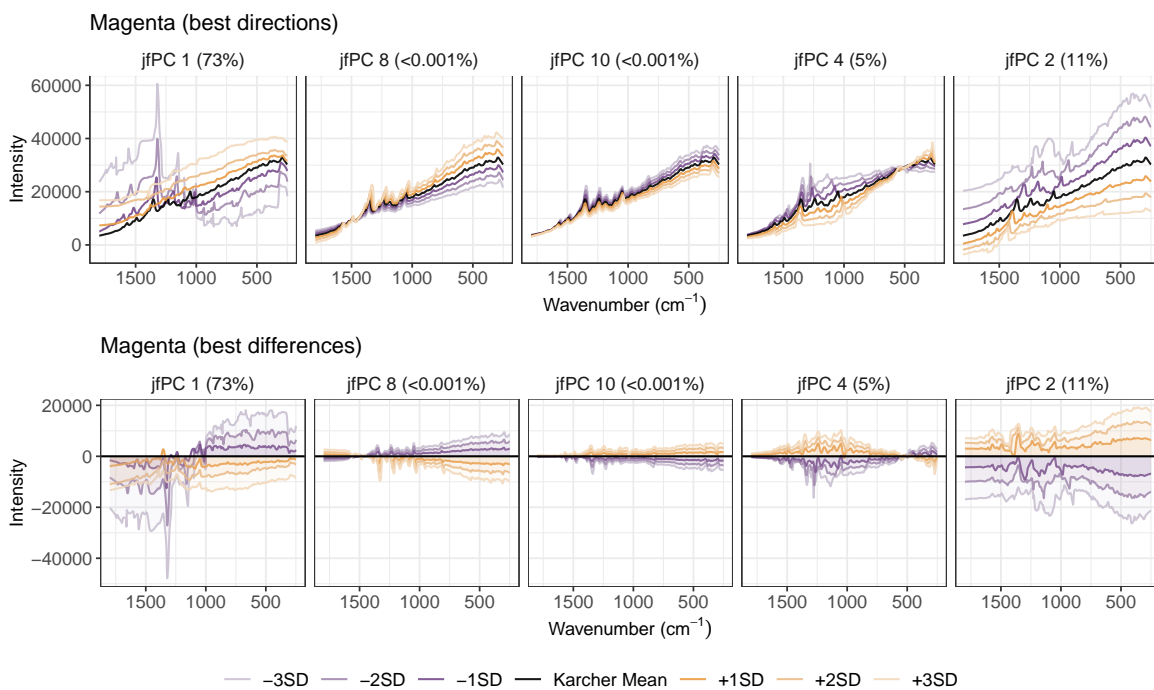


Figure S18: The principal direction and principal differences plots for the five most important jfPCs from the best model for magenta inkjet signatures.

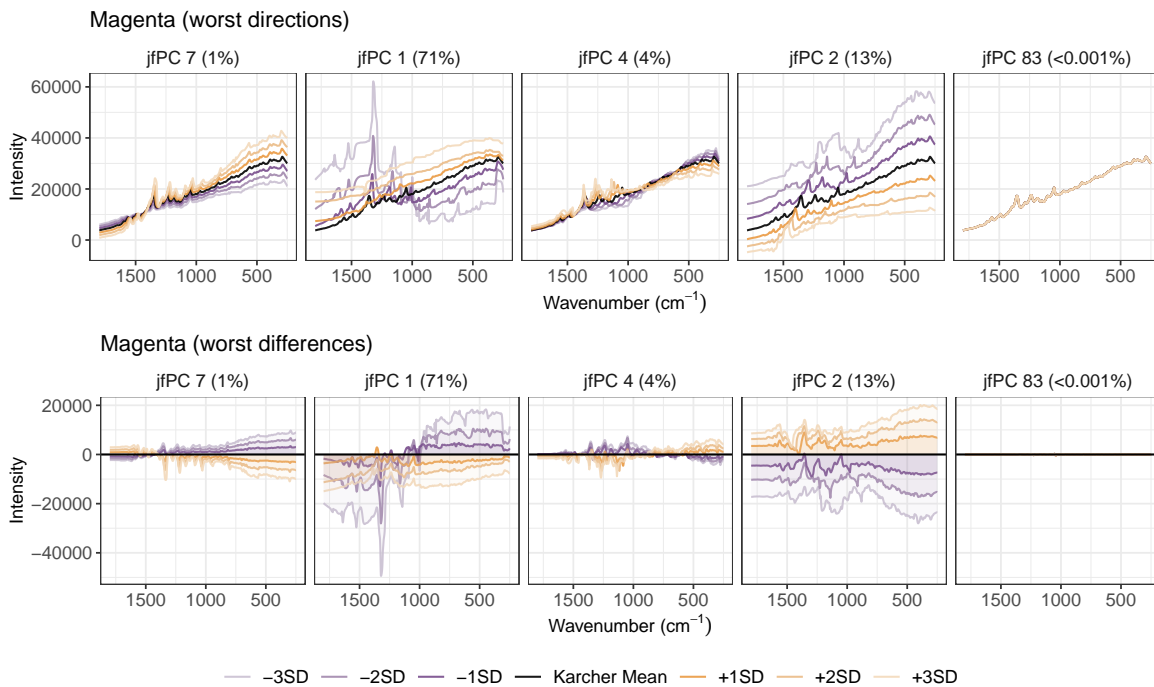


Figure S19: The principal direction and principal differences plots for the five most important jfPCs from the worst model for magenta inkjet signatures.

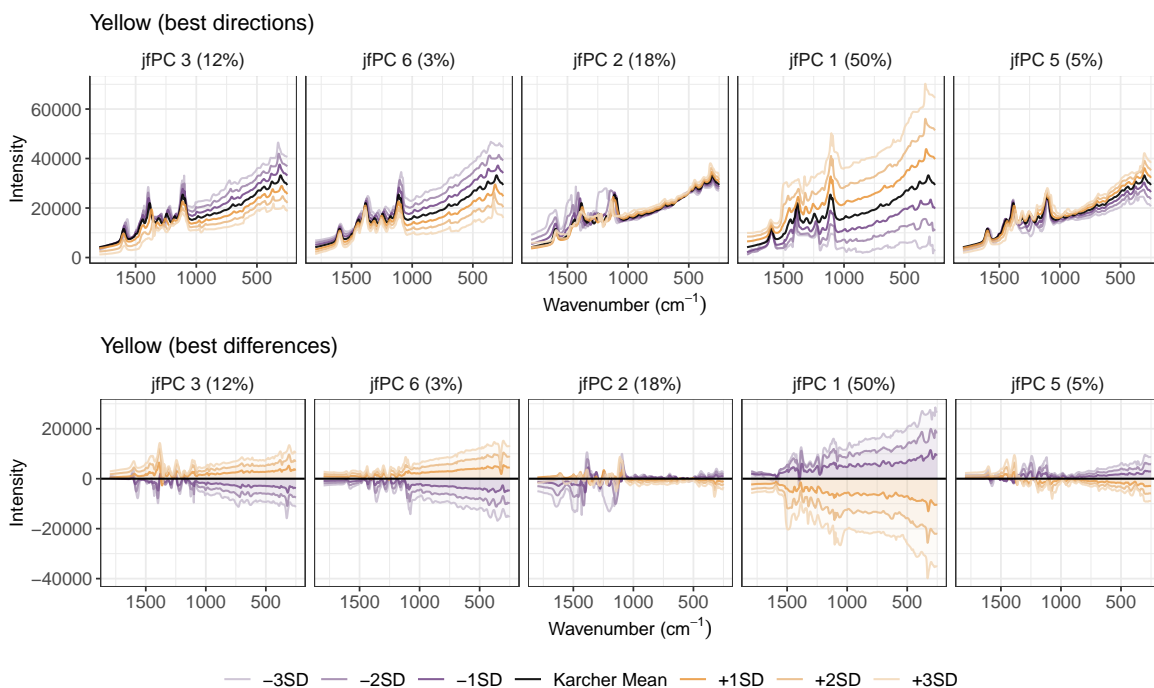


Figure S20: The principal direction and principal differences plots for the five most important jfPCs from the best model for yellow inkjet signatures.

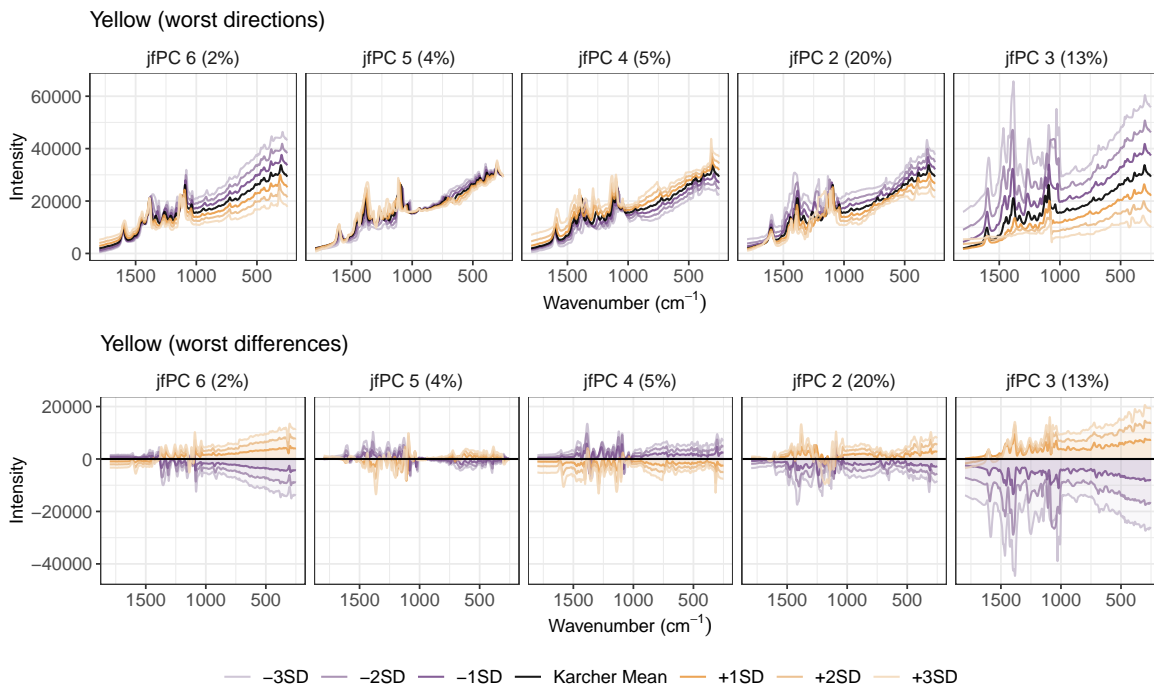


Figure S21: The principal direction and principal differences plots for the five most important jfPCs from the worst model for yellow inkjet signatures.

3.4 Improving Models Using PFI

As a last step in our analysis, we are interested in determining if we can make use of the information from the PFI results of the worst models to improve the predictive performance. We do this by implementing the cross-validation procedure described in Section 3.1. However, for this implementation, we only consider random forests with 500 trees. We choose 500 trees, because it falls between the number of trees associated with the best performing models (i.e., 250 or 1000 trees). Additionally, we use PCs 1 to 25 and PCs 75 to 100 for all models based on the areas with important PCs as seen in Figure 10. We consider the same range of smoothing iterations (0, 5, 10, 15, 20, 25, 30, 35), and again, random forests are implemented separately by color.

Figure S22 shows the average cross-validation accuracies from this analysis separated by color. The number of smoothing iterations is depicted on the x-axis, and the average accuracy is on the y-axis. The colored lines represent the accuracies from our original analysis (Figure S15), and the color of the lines indicates the number of PCs included in the model. The black line represents the average cross-validation accuracy from the updated models with the subset of fPCs selected based on the PFI results. For all smoothing iterations, the down selection of fPCs based on PFI led to a clear increase in test fold average accuracy. The accuracies are still lower than the best performing models, but this provides a case study where PFI provides useful information for improving model predictive performance.

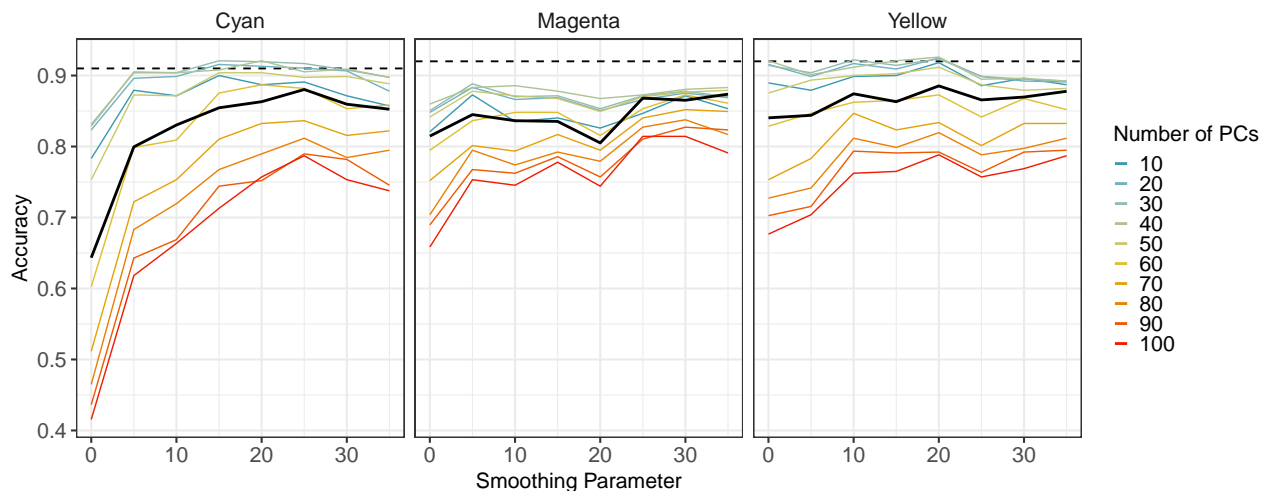


Figure S22: Cross-validation average accuracies. Black lines represent CV accuracies from models with PCs selected via PFI.