Original research paper

JNIRS

Journal of Near Infrared Spectroscopy
2018, Vol. 26(1) 61–75

DOI: 10.1177/0967033518756175
journals.sagepub.com/home/jns
SAGE

Methods for performing dimensionality reduction in hyperspectral image classification

Jun-Li Xu, Carlos Esquerre and Da-Wen Sun

Abstract

This paper provides several useful strategies for performing the dimensionality reduction in hyperspectral imaging data, with detailed command line scripts in the Matlab computing language as the supplementary data. Due to the vast number of data dimensionality reduction methods available, this paper will mainly focus on some commonly used approaches adopted in hyperspectral imaging. In this work, transformation-based methods include principal component analysis and linear discriminant analysis, while band selection methods are comprised of partial least squares regression combined with the variable importance in the projection scores, selectivity ratio, and significance multivariate correlation; Monte Carlo sampling-based methods including enhanced Monte Carlo variable selection and competitive adaptive reweighted sampling; model population analysis-based methods from libPLS including uninformative variable elimination, random frog, and PHADIA; Matlab built-in functions for feature selection including Relieff, stepwise regression, and sequential feature selection; and the selection method guided by genetic algorithm. The example data included in supplementary material, also available for download, will be used to simplify decision tree models for differentiation of white stripe and red muscle pixels on salmon fillets, since classification is one of the main application domains of hyperspectral imaging. In this work, there are many original codes and functions developed, such as fast multiple scattering correction preprocessing, outlier detection, optimal cutoff value determination, spikes, and dead spectra identification and correction for hyperspectral image. More importantly, a further selection function based on variance inflation factor is proposed to diagnose and alleviate collinearity problem because collinearity and multicollinearity are always expected to be severe in the spectral data. In this work, step-by-step procedure is provided for easy adaptation of these strategies to individual case.

Keywords

Hyperspectral imaging, dimensionality reduction, decision tree, partial least squares, variance inflation factor, near infrared

Received 30 January 2017; accepted 20 December 2017

Introduction

By integrating spectroscopy and imaging or computer vision[1–10] into one system, hyperspectral imaging technique has emerged as a potential and powerful analytical tool for nondestructive analysis, enabling online control of spatially resolved spectral information of materials.[11] Nowadays, hyperspectral imaging technique has been widely studied for evaluating food quality and safety.[12–27] However, the massive size of hyperspectral image data often limits its adoption in the real word.[28] One of the greatest limitations arises from the relatively lengthy time necessary for hyperspectral image acquisition, image and spectra processing, model development, and model application. To address this problem, a great variety of techniques have been proposed to reduce the hyperspectral data dimensionality. Additionally, a hyperspectral image always has very strong spectral correlation, meaning that there is a problem of collinearity (or multicollinearity) among contiguous wavelength variables.[29] Severe collinearity between predictors in a model will cause singularity of the matrix, and this in turn will impact the coefficients calculated.[30] One effective solution to minimize the collinearity among contiguous wavelengths and avoid singularity is to reduce data dimensionality.[5] Hence, a reduced image cube can be yielded to take the place of the whole hypercube with massive data, and this alteration will remarkably save the subsequent data processing time and will probably enhance model performance in terms of accuracy and robustness.

Food Refrigeration and Computerized Food Technology (FRCFT), School of Biosystems and Food Engineering, University College Dublin, National University of Ireland, Agriculture and Food Science Centre, Belfield, Dublin 4, Ireland

Corresponding author:
Da-Wen Sun, Food Refrigeration and Computerized Food Technology (FRCFT), School of Biosystems and Food Engineering, University College Dublin, National University of Ireland, Agriculture and Food Science Centre, Belfield, Dublin 4, Ireland.
Email: dawen.sun@ucd.ie
Websites: http://www.ucd.ie/refrig, http://www.ucd.ie/sun

There are mainly two types of dimensionality reduction techniques in hyperspectral imaging analysis: a transformation-based or a selection-based method.[6] The main difference relies on whether a dimensionality reduction method transforms or retains the meaning of the original data set. Transformation-based reduction approach, also known as feature extraction, is defined because it irreversibly transforms the original data set features.[6] It is generally implemented in the situations where the original data sets are not required by further process. Selection based, which is also acknowledged as feature selection or band selection, manages to preserve the meaning of the original data set. The selection-based dimensionality reduction technique aims to find a minimal subset of optimal wavelengths containing important information related to certain attributes, and to yield the smallest possible errors no matter it is quantitative determination or qualitative discrimination. So far, there have been a great number of band selection methods applied in multivariate analysis of hyperspectral imaging.

This work is aimed at providing a step-by-step instruction to perform different dimensionality reduction techniques on a hyperspectral imaging data set, with corresponding command line scripts written in Matlab computing language (http://www.mathworks.com). It should be noted that the main goal is to demonstrate how to perform dimensionality reduction methods on imaging dataset and how to compare their results and choose the most suitable variable subsets. The example data in this work are employed to develop reduced models to differentiate between white stripe and red muscle pixels on salmon fillets by using near infrared hyperspectral imaging. Acknowledging that discussing all treatment options, including image acquisition, background removal, dead pixels and spikes identification, and masking, could be too long, here the major issue will be focused on the application of different dimensionality reduction techniques. Starting from spectra extracted from salmon samples to represent either white stripe or red muscle, the typical steps encountered in hyperspectral data compression are then presented, followed by the establishment of the simplified models and prediction maps. It is hoped that this paper will enlighten better understanding of different dimensionality reduction techniques for hyperspectral imaging data and encourage the adaptation of some useful strategies to solve individual hyperspectral imaging problem.

Data set and data partition

In this work, hyperspectral images of farm-raised Atlantic salmon (*Salmon salar*) fillets were chosen as examples to perform dimension reduction in the case of a classification issue. The details about the sample can be found elsewhere.[7] The Matlab scripts used to carry out the analysis are provided as a supplementary material and it is freely available to be downloaded from: https://bitbucket.org/lily-xu. The demo commands for applying all steps in the analysis are given in the script called *VarSel_HSI.m*. It is significant to notice that all scripts are written sequentially following the structure of this work. Hence, every command line actually depends upon previous lines being executed. It is also important to note that the successful running of this work requires the Statistics and Machine Learning Toolbox installed in Matlab software.

Salmon is valued as a fat-rich fish with a large proportion of lipids congregated in white stripes of connective tissue (myocommata), segmenting the red-colored muscle tissue in vertical blocks and presenting a zebra-like appearance.[8] In a previous study Segtnan et al.[9] have demonstrated that the proportion of myocommata in a salmon fillet correlated well with its fat content. Stien et al.[10] established a method for the rapid and automatic measurement of fat content in salmon fillets based on a computer vision system, and revealed that fat content could be estimated based on image analysis to calculate the area of the white stripes visible on the surface compared to the total area of the fillet. More recently, Xu et al.[11] attempted to apply particle analysis on the images to extract only the spectra from white stripe in salmon fillets to assess lipid oxidation. Overall, it is interesting to classify white stripe from the red muscle because the proportions of white stripe in one fillet might contain some valuable information about the fat content and/or lipid oxidation.

A data structure that allows saving of multiple variables within a single variable is used to manage the data in this work. The resulting data structure array with the name of *data.mat* is comprised of five fields and one field called *spec* stores 650 spectra of white stripe and 650 spectra of red muscle extracted from 13 salmon fillet images. It should be pointed out that each spectrum was extracted from a single pixel and every 100 spectra (50 red muscle and 50 white stripe) were extracted from the same salmon fillet. It is well acknowledged that all calibration models are supposed to be validated with independent datasets.[12] In our case, approximately two-thirds of all samples were randomly selected and formed a calibration set with 900 pixels, while the remaining 400 samples were collected as a validation set. Meanwhile, the corresponding response matrix Y is created with ''1'' denotes white stripe and ''2'' represents red muscle category. Moreover, three hyperspectral images of salmon fillets with background removed are stored in the field *im* within the structure and one field named as *d* will be used to place preprocessed spectra. In addition to this, an external data set that was obtained from another nine salmon fillets was used for assessment of the generalization error of the final chosen model. This test set is comprised of 450 spectra of white stripe and 450 spectra of red muscle with every 100 spectra (50 red muscle and 50 white stripe) from the same piece of fillet. The test data set is stored as x*test* and *ytest* in the structure array. Matlab code related with details

about spectra collection and data partition can be found in the online Appendix.

Spectral pretreatment

Spectral preprocessing techniques are commonly implemented in hyperspectral imaging and spectroscopy to reduce many undesired effects such as scattering and baseline shift.[13] In this work, a new function with the name of *msc.m* has been developed to conduct spectral preprocessing. This function is pretty straightforward and easy to implement. One of the biggest advantages is this function can save massive computation runtimes. In Matlab, one of the slowest blocks of code that inflates runtimes is for/while loop. Thereby, one of the most amazingly underrated and unfamiliar functions in the Matlab's repertoire: *bsxfun* was adopted as the alternative to take place of the for-loop. *Bsxfun*, introduced since 2007, has made matrix–vector operation easier when the vector is going to be applied to each row. There are two important reasons why *bsxfun* computes much faster: (1) the calculation happens in compiled code, indicating that the actual replication of the array never happens and (2) *bsxfun* is one of the multithreaded Matlab functions. Conversion of loops to *bsxfun* will streamline the computation process, which opens the door for improved optimization of the algorithm and less chances for bugs in the long term. In our case, the multiple scattering correction (MSC) pretreatments for calibration and validation sets were accomplished from lines 26–28.

Correlation analysis

Column-wise preprocessing is then implemented first to have the mean value subtracted from each column and then divided by the standard deviation. The process of division by the standard deviation accounts for more equal weighting of each individual band within the correlation coefficient map. In the present case, correlation analysis is first conducted to investigate the relationship between each pair of spectral variables. Pearson's linear correlation coefficient between each pair of columns is computed in the line 67 and the corresponding result is shown as a form of figure (lines 70–78 of demo script). A correlation coefficient refers to a number that quantifies some type of correlation and dependence between two variables. Its value falls in the range between -1 and 1, where 1 stands for a perfect positive correlation or similarity, -1 indicates total negative correlation, and 0 is no correlation or no similarity.[14]

Figure 1 shows the wavelength–wavelength correlation coefficient map of 900 pixel spectra in the calibration set. The main diagonal of the resultant matrix corresponds to the correlation between each random variable to itself (all equal to 1). The off-diagonals refer to sample correlation coefficients between each pair of spectral variables. Since correlation coefficient between the *ith* and the *jth* variable is the same as between the *jth* and the *ith* variable, the obtained matrix is therefore symmetric. As can be seen, high correlations including positive and negative correlation are observed from this map, indicating the potential existence of collinearity. Likewise, the relationship between wavelength variables and response variable is calculated using the same function and the wavelengths with highest correlation coefficient in positive and negative are determined (lines 82–84), respectively. It is obtained that wavelength at 1483 nm presents the highest positive correlation, while 1322 nm indicates the highest negative correlation.

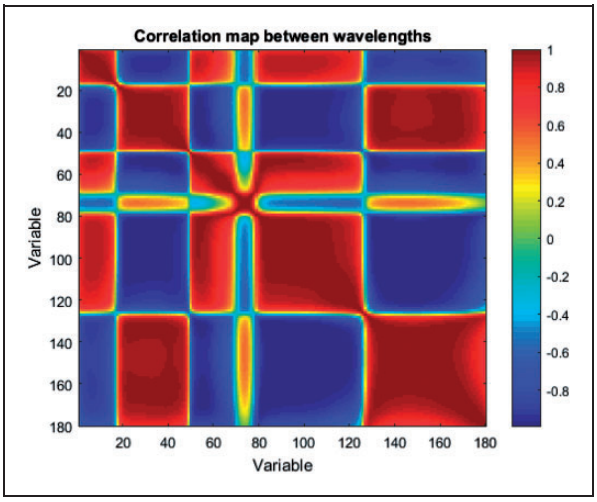

Figure 1. Wavelength–wavelength correlation coefficient map.

Transformation-based data reduction

Principal component analysis (PCA)

PCA is the most popular dimension reduction method used in hyperspectral imaging.[15] Based on orthogonal transformation, it aims to convert a great number of possibly correlated variables into a set of linearly uncorrelated variables called principal components (PCs). PCA is usually performed on two-dimensional matrices. In our case, MSC preprocessed spectral data have to be treated by mean centering before PCA. PCA can be conducted by a simple function called *mypca* in the supplementary material (lines 123–124). PCA transformation could guarantee that the first PC has the largest possible variance and each succeeding PC interprets progressively less of the variance. Therefore, the first few PCs can be selected instead of using all PCs to reach the goal of dimension reduction. Moreover, the first few components are believed to have higher signal-to-noise ratio and the later PCs are likely to be dominated by noise. Therefore, with a wise choice, using PCA transformation can remove the noise from a data set. In order to remove outliers, a simple function *outlier* based on the principle of Euclidean distance is executed on the PCA score space (lines 134–137). This outlier function first computes the distance from

one pixel to the center of the remaining pixels and then the maximum distance can be calculated. A criterion, e.g., the maximum distance falls within 2 standard deviations of the mean, can be used. Pixel with the maximum distance failed to meet this criteria can be excluded and the same procedure is repeated until all pixels meet this criterion. A new PCA model is built based on the data without outliers and its result is shown in Figure 2 (lines 181–186). According to PC 1–PC 2 score plot, it can be concluded that white stripe and red muscle can potentially be differentiated into each other and PC 1 is observed to have the most contribution in classification. Looking at the corresponding loadings (Figure 2), PC 1 grades the two classes mostly relies on the influence of a negative peak centered at 1300 nm together with the other two peaks in its positive part. As indicated, pixel with higher value at 1300 nm is more likely to be white stripe. Interestingly, pixels of red muscles form two blue clusters, probably due to the compositional difference of red muscle between different fillets.

Linear discriminant analysis (LDA)

It is widely believed that the general LDA approach is quite similar to PCA because they both search for linear combinations of variables which best interpret the tested data. The difference relies on the fact that PCA seeks for finding the PCs that maximize the variance of data, but LDA attempts to maximize the separation between different classes.[16] Unlike PCA, LDA is well acknowledged as supervised algorithm which utilizes the label information to find informative projections. In brief, LDA aims to project a feature space with multidimensionality onto a smaller subspace but at the same time retain the class-discriminatory information.[17] In this work, the function named as *myLDA* is developed to maximize the ratio of between-class distance to within-class distance, thereby achieving optimal separation (line 191). In this case, the first eigenvector has accounted for the largest eigenvalues (lines 195–209). Thus, this 180×1 eigenvector matrix can be subsequently used to transform the original matrix onto the new subspace.

Band selection methods

PLS-based variable selection

Variable importance in the projection (VIP) score under PLS[18] has been paid increasing attention these days because it performs very well in the near infrared spectroscopy as well as hyperspectral imaging. The VIP score for the *jth* variable can be computed as follows:

$$VIP_j = \sqrt{\frac{\sum_{f=1}^{F} w_{jf}^2 \times SSY_f \times J}{SSY_{total} \times F}} \quad (1)$$

where w_{jf} is weight of the *jth* predictor variable in component *f*; SSY_f refers to the fraction of variance in response variable explained by *fth* component; J simply means the number of predictor variables; SSY_{total} is the total variance of the response variable and F is the total number of components. Generally speaking, the variable with higher value of VIP score indicates more significance and relevance to predict the response variable. Meanwhile, there is a "greater than one rule" widely accepted as a rule of thumb criterion for variable selection because the average of squared VIP scores is equal to 1. However, more and more people suggest that it is not wise to simply include variables greater than 1.[19]

Similarly, the selectivity ratio (SR) is another measurement for variable importance. Specifically, it refers to the ratio between the explained and residual variance of the predictor variables on the target-projected component. It reflects the ability of a predictor variable for the discrimination of different classes in terms of a classification problem. A high SR value normally means a variable with good predictive performance. In this work, the critical threshold value for a boundary between explanatory variables with high discriminating

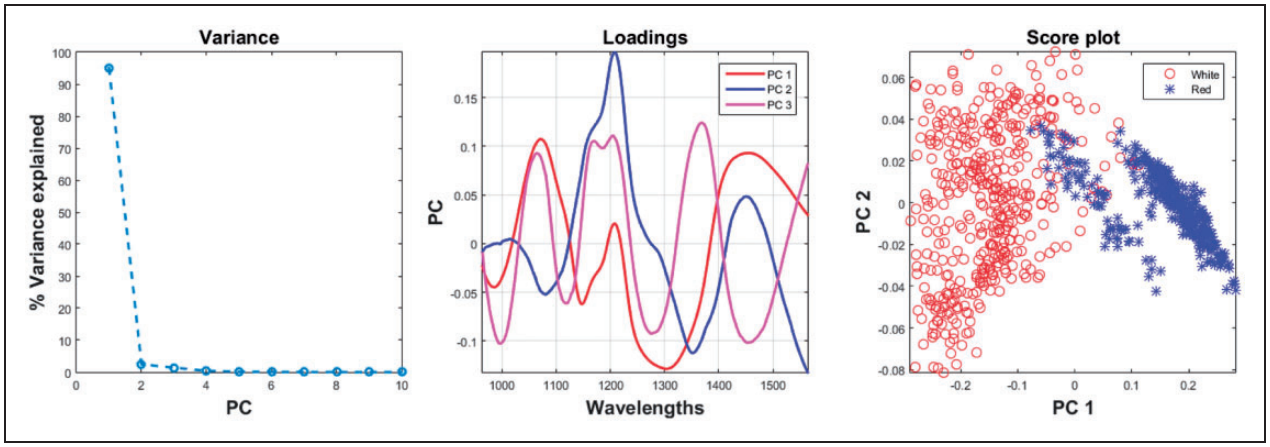

Figure 2. PCA model performance after outlier removal.

ability and less interesting regions is defined as suggested by Rajalahti et al.[20] In detail, it is computed using F-test with n-2 and n-3 degrees of freedom. At the same time, significance multivariate correlation (sMC), proposed by Tran et al.,[21] is carried out in order to minimize the effect of irrelevant predictor variables and highlight the variables most correlated to the response. The goal of sMC is to estimate for each predictor variable the correct sources of variance calculated from the PLS regression and apply them to determine the variable importance. Like SR, F-test is also executed to identify variables that are statistically significant concerning with their correlation to the response. But unlike SR, an F-distribution of $F(1-\alpha, 1, n\text{-}2)$ is used, wherein α is the chosen significance level with 1 numerator and an n-2 denominator degrees of freedom.

In this study, the function named as *pls_selection* is proposed to simultaneously obtain VIP, SR, and sMC indexes (lines 217–219). Their corresponding selected variables are stored in the structure with the name of *Var_Sel*. As we see, the number of chosen spectral variables for VIP, SR, and sMC is 78, 156, and 95, respectively. It is not difficult to detect that the selected variables are mostly in the same spectral neighbor regions, with high correlation among each other. Thus, redundant information is also included in the selected variable subsets.

Monte Carlo sampling (MCS) methods

Ensemble Monte Carlo variable selection. Ensemble Monte Carlo variable selection (Emcvs), suggested by Esquerre et al.,[22] is also adopted to remove uninformative wavelengths from the data set. The principle behind this method is to compute a normalized regression coefficient in terms of different sample subsets and 100 repeated running times. Normally, 10% of samples are firstly randomly selected from the calibration matrix and the corresponding regression vector is calculated from a PLS model. When repeated for 100 times, a normalized regression coefficient that is the ratio of the standard deviation to the mean is obtained. The whole procedure is repeated hundreds of times to compute the mean of the standardized regression coefficient for each spectral variable. According to the absolute values of coefficient, all the variables are ranked from largest to lowest and the coefficient of N_j^{th} variable is set as the cutoff value, meaning that only variables larger than the cutoff value can be retained. In the present situation, a function called *cutoff* was developed to determine the optimal cutoff value on the basis of a decision tree (DT) classifier which was also used as the main supervised classification algorithm in this work. Assuming that 20% of the whole spectral variables are sufficient to achieve a satisfying classification performance, 36 is used as the maximum number of selected variables. It is critical to feed this function the ranking sequence of variables, calibration set, and its corresponding response labels. This function attempts to seek for the optimal cutoff value on the basis of a venetian blinds cross-validation in terms of misclassification rate. Generally speaking, before reaching optimal N_j^{th}, the robustness and accuracy of the model will be poor, due to insufficient informative variables included. On the contrary, when the number of retained variables is larger than N_j, uninformative variables could be embodied, accounting for relatively bad performance. In this work, the best number is investigated with step of 3 from 3 to 36. For each interval, error of cross-validation is computed and the optimal N_j^{th} is determined with the minimum error. If the obtained optimal number is 36, the maximum number of selected variables should be increased. Emcvs is conducted by running line 224 with some parameters presented. It is important to note that the parameters are free to be changed for obtaining different selection results.

Competitive adaptive reweighted sampling. Competitive adaptive reweighted sampling (CARS) coupled with PLSDA was developed in light of a simple but effective principle "survival of the fittest" based on Darwin's Evolution Theory.[23] It has great potential to collect an optimal combination of the wavelengths among the full spectrum. In this work, CARS is applied via libPLS library[24] which can be freely downloaded online at www.libpls.net. In order to provide some convenience and save time for searching, libPLS library source was also included without any modifications in this material. If using any script within the libPLS library folder, it is important to cite the original source. It is very important to notice that the response variables that are suitable for a libPLS application have to contain only "1" and "−1". Therefore, lines from 229 to 231 are presented to alter all the "2" class to "−1". One important parameter in CARS is the number of MCS runs. In each sampling run, exponentially decreasing function is employed to eliminate wavelengths of little or no information in a stepwise and efficient manner. Following that, adaptive reweighted sampling is utilized to realize a competitive further selection of wavelengths. To evaluate the subset, cross-validation is adopted in the PLS-DA model. Generally, the cross-validation error value will first decrease and then change back and forth until it reaches the minimum value, and finally increases rapidly. The first phase results from the removal of interfering and irrelevant variables. After the fluctuation stage, it then rapidly increases and this can be ascribed to the loss of important information due to the elimination of some informative variables from the optimal subset. The best subset is identified with the lowest cross-validation error. In this setting, $N = 500$ sampling runs are determined and the rest parameter values can be witnessed in line 232.

Model population analysis (MPA) methods

MPA was proposed by Li et al.[25] as a general framework to design new forms of chemometrics and

bioinformatics algorithms. The essence of MPA is to develop data-analysis method by statistically analyzing the distribution of a large population of sub-models derived with the aid of MCS.

Uninformative variable elimination (UVE) method. In virtue of the MCS runs and the stability criterion in UVE algorithm, an integrated method for variable selection is proposed with the name of MC-UVE.[26] Following the principle of the Monte Carlo method, the whole procedure is started with the random selection of calibration set samples with a certain ratio. A matrix of the regression coefficients is calculated from all the built PLS models. The reliability index (RI) to evaluate the stability of each variable is defined as the ratio of the mean and standard deviation of the regression coefficients. The RI value is able to assess variable importance, and therefore larger values usually correspond to more importance. In this case, MC-UVE in conjunction with *cutoff* function is conducted by running lines from 245 to 247.

Random frog method. Random frog (RF) is a recently proposed approach to borrow the merits of reversible jump Monte Carlo Markov chain techniques.[27] On the basis of MPA, it manages to develop a mathematically simple and computationally efficient approach for variable selection. When coupled with PLS-DA, RF will generate a selection probability to measure variable relevance and importance, and this can be used as the selection criterion. There are two main steps involved in the RF technique: the model search step and the model ensemble step. The first step begins with random generation of an initial subset V_0 with Q variables. Then a candidate variable subset V^* containing Q^* variables is proposed and a probability-based rule can be applied to determine whether V^* can be accepted. If the answer is positive, V_1 will be set to V^*, otherwise, V_1 will be assigned the value of V_0. This procedure will repeat N times before it steps into the second phase. The frequency of occurrence for each variable in all these N subsets will be finally computed as a criterion to assess variable importance. There are some tuning parameters controlling the performance of RF, among which the number of iterations N is a very important factor. In the present case, N is set to 2000 because it is sufficiently large to reach convergence for this data set. Analogously, the function *cutoff* is then used to define the optimal cutoff value and reserve the first N_j variables (lines 251–253).

PHADIA method. Like the RF method, PHADIA is also designed on the basis of MPA.[28] It aims to output a phase diagram presenting the predictive ability of each variable, which proves to be an intuitive manner for selecting important variables. MCS is used to form a sub-data set with Q out of full predictor variables randomly selected at each run. For each sub-data set, a PLS-LDA model is built and assessed with cross-validation. All the sub-models then can be partitioned into two groups, one contains all the sub-models including a certain variable, say *ith* variable, and the other group includes the remaining sub-models. Then, the mean and standard deviation of prediction errors can hence be easily calculated from these two groups. DMEAN is defined by the difference between mean prediction error of the group with models including the *ith* variable and that of the other group. According to the definition, DMEAN has the ability to measure the increment of prediction performances of model adding the *ith* variable over models without it. In other words, when DMEAN is a positive value, it can be indicated that the predictive ability could be improved if a model contains the *ith* variable, and vice versa. Additionally, the nonparametric Mann–Whitney U test is utilized to computer a p value to judge whether the mean prediction errors of these two groups are different. Hence, DMEAN in combination with p value shall provide important information about whether a variable can significantly promote prediction performance or not. As suggested, it was recognized that the variables with positive DMEAN value and p value less than 0.01 as the informative predictor variables (lines 257–258).

Relieff variable selection

Relieff is reported to be a very powerful feature selection algorithm used in binary classification since it first developed by Kira and Rendell.[29] It was further extended to the algorithm (Relieff) by Kononenko et al.[30] so as to enhance the stability of the probability approximation and make it applicable to deal with incomplete data and multi-class problem. This built in function for Matlab can compute both ranks and weights of predictor variables. The input requires the data matrix X with rows for each observation and columns for each attribute, response vector Y and parameter K. In the case of classification, Relieff uses K nearest neighbors per class. It is critical to set the right K value, because small K values tend to be unreliable for noisy data and large K values, which are comparable with the number of observations are likely to fail to find important variables. To find the optimal K value, it is necessary to try different K nearest neighbors for the algorithm, for example, from 3 to 180 with an interval of 3. The ranks and weights computed for these vary wildly initially and then stabilize as K approaches optimal value. In our case, $K = 60$ is determined after investigating the stability of the resultant ranks and weights for various values of K (lines 263–264). After ranking predictors by importance, the cutoff threshold can be obtained as introduced previously.

Stepwise regression selection

Meanwhile, stepwise regression is another versatile technique in which the choice of predictive variables

is carried out by adding and deleting terms from a multilinear model based on certain statistical selection criterion. This algorithm proceeds in three steps. First of all, it fits an initial model and compares the explanatory power of incrementally larger and smaller models. It is worthwhile to notice that the p value of an F-statistic is calculated at each step to test models with and without a potential term. The second step aims to add a term if this variable is not currently in the model but have the smallest p value among any other out of model variables having p values less than an entrance tolerance. In the third step, among any terms in the model having p values larger than an exit tolerance, the one with highest p value will be removed and the algorithm reverses back to step 2. The *stepwisefit* function worked in Maltab is used with default setting in our work by executing command line 268.

Sequential feature selection

Sequential forward selection (SFS) algorithm carries out a bottom-up search strategy which initiates from an empty feature set and sequentially adding each of the features not yet selected while evaluating the criterion. At each iteration, the sequential search algorithm will add features to the feature set and it is expected that the new extended feature set should yield a minimum classification error compared with the addition of any other feature. Because of its simplicity and speed, SFS is widely used for the application of variable selection.[31] For the sake of simplicity, a function named as *SFS* is developed in this work based on the *sequentialfs* function. The selected variables can be obtained after running the command line of 273.

Genetic algorithm guided selection method

In addition to the aforementioned approaches, feature selection can also be applied based on combinatorial optimization by means of genetic algorithm (GA). This method was first proposed to avoid some limitations of the existing methods by Ludwig and Nunes.[32] In order to consider redundancy among input variables, this method aims to minimize the joint conditional entropy between the input and output variables by using GA. According to Ludwig and Nunes,[32] joint conditional entropy enables the evaluation of the amount of information that is lacking to determine the target output variable. Detailed information about GA can be found in Ludwig and Nunes.[32] In this method, GA was operated on the decimal basis. The script is accessible online from MathWorks file exchange. It was also included in the corresponding material with no modification. Thereby, it is kindly to remind that it is significant to cite the original source if this method is used. The algorithm requires three arguments: the desired number of selected features, a matrix where each column is an observation and each row is a feature, and its respective target data arranged in a row vector. The desired output is a vector containing the selected feature indexes, yet the order of features is not related with their importance. Assuming that 20% of the total variables could be informative, the desired number of selected features is set to 36 (line 278).

Supervised classification model performance

DT classifier

DT classification is believed to have several advantages in terms of rapid execution and its simplicity to interpret. The goal for DT classification is to develop a model that predicts the response by learning simple decision rules inferred from the data features. A DT is a flowchart-like structure in which each internal node refers to a test on an attribute, each branch denotes the outcome of the test and each leaf node contains the response. The paths from root (beginning) node down to a leaf node represent classification rules. Deep tree with many leaves is highly accurate on the training data, but not necessary to show a comparable accuracy on an independent test set. A leafy tree is likely to cause a problem of overfitting where its test accuracy is often far less than its training accuracy, while a shallow tree might not be able to attain high training accuracy but can be more robust. Therefore, in this work, MinLeafSize of 10 is determined in order to control the depth of resulting DTs.

Statistical assessments of the model

Classification is the problem of identifying to which of a set of categories a new observation belongs, based on a training set of data containing observations whose category response is known. The statistical measures of the classification performance including the correct classification rate (CCR), sensitivity, specificity G-mean, and Matthews correlation coefficient (MCC) for the cross-validation and validation set. CCR stands for percentage of all samples correctly classified. Sensitivity assesses the proportion of positives that are correctly identified as such, while specificity measures the proportion of negatives that are correctly identified as such. Meanwhile, G-mean indicates the balanced performance of classifier between two classes. Therefore, the higher G-mean value can be achieved only if both sensitivity and specificity are expected to be high simultaneously. As suggested by Kubat et al.,[33] the G-mean is defined as:

$$\mathrm{G-mean} = \sqrt{\mathrm{Sensitivity} \times \mathrm{Specificity}} \tag{2}$$

MCC, introduced by Matthews,[34] is a measure of the quality of binary (two-class) classifications. It takes into account true and false positives and negatives and works as a balanced measure which can be used even if the classes are of very different sizes. Model performance built from the full spectral variables is obtained in

a straightforward manner by running line 290. In a subsequent step, PCA-DT model and LDA-DT models are about to be developed in the light of transformation-based data reduction. For more details, the obtained PC scores matrix (three PCs for PCA and two for LDA) are utilized as the explanatory variables to build classification models. When it comes to validation, a new data set has to be first mean-centered and then projected along the loadings by matrix multiplication producing scores matrix which is appropriate for model validation (lines 295–309).

Further selection based on variance inflation factor

Although the informative variables were selected by different band selection methods, there are strong correlations among these selected variables. As can be seen from Figure 1, most of spectral variables are significantly correlated with each other. The existence of collinearity or multicollinearity is very possible to result in overfitting and underestimation of model error. To address this problem, a post-selection method based on variance inflation factor (VIF) is proposed in this work to remove the variables with strong collinearity. As the name suggested, VIF estimates how much the variance of a coefficient is inflated because of linear dependence with other predictors. It measures the severity of multicollinearity in an ordinary least squares regression analysis. The calculation of VIF for each predictor can be achieved by doing a linear regression of that predictor on all the other predictors. The VIF for the *ith* variable is defined with the following formula:

$$VIF_i = \frac{1}{1 - R_i^2} \tag{3}$$

where R_i^2 is the coefficient of determination of the regression between *ith* variable and all the other variables.

The introduced algorithm called *vif_evaluate* starts with the calculation of VIF value for each selected variable and arrange them in an ascending order based on VIF value. Following that, the first three predictors with the smallest VIF values forms an initial candidate subset. Subsequently, an increasing number of features is selected based on the ranking order to produce more candidate subsets. To evaluate the subset, cross-validation is adopted in the DT model. The best subset is determined with the minimum classification error and set the VIF of the N_j^{th} as the cutoff value. After acquiring the new subset, an important point to note is that the VIF value for each variable is no longer the same as before. Therefore, the whole procedure should be repeated until the optimal cutoff value is the VIF value of the ranking last variable. This function is performed on the selected bands achieved from 12 different approaches (lines 314–323). At the same time, model performance metrics are computed to compare the effectiveness of performing *vif_evaluate* (lines 328–336). Table 1 shows the classification performance after using different band selection methods, while Table 2 exhibits the results after performing *vif_evaluate* to alleviate collinearity. It is quite clear to observe some obvious improvements in model performance. For instance, there are 78 spectral variables collected by VIP scores and the number shrinks to 3 after applying *vif_evaluate*. Nevertheless, better predicted results are obtained after further selection with CCR from 0.9587 to 0.9621 for cross-validation and G-mean from 0.9419 to 0.9620 for validation. It is necessary to announce that variable selection methods based on MC-UVE, RF, PHADIA, and GA are not able to

Table 1. Simplified model performance obtained from various band selection methods.

	Variable number	Cross-validation					Validation				
		CCR	Sensitivity	Specificity	G-mean	MCC	CCR	Sensitivity	Specificity	G-mean	MCC
VIP	78	0.9587	0.9556	0.9609	0.9583	0.9175	0.9425	0.9235	0.9608	0.9419	0.8854
SR	156	0.9632	0.9645	0.9606	0.9625	0.9265	0.9350	0.9031	0.9657	0.9338	0.8714
sMC	95	0.9632	0.9604	0.9655	0.9629	0.9266	0.9350	0.9031	0.9657	0.9338	0.8714
Emcvs	3	0.9653	0.9621	0.9683	0.9652	0.9318	0.9550	0.9337	0.9755	0.9544	0.9106
CARS	43	0.9732	0.9733	0.9721	0.9727	0.9467	0.9450	0.9031	0.9853	0.9433	0.8927
UVE	3	0.9476	0.9471	0.9473	0.9472	0.8952	0.9275	0.8980	0.9559	0.9265	0.8561
RF	4	0.9542	0.9560	0.9529	0.9544	0.9093	0.9175	0.8827	0.9510	0.9162	0.8365
PHADIA	9	0.9721	0.9696	0.9748	0.9722	0.9448	0.9300	0.9133	0.9461	0.9295	0.8602
Relieff	6	0.9642	0.9594	0.9682	0.9638	0.9285	0.9675	0.9490	0.9853	0.9670	0.9355
Stepwise	8	0.9698	0.9672	0.9729	0.9700	0.9401	0.9450	0.9388	0.9510	0.9449	0.8900
SFS	3	0.9799	0.9824	0.9775	0.9800	0.9603	0.9400	0.9184	0.9608	0.9393	0.8806
GA	36	0.9743	0.9755	0.9733	0.9744	0.9487	0.9475	0.9337	0.9608	0.9471	0.8952

CARS: competitive adaptive reweighted sampling; CCR: correct classification rate; GA: genetic algorithm; MCC: Matthews correlation coefficient; RF: random frog; SFS: sequential forward selection; SR: selectivity ratio; sMC: significance multivariate correlation; Emcvs: Ensemble Monte Carlo variable selection; UVE: uninformative variable elimination; VIP: Variable importance in the projection.

Table 2. Simplified model performance after using further selection based on variance inflation factor (VIF).

	Variable number	Cross-validation					Validation				
		CCR	Sensitivity	Specificity	G-mean	MCC	CCR	Sensitivity	Specificity	G-mean	MCC
VIP	3	0.9621	0.9541	0.9714	0.9627	0.9287	0.9625	0.9439	0.9804	0.9620	0.8714
SR	5	0.9699	0.9691	0.9700	0.9695	0.9426	0.9675	0.9490	0.9853	0.9670	0.8900
sMC	29	0.9743	0.9751	0.9730	0.9741	0.9175	0.9475	0.9286	0.9657	0.9469	0.8766
Emcvs	3	0.9653	0.9621	0.9683	0.9652	0.9252	0.9550	0.9337	0.9755	0.9544	0.9255
CARS	11	0.9688	0.9585	0.9794	0.9689	0.9401	0.9500	0.9337	0.9657	0.9495	0.9355
UVE	3	0.9476	0.9471	0.9473	0.9472	0.9486	0.9275	0.8980	0.9559	0.9265	0.8954
RF	4	0.9542	0.9560	0.9529	0.9544	0.9318	0.9175	0.8827	0.9510	0.9162	0.9106
PHADIA	7	0.9720	0.9711	0.9730	0.9721	0.9385	0.9450	0.9235	0.9657	0.9443	0.9003
Relieff	4	0.9653	0.9594	0.9707	0.9650	0.8952	0.9675	0.9490	0.9853	0.9670	0.8561
Stepwise	4	0.9621	0.9626	0.9614	0.9620	0.9093	0.9450	0.9388	0.9510	0.9449	0.8365
SFS	3	0.9799	0.9824	0.9775	0.9800	0.9440	0.9400	0.9184	0.9608	0.9393	0.8906
GA	9	0.9754	0.9698	0.9800	0.9749	0.9307	0.9575	0.9643	0.9510	0.9576	0.9355

CARS: competitive adaptive reweighted sampling; CCR: correct classification rate; GA: genetic algorithm; MCC: Matthews correlation coefficient; RF: random frog; SFS: sequential forward selection; SR: selectivity ratio; sMC: significance multivariate correlation; Emcvs: Ensemble Monte Carlo variable selection; UVE: uninformative variable elimination; VIP: Variable importance in the projection.

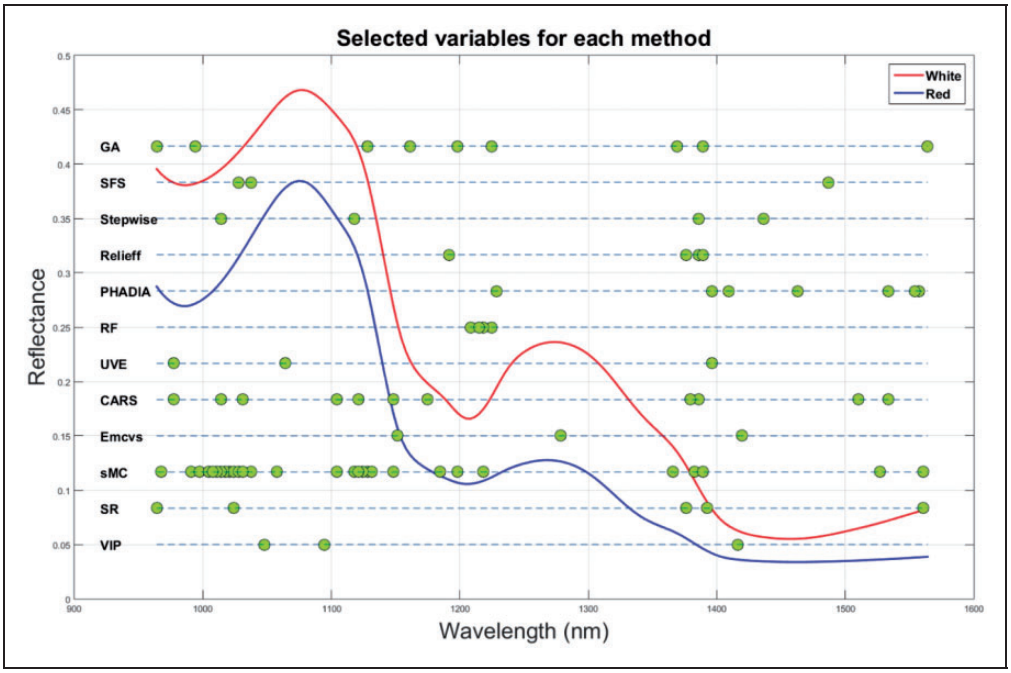

Figure 3. The visualization and comparison of selected variable subsets achieved from different methods.

produce reproducible results at each run. Therefore, results obtained might differ from those presented here.

Selected variables subsets analysis

More often than not, variable selection is conducted in an automated fashion in a black-box approach. In order to build proper models, it is necessary to inspect what variables are selected. The vibrations of the molecular chemical bonds, such as O–H, C–H, and N–H, seen in the near infrared region can provide important information about chemical components. For illustrative purposes, all the spectral variables collected from 12 band selection methods are shown in Figure 3 (lines 342–365). Visually, there is a large variation in the selected variables by different methods. It is necessary to realize that each variable selection approach works under certain assumption and based on various statistical or mathematic rationales, therefore, it makes sense why there are no two methods that generate the same outcome. It is obvious to detect that the most dominant region is located around 1400 nm and almost every method has selected at least one spectral variable in this neighborhood region. It was reported that 1400–1500 nm

region is generally related with absorption ascribed to the first overtone of O–H stretch vibration.[35] Besides, sMC, CARS, and Relieff imply that the wavelengths located around 1200 nm could also be important, which corresponds to the fact that the absorption peaks at about 1200 are due to overtone bands for the lipid.[36] It is critical to note that the results could be slightly different because of the inconsistency in some methods. It should be clarified that optimal variables are selected after MSC preprocessing, and therefore caution is needed during wavelength interpretation.

Classification model performance comparison

In order to use the information obtained from the correlation analysis, wavelengths at 1322 nm with the highest positive correlation and 1483 nm with the most negative correlation are selected as anther variable subset and it is named as Cor which is short from correlation (lines 396–399). Table 3 summaries the classification performance of models built in this work. The main goal of data reduction is to produce parsimonious models with better predictive performances in comparison with the model based on the whole original variables. In this sense, LDA and band selection based on UVE, RF, and correlation are not recommended in this case because their performance is inferior in every respect of statistical assessments. However, PCA, as a widely-used data reduction approach, has proven to be very effective with higher CCRs and G-means in cross-validation and validation compared with the full model. Given enough computation time, band selection method based on GA might eventually match and possibly surpass other methods, yet its stochastic nature makes it difficult to estimate the number of iterations which would be required. For a fair comparison, it is not difficult to detect that VIP, SR, and Relieff have outperformed other methods in terms of validation results. SR is observed to rank the first place in CCR, G-mean, and MCC for both cross-validation and validation. Other methods also work well as they have shown better performance in terms of some statistical assessments. It should be noted that optimization of these methods could be achieved if different parameter settings were used. In addition, confusion metrics for validation set are subsequently provided in Figure 4 (lines 406–415). For most built models, the sensitivity is comparably lower than specificity, meaning that it is more likely to wrongly classify actual white stripe pixels into red muscle category. However, approach based on GA presents the opposite phenomenon where only seven red muscle pixels are wrongly predicted as white stripe.

Prediction maps

Three hypercubes from different parts of salmon are employed to further test the model accuracy. To save computing time, these hypercubes have suffered from image calibration and background removal. The occurrence of dead pixels and spikes is common in hyperspectral imaging, thereby it is necessary to identify and compensate for them by interpolation. Therefore, a simple and straightforward function named as *findspike_im* was developed. The core of this function is to generate a difference spectrum obtained by subtracting the intensity value at each wavelength from average

Table 3. Final models developed in this work to classify white strip and red muscle for salmon fillets.

	Variable number	Cross-validation					Validation				
		CCR	Sensitivity	Specificity	G-mean	MCC	CCR	Sensitivity	Specificity	G-mean	MCC
Full	180	0.9643	0.9667	0.9606	0.9636	0.9287	0.9350	0.9031	0.9657	0.9338	0.8714
PCA	3	0.9710	0.9665	0.9753	0.9709	0.9426	0.9450	0.9388	0.9510	0.9449	0.8900
LDA	1	0.9509	0.9514	0.9506	0.9510	0.9023	0.9350	0.9235	0.9461	0.9347	0.8701
VIP	3	0.9621	0.9541	0.9714	0.9627	0.9252	0.9625	0.9439	0.9804	0.9620	0.9255
SR	5	0.9699	0.9691	0.9700	0.9695	0.9401	0.9675	0.9490	0.9853	0.9670	0.9355
sMC	29	0.9743	0.9751	0.9730	0.9741	0.9486	0.9475	0.9286	0.9657	0.9469	0.8954
Emcvs	3	0.9653	0.9621	0.9683	0.9652	0.9318	0.9550	0.9337	0.9755	0.9544	0.9106
CARS	11	0.9688	0.9585	0.9794	0.9689	0.9386	0.9500	0.9337	0.9657	0.9495	0.9003
UVE	3	0.9476	0.9471	0.9473	0.9472	0.8952	0.9275	0.8980	0.9559	0.9265	0.8561
RF	4	0.9542	0.9560	0.9529	0.9544	0.9093	0.9175	0.8827	0.9510	0.9162	0.8365
PHADIA	7	0.9720	0.9711	0.9730	0.9721	0.9440	0.9450	0.9235	0.9657	0.9443	0.8906
Relieff	4	0.9653	0.9594	0.9707	0.9650	0.9307	0.9675	0.9490	0.9853	0.9670	0.9355
Stepwise	4	0.9621	0.9626	0.9614	0.9620	0.9242	0.9450	0.9388	0.9510	0.9449	0.8900
SFS	3	0.9799	0.9824	0.9775	0.9800	0.9603	0.9400	0.9184	0.9608	0.9393	0.8806
GA	9	0.9754	0.9698	0.9800	0.9749	0.9509	0.9575	0.9643	0.9510	0.9576	0.9151
Cor	2	0.9643	0.9693	0.9603	0.9648	0.9203	0.9375	0.8980	0.9755	0.9359	0.8772

CARS: competitive adaptive reweighted sampling; Emcvs: Ensemble Monte Carlo variable selection; GA: genetic algorithm; LDA: linear discriminant analysis; PCA: principal component analysis; RF: random frog; SFS: sequential forward selection; sMC: significance multivariate correlation; SR: selectivity ratio; UVE: uninformative variable elimination.

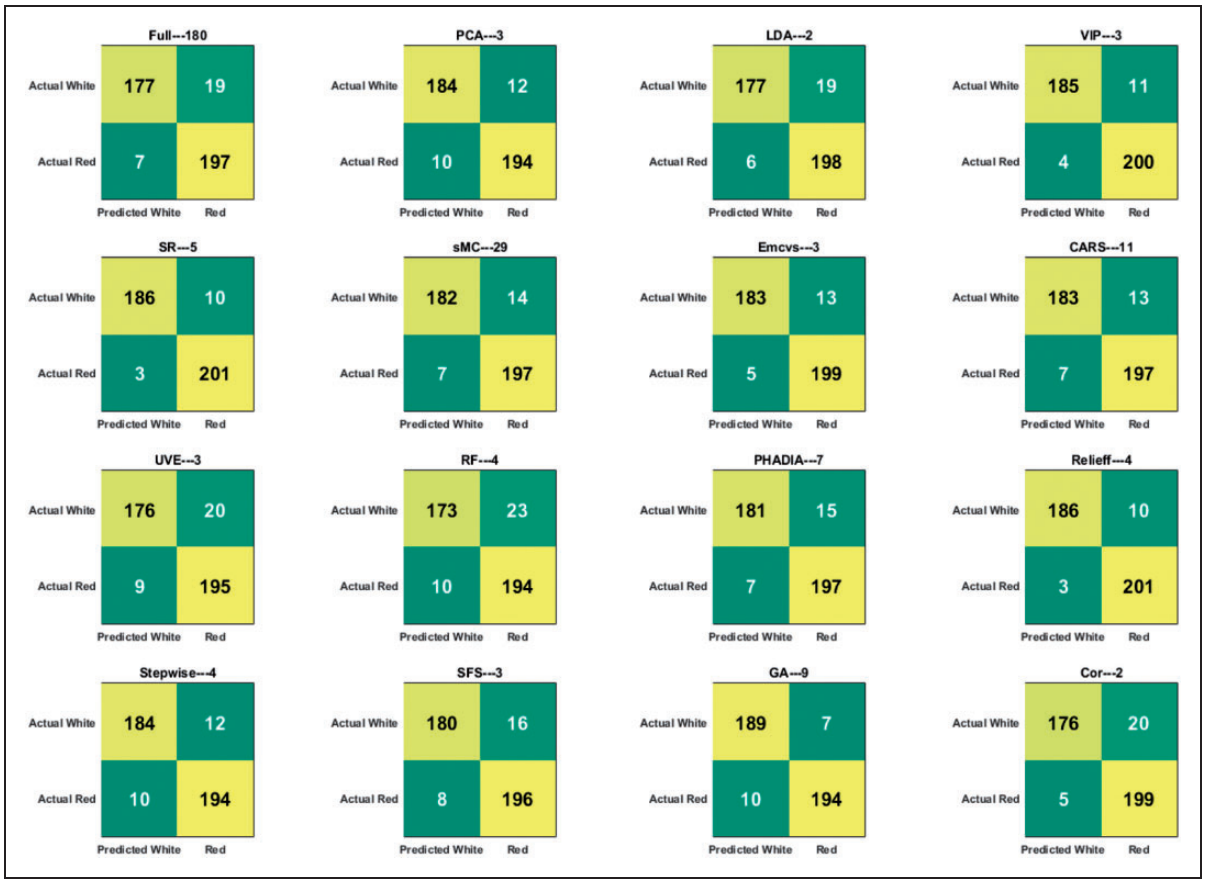

Figure 4. Confusion matrices for validation dataset obtained from different models.

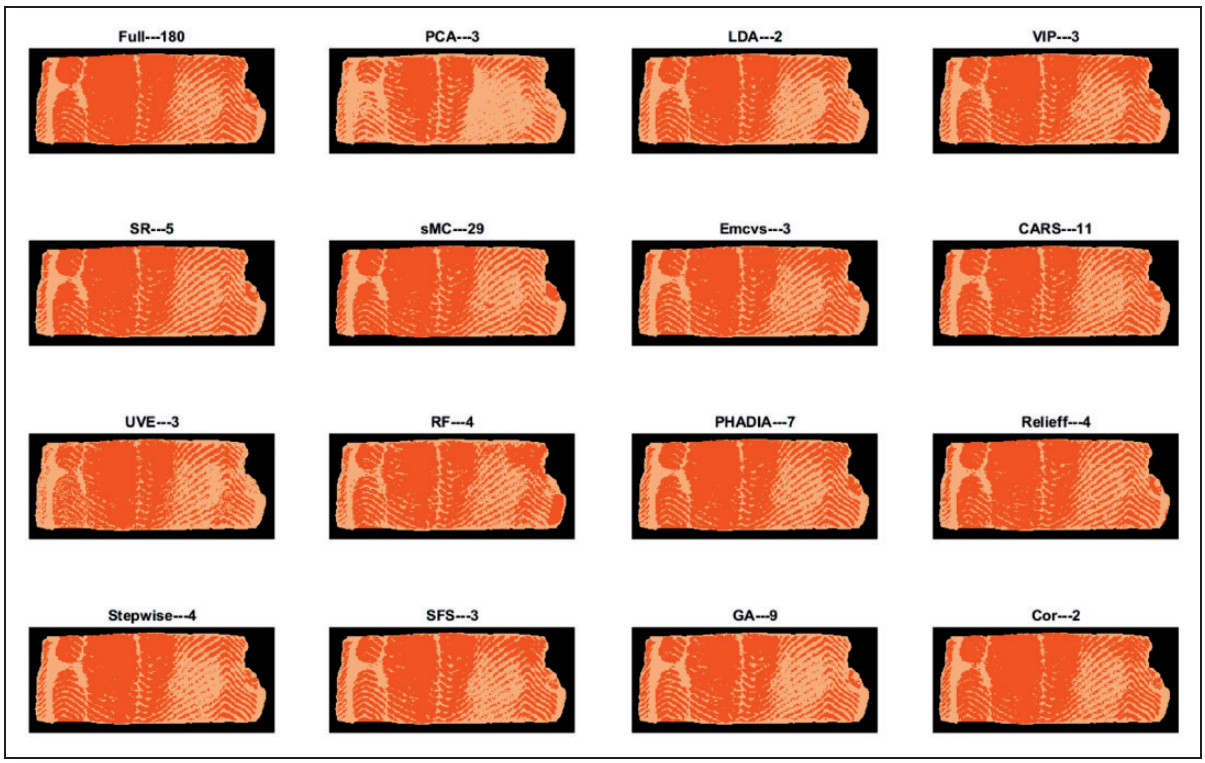

Figure 5. Prediction maps obtained from different models for sample 1.

intensity of the preceding and posterior wavelengths. The ideal behind this is that the difference spectrum is quite flat for normal spectra ascribed to the smoothly varying nature of spectra in general, but for spike and dead pixel, the difference spectrum of spectra will present a sharp discontinuity. Likewise, a ''for'' loop is adopted to apply this function to three hypercubes (420–423). The next step is to apply MSC on each hypercube and this can be easily implemented by a function called *msc_im* and the reference spectrum is the mean spectrum of the calibration set (lines 426–428).

When it comes to generate prediction map, the first and fundamental step is to ''unfold'' hypercubes such that the three-dimensional information is rearranged in two-dimensional matrix where each row represents the spectrum from each pixel and each column refers to a certain wavelength variable. The established classification models are then employed to classify each pixel

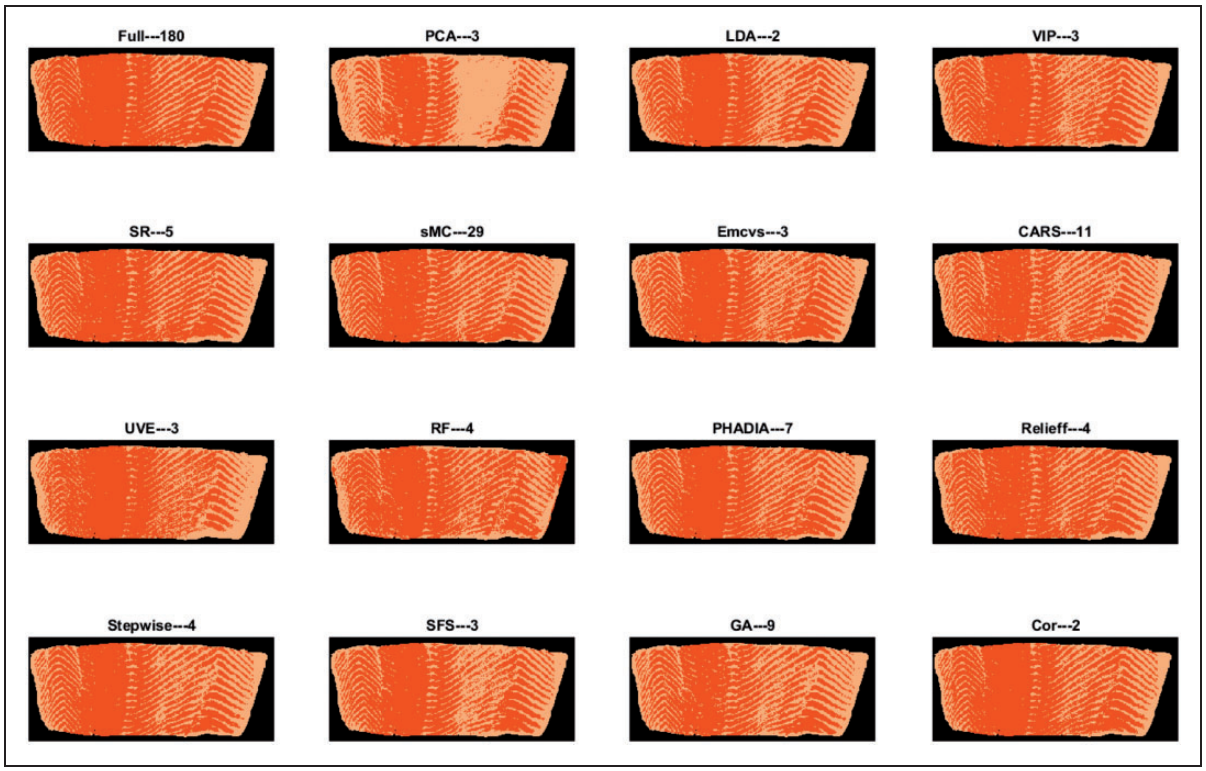

Figure 6. Prediction maps obtained from different models for sample 2.

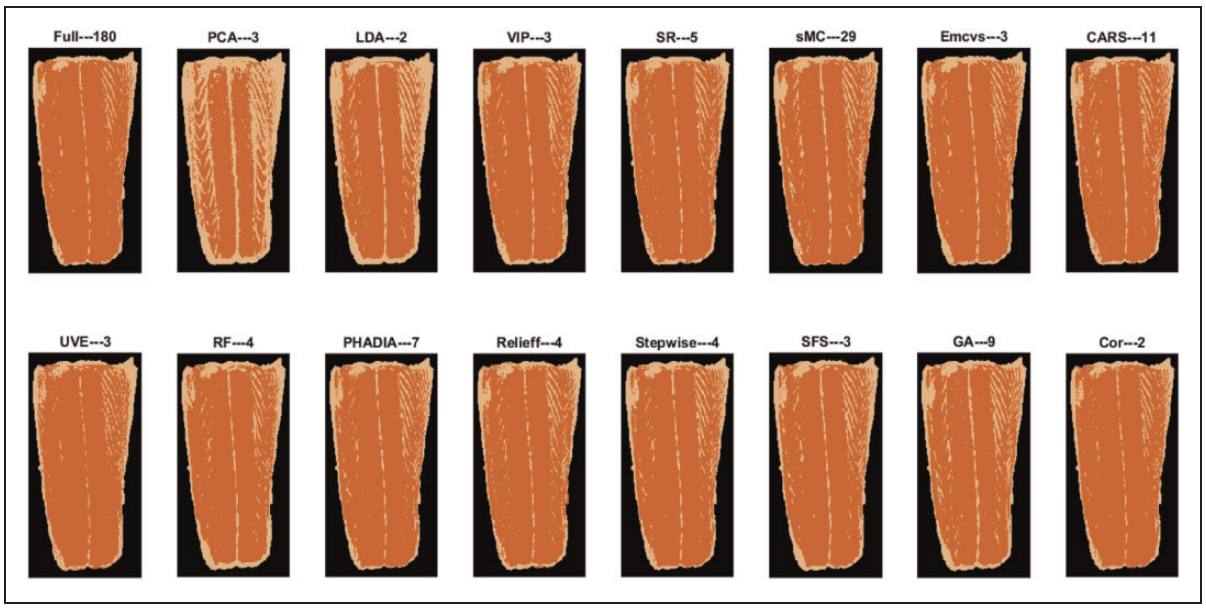

Figure 7. Prediction maps obtained from different models for sample 3.

into white stripe or red muscle categories. The obtained matrix contains "0" if the pixel was identified as background; "1" if the pixel was classified as white stripe or "2" if it was recognized as red muscle. This matrix is then transformed to produce a classification map where white colour is assigned to white stripe class and red color to red muscle to match the true color in the real world. In our case, the classification maps are built by running command lines from 470 to 493 and the figures are generated by executing codes from lines 499 to 520.

Visually, it is found in Figure 5 that most models for sample 1 have presented satisfactory results. However, with relatively high sensitivity and high specificity for validation performance in Table 3, PCA has presented unpleasant result in sample 1 prediction map with a large number of red muscle pixels wrongly classified as white stripe. VIP, SR, and Relieff are witnessed to outperform other selection methods with satisfying classification map presented. In analogy to sample 1, the prediction maps for sample 2 are displayed in Figure 6 with similar results obtained. Sample 3 (Figure 7) is from a tail sample where there are less obvious white stripes. High false alarm rate for white stripe is also observed in PCA.

Conclusion

Data reduction and wavelength selection are an old and ever-growing research field in chemometrics. There is a large number of studies conducted to explore different approaches and investigate the possibility for useful applications. It is not possible to list all the methods, therefore, only some frequently used methods are included in this work. These algorithms work in different manners and they have been designed for various applications. It is thereby difficult to draw a conclusion of which algorithm is best suited for a certain type of data. It is thus necessary to conduct trial and error studies to find the optimal one.

This work provides a framework for the manipulation of different data dimension reduction methods for near infrared hyperspectral imaging data using the Matlab computing language. Starting with raw spectra, preprocessing and outlier detection were conducted prior to using various data reduction methods. Reduced models were built by DT classifier and their performance was compared in terms of cross-validation and validation. More importantly, visual observation of prediction maps in conjunction with the performance metric was testified to be a useful method for evaluating model performance. Several conclusions can be drawn after comparing these data reduction methods. First of all, selection-based methods are highly recommended in this case. For the application of hyperspectral imaging on monitoring food quality issues, a band selection method is typically preferred over transformation when one wishes to find important wavelength variables and then develop the corresponding multispectral imaging system. Transformation-based methods would be advantageous when the extracted features are most related to the classification problem, but this is not the case for our example. Among all these band selection methods, PLS-based methods (VIP, SR, and sMC) have yielded reasonably good classification results. Compared to other band selection methods, PLS-based methods are remarkably simple, reliable, and fast. Nevertheless, the rule of thumb criterion for PLS-based methods is likely to produce too many selected variables which are mostly in the same spectral neighbor regions with high correlation among each other. MCS methods (Emcvs and CARS) used in this example also have led to relatively good classification performances. However, MCS methods have the bottleneck of being potentially slow. With respect to MPA approaches (UVE, RF, and PHADIA), no outstanding performances were observed. Worse still, these methods tend not to produce reproducible results at each run and it is usually quite time-consuming to perform. Relieff has presented impressive classification results with acceptable computation runtime, while Stepwise and SFS have revealed medium prediction ability. As for the GA-guided method, it has always been reported to be powerful yet modest performance was achieved in this work. A band selection method based on GA is recommended if enough computation runtime can be guaranteed. Thirdly, the VIF further selection method proposed from this work was testified to be very useful to alleviate collinearity problem. For a fair comparison, the best method is SR coupled with VIF further selection based on the statistical assessments and prediction maps. In order to provide a realistic estimate of the performance of the final-chosen SR model on completely unseen data, test data set can be used by running command line at 525 with the accuracy of 91% achieved. In a brief conclusion, different data reduction methods will present different applicability for a particular imaging data set. It is always useful to compare their results in different manners and choose the best one in terms of accuracy and computation runtime.

All of these aforementioned data reduction methods can be easily adapted to solve binary classification problem occurred in hyperspectral imaging analysis. Modifications are required in case of a multi-class classification issue. This work also offers some original codes and functions in Matlab programming language, such as fast MSC preprocessing, optimal cutoff value determination, spikes, and dead spectra identification and correction, etc. Although these procedures are common practice when working with hyperspectral imaging, each step plays an important role in producing satisfying performance and minor changes will alter the final results. These codes and functions can be modified when attempting to solve each individual problem in hyperspectral imaging analysis. In a nutshell, although a multispectral imaging system with a limited number of wavelengths can remarkably reduce the prolonged time, it is usually very tricky to decide which subsets of selected wavelengths are optimal with best performance provided, because employing various

approaches can always lead to different effective wavelengths combinations.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors would like to acknowledge the UCD-CSC Scholarship Scheme supported by University College Dublin (UCD) and China Scholarship Council (CSC) for financial support of this study.

References

1. Jackman P, Sun D-W, Du C-J, et al. Prediction of beef eating qualities from colour, marbling and wavelet surface texture features using homogenous carcass treatment. *Pattern Recognit* 2009; 42: 751–763.
2. Sun D-W and Brosnan T. Pizza quality evaluation using computer vision – part 1 – pizza base and sauce spread. *J Food Eng* 2003; 57: 81–89.
3. Sun D-W. Computer vision – an objective, rapid and non-contact quality evaluation tool for the food industry. *J Food Eng* 2004; 61: 1–2.
4. Jackman P, Sun D-W and Allen P. Automatic segmentation of beef longissimus dorsi muscle and marbling by an adaptable algorithm. *Meat Sci* 2009; 83: 187–194.
5. Du CJ and Sun D-W. Pizza sauce spread classification using colour vision and support vector machines. *J Food Eng* 2005; 66: 137–145.
6. Sun D-W and Brosnan T. Pizza quality evaluation using computer vision – part 2 – pizza topping analysis. *J Food Eng* 2003; 57: 91–95.
7. Sun D-W. Computer vision – an objective, rapid and non-contact quality evaluation tool for the food industry. *J Food Eng* 2004; 61: 1–2.
8. Jackman P, Sun D-W and Allen P. Recent advances in the use of computer vision technology in the quality assessment of fresh meats. *Trends Food Sci Technol* 2011; 22: 185–197.
9. Xu J-L, Riccioli C and Sun D-W. Comparison of hyperspectral imaging and computer vision for automatic differentiation of organically and conventionally farmed salmon. *J Food Eng* 2017; 196: 170–182.
10. Xu J-L and Sun D-W. Identification of freezer burn on frozen salmon surface using hyperspectral imaging and computer vision combined with machine learning algorithm. *Int J Refrig* 2017; 74: 151–164.
11. Dorrepaal R, Malegori C and Gowen A. Tutorial: time series hyperspectral image analysis. *J Near Infrared Spectr* 2016; 24: 89–107.
12. ElMasry G, Sun D-W and Allen P. Chemical-free assessment and mapping of major constituents in beef using hyperspectral imaging. *J Food Eng* 2013; 117: 235–246.
13. Cheng J-H and Sun D-W. Rapid and non-invasive detection of fish microbial spoilage by visible and near infrared hyperspectral imaging and multivariate analysis. *LWT Food Sci Technol* 2015; 62: 1060–1068.
14. Xiong Z, Sun D-W, Pu H, et al. Non-destructive prediction of thiobarbituric acid reactive substances (TSARS) value for freshness evaluation of chicken meat using hyperspectral imaging. *Food Chem* 2015; 179: 175–181.
15. Cheng J-H, Sun D-W, Pu H, et al. Development of hyperspectral imaging coupled with chemometric analysis to monitor K value for evaluation of chemical spoilage in fish fillets. *Food Chem* 2015; 185: 245–253.
16. Pu H, Kamruzzaman M and Sun D-W. Selection of feature wavelengths for developing multispectral imaging systems for quality, safety and authenticity of muscle foods – a review. *Trends Food Sci Technol* 2015; 45: 86–104.
17. Ma J, Sun D-W and Pu H. Spectral absorption index in hyperspectral image analysis for predicting moisture contents in pork longissimus dorsi muscles. *Food Chem* 2016; 197: 848–854.
18. Cheng J-H, Sun D-W, Qu J-H, et al. Developing a multispectral imaging for simultaneous prediction of freshness indicators during chemical spoilage of grass carp fish fillet. *J Food Eng* 2016; 182: 9–17.
19. Cheng J-H and Sun D-W. Partial least squares regression (PLSR) applied to NIR and HSI spectral data modeling to predict chemical properties of fish muscle. *Food Eng Rev* 2017; 9: 36–49.
20. Li J-L, Sun D-W, Pu H, et al. Determination of trace thiophanate-methyl and its metabolite carbendazim with teratogenic risk in red bell pepper (Capsicumannuum L.) by surface-enhanced Raman imaging technique. *Food Chem* 2017; 218: 543–552.
21. Xu J-L, Riccioli C and Sun D-W. Development of an alternative technique for rapid and accurate determination of fish caloric density based on hyperspectral imaging. *J Food Eng* 2016; 190: 185–194.
22. Cheng J-H, Sun D-W, Pu H-B, et al. Suitability of hyperspectral imaging for rapid evaluation of thiobarbituric acid (TBA) value in grass carp (Ctenopharyngodon idella) fillet. *Food Chem* 2015; 171: 258–265.
23. Pu H, Xie A, Sun D-W, et al. Application of wavelet analysis to spectral data for categorization of lamb muscles. *Food Bioprocess Technol* 2015; 8: 1–16.
24. Dai Q, Cheng J-H, Sun D-W, et al. Prediction of total volatile basic nitrogen contents using wavelet features from visible/near-infrared hyperspectral images of prawn (Metapenaeus ensis). *Food Chem* 2016; 197: 257–265.
25. Cheng W, Sun D-W and Cheng J-H. Pork biogenic amine index (BAI) determination based on chemometric analysis of hyperspectral imaging data. *LWT Food Sci Technol* 2016; 73: 13–19.
26. Pu H, Liu D, Wang L, et al. Soluble solids content and pH prediction and maturity discrimination of lychee fruits using visible and near infrared hyperspectral imaging. *Food Anal Meth* 2016; 9: 235–244.
27. Cheng J-H and Sun D-W. Partial least squares regression (PLSR) applied to NIR and HSI spectral data modeling to predict chemical properties of fish muscle. *Food Eng Rev* 2017; 9: 36–49.
28. Lorente D, Blasco J, Serrano A, et al. Comparison of ROC feature selection method for the detection of decay in citrus fruit using hyperspectral images. *Food Bioproc Tech* 2013; 6: 3613–3619.
29. Zhang C, Jiang H, Liu F, et al. Application of near-infrared hyperspectral imaging with variable selection methods to determine and visualize caffeine content of coffee beans. *Food Bioproc Tech* 2017; 10: 213–221.

30. Kononenko I, Šimec E and Robnik-Šikonja M. Overcoming the myopia of inductive learning algorithms with RELIEFF. *Appl Intellig* 1997; 7: 39–55.
31. Marcano-Cedeno A, Quintanilla-Domínguez J, Cortina-Januchs M, et al. Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network. In: *IECON 2010-36th annual conference on IEEE industrial electronics society*. IEEE, Glendale, AZ, USA, 2010, pp. 2845–2850.
32. Ludwig O and Nunes U. Novel maximum-margin training algorithms for supervised neural networks. *IEEE Trans Neural Networks* 2010; 21: 972–984.
33. Kubat M, Holte RC and Matwin S. Machine learning for the detection of oil spills in satellite radar images. *Mach Learn* 1998; 30: 195–215.
34. Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta–Protein Struct* 1975; 405: 442–451.
35. Osborne BG and Fearn T. *Near infrared spectroscopy in food analysis*. Harlow, Essex, England: Longman Scientific & Technical, 1986.
36. Nagao R, Ishii K and Awazu K. Optimal wavelengths for near-infrared multispectral imaging of atherosclerotic plaque. *Opt Rev* 2015; 22: 329–334.