Machine Learning Assisted Species Assignment in Oak Trees

A thesis submitted to attain the degree of

Diploma of Advanced Studies in Applied Statistics

ETH Zurich

presented by

Diana Coman Schmid

PhD

E-mail: diana.coman@id.ethz.ch

Current affiliation: Department of Informatics, Scientific IT Services, ETHZ, Switzerland

2017

Abstract

In European white oaks (*Quercus spp.*), species delimitation is not trivial because of the large overlap of morphological characteristics, which are likely due to hybridization of these species. Here, I evaluate the feasibility of several machine learning methods for accurately predicting oak tree species based on morphological and molecular data. The best machine learning model according to the 10-fold cross-validation misclassification rate, correctly assigned the species in 90 % of the individual oak trees. In conclusion, machine learning techniques can be useful in identifying oak tree species based on morphological and molecular data.

Keywords

machine learning, classification, prediction, species, oak trees

Table of Contents

1 Background

Species identification and delimitation in the most common European white oaks *Quercus petraea*, *Quercus pubescens* and *Quercus robur* is subject to debate because of the large overlap of morphological characteristics (e.g. leaf related characteristics), which in turn are likely due to intraspecific hybridization [1]. Morphological variation in mixed oak stands (relatively homogeneous tree communities) composed of *Q. petraea*, *Q. robur* and *Q. pubescens* and consequently the ability to distinguish between the species is of general interest in Europe. Hence, the need for reliable methods for oak trees species spans a wide range of domains, from real-world applications to research. Foresters need out-of-box morphological screening methods, dendrologists could make use of classification criteria for taxonomic purposes, whereas biologists and ecologists focus on finding traits which could be used for studying introgression between these species known to hybridize [2].

Significant efforts have been devoted over the years to define reliable morphological traits to be assessed together with the appropriate statistical methods to analyze such data [2]. More recently, molecular techniques, potentially able to distinguish intermediate morphologies occurring due to intraspecific hybridization, have been used to assist species identification [1]. Yet, there is no consensus method to date that is widely applied to delineate oak tree species. Multivariate statistical analyses of leaf morphological characters and molecular markers were shown to contribute to a better species assignment in European white oaks. Furthermore, model-based approaches using exclusively molecular markers data was sufficient to provide a satisfying congruency in species assignment [1]. To date, machine learning (ML) methods have not been used to predict species assignment based on morphological and molecular data in white oak trees.

Here, I evaluate several statistical machine learning methods for their capacity to reliably predict species assignment in European white oaks (*Q. petraea*, *Q. pubescens*, *Q. robur*). The data set was provided by Rellstab et al. and consisted of the variable measurements for 1,369 individual oak trees from 71 populations sampled across Switzerland [1]. I assessed seven ML methods with three sets of variables: morphological, molecular and combined. To evaluate the ML model performance, I used the misclassification rate estimated in a 10-fold cross-validation procedure. A schematic workflow of this analysis is shown in Figure 1.

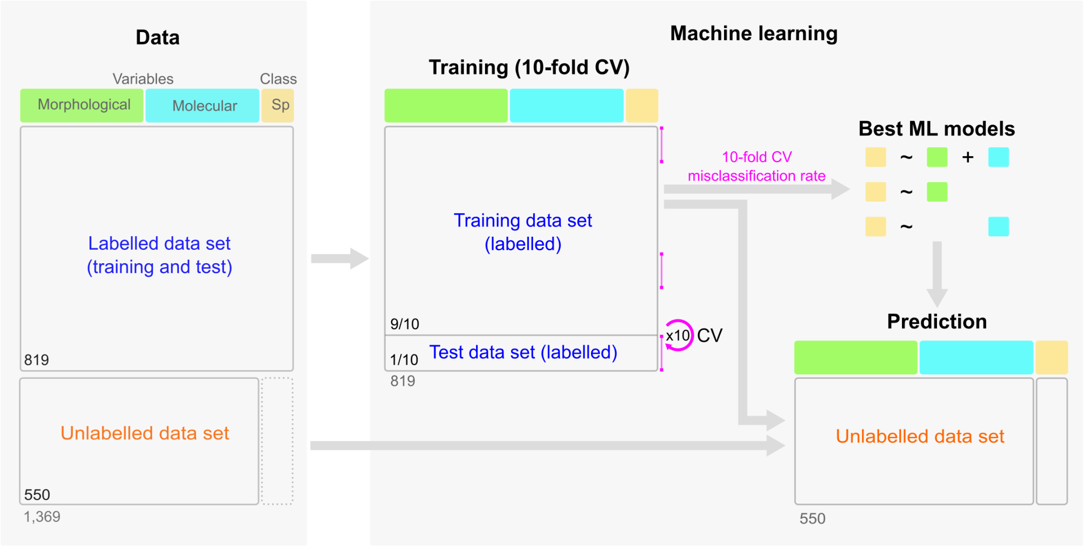


Figure 1. Schematic representation of the data analysis workflow, which shows the structure of the data and the analytical approach. The data set consists of 1,369 individual oak trees and their measured morphological and molecular variables (additional available variables, e.g. geographical location, are not shown). Almost 60 % of the individual observations could be assigned to one of the three species based on the leaf trichome profiles (see Statistical Methods and Results) and is interchangeable referred here to as "labelled data set" or "training and test data set". The ML models are trained on the labelled data set. To estimate ML model`s performance, 10-fold CV is used: at each of the 10 iterations, 1/10 of the labelled data set is used as "test data" and the remaining 9/10 constitutes the "training data". The splitting of the data is random with1/10 or the "test data" occurring randomly at different location within the labelled data set, indicated by the magenta lines. The species for the remaining 40 % of the data (interchangeable referred here to as "unlabelled data set” or "new data set") are predicted based on the best performing models.

I begin the **Data** section with an exploratory data analysis followed by data preprocessing as a prerequisite step for ML data input. Next, in the **Statistical Methods and Results** section I describe and summarize the ML models that have been trained and evaluated by 10-fold cross-validation for their performance to accurately predict the oak trees species assignment. Based on the best performing models (according to the misclassification rate) I show the species prediction results for the oak trees without taxon identification information. Finally, I display graphically the species composition (“known” based on the available trichome profiles and “ML-predicted” based on the best ML models) of the 71 oak tree populations onto the map of Switzerland, providing support in evaluating which oak tree population are likely mixed and therefore prone to hybridization. Finally, I conclude that the LDA and Neural Networks are the best performing ML models on this data set, correctly assigning the species in 90 % of the cases. Hence, ML methods have good potential to be used as tool for reliable prediction of oak trees species.

2 Data

The data set used in this thesis was provided by Dr. C. Rellstab (WSL, Swiss Federal Institute for Forest, Snow and Landscape Research) and consists of morphological and molecular measurements for 1,369 trees of 71 (pure and mixed) oak trees populations sampled across Switzerland [1]. Covering a wide range of morphological characteristics, this data set provides seven relative (size independent) leaf parameters: lamina shape or obversity (OB), petiole ratio (PR), lobe depth ratio at first lobe (LDR), lobe depth ratio at widest lobe (LDRW), percentage of venation (PV), lobe width ratio (LWR), lobe number ratio (LNR). Additionally, the basal shape of the lamina (BS) is provided. Furthermore, the presence or absence of the following trichome types (leaf hair) is available: laminal (leaf blade) stellate trichomes (LS), clustered trichomes on the lamina (LC), intermediate (between stellate and clustered) trichomes on the lamina (LI), stellate trichomes on the leaf vein (VS), and clustered trichomes on the leaf vein (VC). The molecular variables are the principal components scores (PCo1 to PCo6) obtained based on the pairwise co-dominant genotypic distances analysis of eight nuclear microsatellite markers [1]. In addition to the morphological and molecular data, the identity and geographic coordinates (latitude, longitude, elevation) for each of the 71 oak tree populations is provided.

Here, I use the term "observations" to refer to the individual trees and the term "variables" to indicate the morphological and molecular variables (e.g., PR, LNR, PCo1, PCo2). The following abbreviations are used for the oak tree species: "pe" for *Q. petraea*, "pu" for *Q. pubescens* and "ro" for *Q. robur*. The data analysis presented here is performed with the R software version 3.3.3 [5] and RStudio Version [6].

The R script used for this analysis can be found in Appendix 1.

2. 1 Data Preprocessing

The original data file containing the values for the morphological and molecular variables of the 1,369 individual oak trees was merged with the original data file containing the geographical information for the 71 oak tree populations. A summary of the original data is shown in Supplementary Table 1. Based on the evident positive skewness of the LNR variable and the different scales of the morphological variables (Figure 2A), the data was transformed as follows: the values for the LNR variable were transformed onto log (base 2) scale and subsequently, all morphological variables (BS, OB, PR, LDR, LDRW, LNR, PV and LWR) were scaled as follows: for each variable, the mean was subtracted from each value, which was next divided by the respective standard deviation.

To quantitatively evaluate the ML model accuracy (the misclassification rate), the observations in the train and test data set must have labels, i.e., their class identity is known [3][4]. Specifically, the individual trees must be assigned to one of the three oak species considered here *Q. petraea*, *Q. pubescens* or *Q. robur*.

The trichome types on the lamina (leaf blade hairs) are morphological characteristics that can be used to delineate oak tree species. However, assigning trichomes to specific types is not trivial due to the existence of intermediate leaf hair types. *Q. pubescens* has mostly clustered trichomes, whereas *Q. petraea* has stellate trichomes. *Q. robur* is glabrous (hairless) on the lamina [1].

To generate a surrogate variable indicating the species assignment (class labels), the trichome variables were used as follows: trees with stellate trichomes exclusively (no other or intermediate trichome types) were assigned to *Q. petraea*, trees with clustered trichomes exclusively were assigned to *Q. pubescens* and glabrous trees were assigned to *Q. robur*. This resulted in a data set, referred here to as "the labelled data set" of 819 individual trees for which morphological, molecular and species identity data is available. The remaining 550 observations, referred here to as "the unlabelled data set" with unresolved species identity, were kept aside and used for prediction, once the best ML models had been selected (Figure 1). The trichome variables (LS, LC, LI, VS and VC) used to generate the class labels were not included in the ML models.

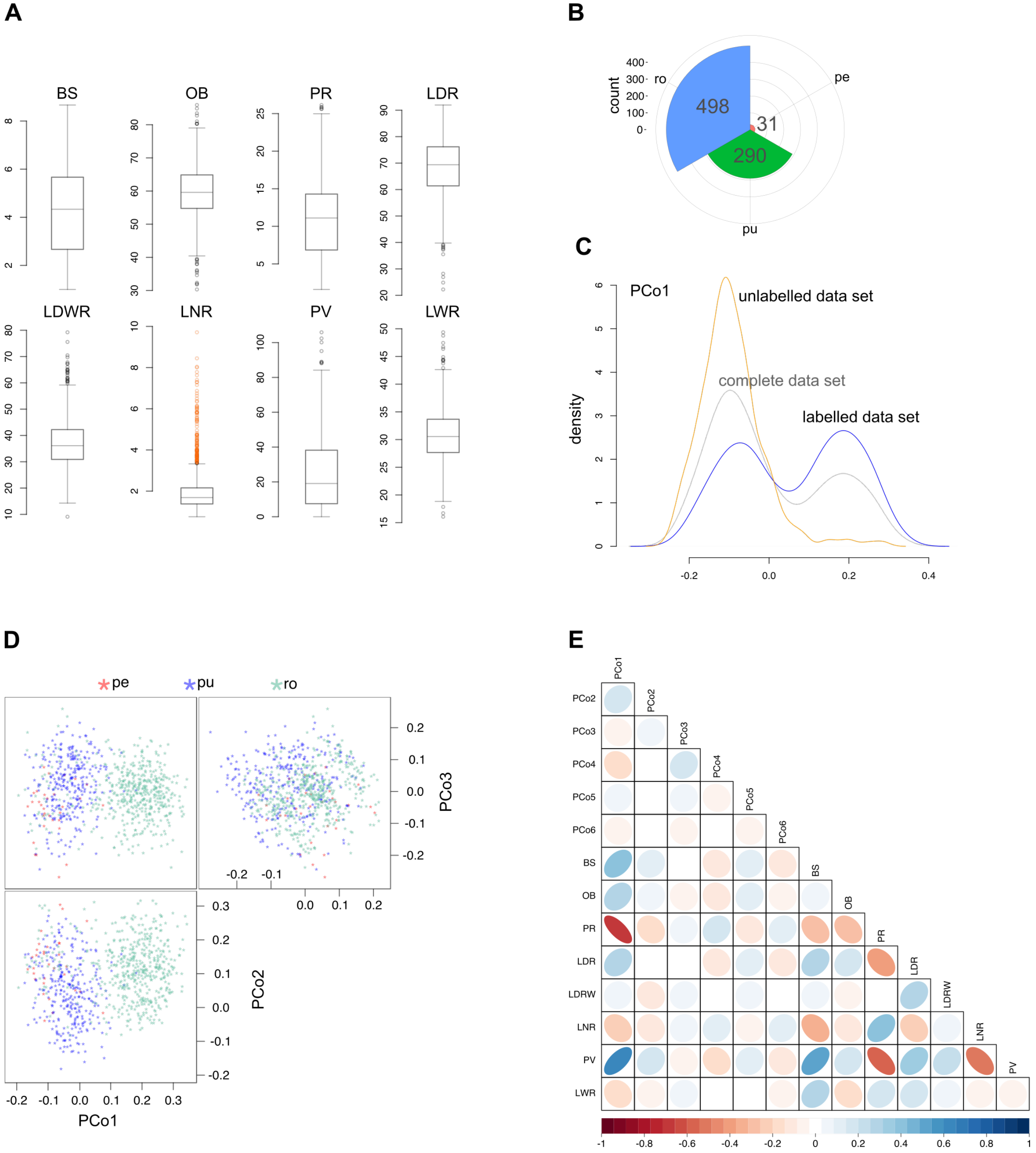


Figure 2. Exploratory data analysis. **A.** The box plots show the different scales of the morphological variables measured values. Specifically, the positive skewness of the LNR predictor is emphasized in orange. **B.** The pie chart shows the proportion of the three oak tree species in the labelled data set. The number of observation for each of the three oak tree species is shown. **C.** The distinct distribution of the PCo1 values in the labelled, unlabelled and complete data set. **D.** The scatter plot matrix of the PCo1, PCo2 and PCo3 molecular variables indicate their influence on the species delineation. The three oak tree species are displayed in distinct colors, **E.** The pairwise correlation of the morphological and molecular variables is shown. For example, the PR morphological variable is strongly negatively correlated to the PCo1 molecular variable.

2.2 Exploratory Data Analysis

The species proportion in the labelled data set is shown in Figure 2B and shows the underrepresentation of the *Q. petraea* species, with only 31 individuals. Furthermore, the distribution of the values for the PCo1 and PR variables in the labelled and unlabelled data sets is different (Figure 2C and Supplementary Figure 1). For example, the PCo1 distribution in the unlabelled data set peaks near to -0.1, whereas the distribution of PCo1 in the labelled data set is bimodal with lower peaks, near to -0.1 and 0.2. Together, these raise a warning for the interpretation of the ML approach on this data, because the data set used for training the model might not fully reflect the unlabelled data set, for which the prediction should be made. To reveal possible influence of the morphological and molecular variables on the species delineation, the scatter plot matrices of the transformed variables (see 2.1 Data Preprocessing and Supplementary Figure 2) were inspected (Figure 2D). The molecular variable PCo1 appears to drive the separation into two groups, one containing the *Q. petraea* and *Q. pubescens* species, and the other populated with the *Q. robur* trees. The PR and BS morphological variables also show a potential contribution to species delimitation (Supplementary Figure 2). Furthermore, there is a negative correlation between PCo1 and PR, whereas the remaining variables show moderate or no correlation (Figure 2E). These indicate that a subset of the variables (PCo1, PCo2, PR) most likely contribute most to delimitation of the three oak tree species.

3 Statistical Methods and Results

The goal of statistical classification, a domain of machine learning, is to assign a new observation to a defined class based on a training set of the data containing observations whose class membership is known. A broad range of ML methods are available for classification purposes, for example, discriminant analyses, decision trees, neural networks. An ML model should perform well not only on the training data set, but also on the test data set, containing observations with known class membership but that were not used to train the ML model. This represents the generalization error (test error) of an ML model and it reflects its prediction capability on separate test data, qualitatively guiding in practice the choice of an ML model (Figure 1). For example, cross-validation (CV) is a simple method that can be used to directly estimate the generalization error (e.g., the number of misclassified observations) associated with a given statistical learning method in order to evaluate its performance. Once a ML model is selected based on the aforementioned criteria, it can be used to classify new observations for which variable measurements are available [3][4].

Note 1

Reproducibility: each ML method is run under a specified seed for the random number generation process (using an arbitrary value, e.g., "set.seed(909)"), to insure reproducibility.

Note 2

Cross-validation: the generalization error (test error) results from using a ML model to predict the response on an observation that was not used in training the model. This is trivial if a labelled test set is available. However, this is usually not the case in practice. The training error can be easily calculated by applying the ML model to the observations used for the model training, but this can dramatically underestimate the test error [3][7]. In the absence of a large labelled test set that can be used to directly estimate the test error rate, an alternative method is to hold out a subset of the training observations from the fitting process, ignore the labels, and then apply the ML model to those held out observations in order to classify them. K-fold CV is one such method, which randomly (sampling without replacement) divides the set of observations into K disjoint groups or folds (e.g. K=10) of approximately equal size. One of the folds is treated as a test set, and the model is fit on the remaining K-1 folds. The test error of the fitted model is calculated when predicting the k-th part (test set) of the data. This procedure is repeated for k = 1, 2, ... , K and the K estimates of test error are finally combined to produce the cross-validation estimate of the test error. It has been shown empirically that using K = 5 or K = 10, yields test error rate estimates with low bias and variance [4]. Here, the 10-fold CV, without stratification (species proportion in the CV groups might be unbalanced), is used for each ML model.

3.1 ML Training

To train and evaluate the performance of several ML methods, the labelled data set consisting of the 819 observations together with the variables (morphological and molecular measurements as continuous variables) as well as the species assignment information (as a categorical variable with three levels), was used. Specifically, seven ML methods have been evaluated: K-Nearest Neighbor (KNN), Logistic Regression for Multiple Classes (LRmc), Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Recursive Partition and Regression Trees (rpart), Random Forests (RF) and Neural Networks (NN). The following eight morphological variables were used: basal shape of the lamina (BS), lamina shape or obversity (OB), petiole ratio (PR), lobe depth ratio at first lobe (LDR), lobe depth ratio at widest lobe (LDRW), percentage of venation (PV), lobe width ratio (LWR), lobe number ratio (LNR). As molecular variable, the first six principal coordinates components (PCo1 to PCo6) were used. Three sets of variables were used for training the ML models: the morphological, the molecular and the combined morphological-molecular variables. 10-fold CV together with the misclassification error rate on the test data sets were used to assess the accuracy of the ML models. The results of the ML models using the combined morphological and molecular variable set are described in detail because of the overall smaller misclassification rates. A summary of the 10-fold CV misclassification rates for each model and each of the three variable sets used (morphological, molecular and combined) are presented in Table 1.

3.1.1 K-Nearest Neighbor (KNN)

One of the simplest ML method is the K-Nearest Neighbor algorithm, which does not require a model to be fit [3][4][7]. Given a query point, the K training points, which are closest in distance (e.g. Euclidean) to the query point are first identified. Then the classification of the query point is done based on the majority vote among the K neighbors with ties being broken at random. To fit a KNN model in R, the *knn()* function implemented in the *class* package can be used (see Appendix 1). The KNN model with the combined variable set (morphological and molecular) was applied to the oak trees data set. Because the choice of K impacts significantly the result of the KNN classification, four values (K=1 to 4) were tested. Regardless of the K values, the misclassification affected mostly the *Q. petraea* species, as expected based on the lowest representation of this species in the data set (Figure 2B). The most optimal value of K appears to be K=4, yielding the smallest 10-fold CV misclassification rate of 16.24 %. Despite its simplicity, flexibility and relatively good performance, the KNN method does not provide information on which predictors are important.

3.1.2 Logistic Regression for Multiple Classes (LRmc)

Logistic regression for multiple classes (multinomial regression) is a method that generalizes the logistic regression, by considering one reference class against all remaining classes [3][4][7]. It predicts the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables. One of the assumption of LR for multiple classes is that there is a low correlation between the predictors (Figure 2E), however it does not make a Gaussian assumption for the class distributions. Parameters fitting is usually done by maximum-likelihood, using the conditional likelihood of a class given the predictors and the multinomial distribution [3][4]. The LR for multiple classes can be fitted in R using the *multinom()* function implemented in the *nnet* package (Appendix 1). The LRmc model with the combined variable set (morphological and molecular) has a 10-fold CV misclassification rate of 12.09 %, with the *Q. petraea* species having most of the observation wrongly classified. In practice, logistic regression for multiple classes is not used all that often, mainly because of the more appropriate discriminant methods, described next.

3.1.3 Linear Discriminant Analysis (LDA)

The linear discriminant analysis is a method that finds a linear combination of predictors that separates two or more classes of observations [3][4][7]. LDA models the conditional class densities as multivariate Gaussian distributions, where each class has its own mean but shares a common covariance matrix. This conditional distribution can be used as the *a-posteriori* distribution of the response variable (the class or species) given the predictors by using the *a-priori* distribution for the response. In practice the parameters of the Gaussian distributions are not known, and they need to be estimated based on the training data. As such, the training data set is used to estimate the class priors (the proportion of instances of certain class or species), the class means (the empirical sample class means) and the covariance matrix (the empirical sample class covariance matrix). Next, the *a-posteriori* distribution (the probability of an observation to belong to a specific class or species) is calculated. Finally, an observation is assigned to the class with the highest probability. This is a linear discriminant classifier because the estimated decision functions are linear in the predictor variables [3][4][7]. LDA can be fitted in R using the *lda()* function implemented in the *MASS* package. The LDA model with the combined variable set (morphological and molecular) applied to the oak trees data set yields the smallest 10-fold CV misclassification rate of 9.89 %. Notably, the classification of the *Q. petraea* observations is improved, albeit still not satisfactory. Less than half of the observations are correctly assigned to this species (41.9 %), however this is an improvement of 20 % compared to the LRmc model. LDA is considered to be more stable than LR for multiple classes when there are more than two classes and when the number of observations is small [3][4]. Given the relatively small number of observations (819) and the underrepresentation of the *Q. petraea* species (Figure 1 and Figure

2B), LDA appears to be a suitable classification model for this data set, supported by the good performance of accurately predicting the species in 90 % of the cases.

3.1.4 Quadratic Discriminant Analysis (QDA)

The quadratic discriminant analysis is a generalization of the LDA and it assumes that the observations in each class (the species) are drawn from a Gaussian distribution but that each class has its own covariance matrix. The estimates for QDA are similar to those for LDA, except for the covariance matrices, which must be estimated separately for each class [3][4][7]. In R, the QDA can be fitted with the *qda()* function implemented in the *MASS* package. The QDA model with the combined variable set (morphological and molecular) has a 10-fold CV misclassification rate of 12.33 %, with a poor classification of the *Q. petraea* observations (90 % of the observations are misclassified). QDA is known to underperform compared to LDA if there are many predictors and relatively few training observations. That is because QDA assumes class specific covariance matrices and hence, it estimates a higher number of parameters compared to LDA [7].

3.1.5 Neural Networks (NN)

Neural networks are two stage methods, relying on extracting linear combinations of the predictors as derived features, and then modelling the response as a nonlinear function of these features [3][4][7]. For the K-class classification with a feed-forward neural network with a single hidden layer, there are K units at the top layer, with the k-th unit modelling the probability of class k. There are K response measurements, each being coded as a 0/1 variable for the k-th class. The predictors are at the bottom layer. The hidden layer consists of the derived features, which are created from linear combinations of the predictors [3][4][7]. The response (class or species assignment) is modelled as a function of linear combinations of the derived features. The unknown parameters of the neural network, often called weights, are estimated by maximum likelihood. Feedforward neural networks can be fitted in R with the function *nnet()* from the *nnet* package. It is important that the variable values are centered and scaled (see Data Preprocessing) to avoid that gradient methods for optimizing the likelihood get stuck in the at regions of the sigmoid functions [7]. The NN model with the combined variable set (morphological and molecular) yields a 10-fold CV misclassification rate of 10.5 %. Most of the observation assigned to *Q. petraea* are misclassified.

3.1.6 Tree-based Classification Methods

Tree-based methods partition the predictors space into a set of disjoint regions (rectangles) and then fit a model for each region. A classification tree predicts that within regions, each observation belongs to the most commonly occurring class (species) among the observations of that region [3][4][7]. A classification tree is grown by recursive binary splitting. As a criterion for determining the binary splits, the misclassification rate could be used (the fraction of the training observations in that region that do not belong to the most common class). However, the classification error is not sufficiently sensitive for tree-growing, and in practice the Gini index is used [3][4][7]. The Gini index is a measure of total variance across all classes and as such, a measure of node purity (a small value indicates that a node contains predominantly observations from a single class). Based on the best split, the data is partitioned in two regions and the splitting process is repeated on each of the two regions. This process is performed on all of the resulting regions (recursive binary splitting). Once a tree is grown, its optimal size should be adaptively chosen based on the data. A recommended option is to first, grow a large tree, then stop the splitting process once some minimum node size is reached (e.g., 5, the number of observations in a node) and then prune (backward elimination) the large tree by using cost-complexity pruning approach [3][4][7].

3.1.6.1 Recursive Partitioning and Regression Trees (rpart) for Classification

The function *rpart()* implemented in the *rpart* package can be used in R to fit recursive partitioning classification trees. The performance of the rpart model with the combined variable set (morphological and molecular) is poor, especially related to the *Q. petraea* species where all observations are misclassified. Although tree-based classification methods are simple and offer intuitive interpretation, they typically are inferior in terms of prediction accuracy to other supervised learning approaches (e.g. LDA). Furthermore, the greedy tree-type algorithm may produce unstable splits. As such, if one of the first splits is wrong, all subsequent splits will be wrong [3][4][7]. However, by aggregating many decision trees, as implemented in the random forests ML method, the predictive performance of trees can be substantially improved.

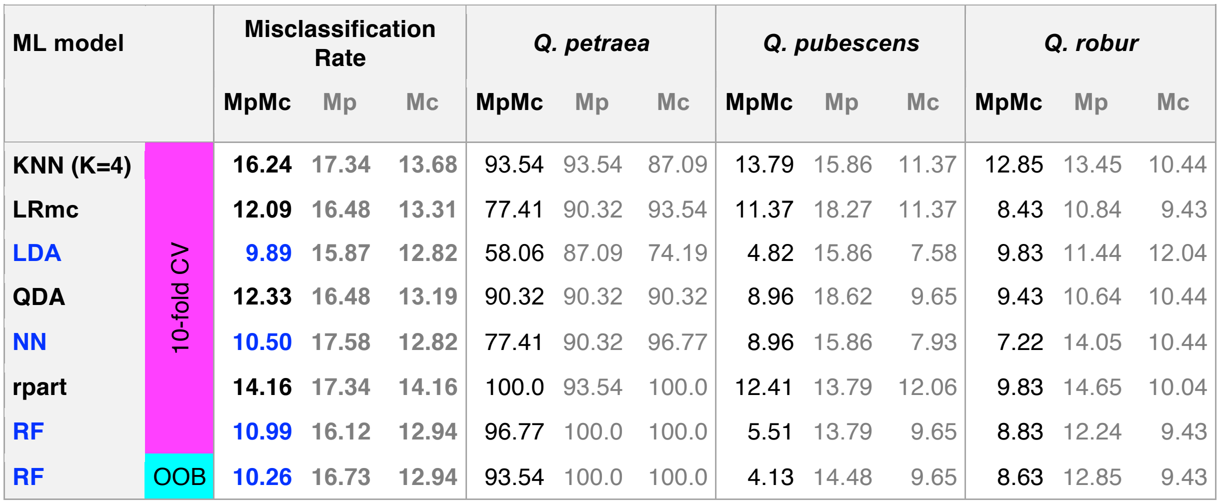
3.1.6.2 Random Forests (RF)

Random forests is a powerful and stable algorithm that builds a certain number of decision trees based on bootstrapped training samples [3][4][7]. Each time a split in a tree is considered, a random sample of *m* predictors is chosen as split candidates from the full set of the *p* predictors. The split is allowed to use only one of those *m* predictors. A fresh sample of *m* predictors is taken at each split, and typically the number of predictors considered at each split is approximately equal to the square root of the total number of predictors. Finally, the prediction of the class for the new data is done by aggregating the predictions over all grown trees (majority votes). To estimate the misclassification rate from the training data, at each bootstrap iteration the class of the observations not included in the bootstrap sample (out-of-bag data, OOB) is predicted using the tree grown with the bootstrap sample. Next, the OOB predictions are aggregated and the overall OOB misclassification rate is computed [3][4][7]. The *randomForest()* function implemented in the *randomForest* package can be used to fit a RF model and it optionally provides a measure of the importance of the predictor variables. The RF model with the combined variable set (morphological and molecular) has a 10-fold CV misclassification rate of 10.99 % and a OOB misclassification rate of 10.26 %. Similar to the rpart model, most of the observations assigned to *Q. petraea* based on the RF model are misclassified. However, the RF model performs better than rpart model in predicting the *Q. pubescens* species with an 8 % increase in prediction accuracy based on the misclassification rate.

3.2 ML Model Evaluation

Overall, the seven evaluated ML models considering the combined variable sets (morphological and molecular) perform better in terms of the 10-fold CV (and OOB for the RF model) misclassification rate compared to the models using either the morphological or the molecular variable set separately (Table 1).

Table 1. The performance summary of the seven ML models using each of the three variable sets (Mp: morphological, Mc: molecular and MpMc: combined). The 10-fold CV (and OOB for the RF models) misclassification rate and the misclassification rate relative to each class for each species is shown in percentages. The best performing ML models are emphasized in blue.



ML model		Misclassification Rate			*Q. petraea*			*Q. pubescens*			*Q. robur*		
		MpMc	Mp	Mc	**MpMc**	Mp	Mc	**MpMc**	Mp	Mc	**MpMc**	Mp	Mc
KNN (K=4)	10-fold CV	**16.24**	17.34	13.68	93.54	93.54	87.09	13.79	15.86	11.37	12.85	13.45	10.44
LRmc		**12.09**	16.48	13.31	77.41	90.32	93.54	11.37	18.27	11.37	8.43	10.84	9.43
LDA		**9.89**	15.87	12.82	58.06	87.09	74.19	4.82	15.86	7.58	9.83	11.44	12.04
QDA		**12.33**	16.48	13.19	90.32	90.32	90.32	8.96	18.62	9.65	9.43	10.64	10.44
NN		**10.50**	17.58	12.82	77.41	90.32	96.77	8.96	15.86	7.93	7.22	14.05	10.44
rpart		**14.16**	17.34	14.16	100.0	93.54	100.0	12.41	13.79	12.06	9.83	14.65	10.04
RF		**10.99**	16.12	12.94	96.77	100.0	100.0	5.51	13.79	9.65	8.83	12.24	9.43
RF	OOB	**10.26**	16.73	12.94	93.54	100.0	100.0	4.13	14.48	9.65	8.63	12.85	9.43

The LDA model with the combined variable sets (morphological and molecular) performs best, predicting accurately the oak tree species in 90 % of the cases. Furthermore, the misclassification rate for the observations assigned to the *Q. petraea* species is the smallest among all ML models evaluated. Accordingly, the LD1 component delineates *Q. robur* species from *Q. pubescens* and *Q. petraea*, whereas the LD2 component additionally delimits the *Q. petraea* species (Figure 3A).

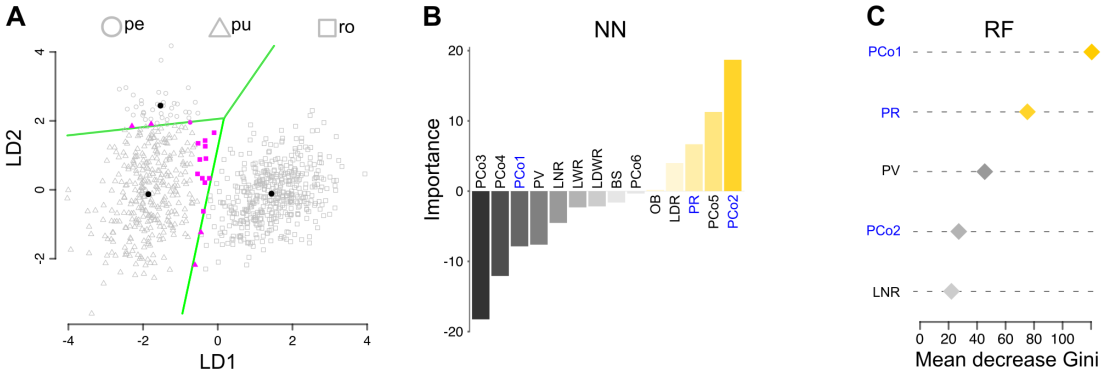


Figure 3. The results of the best three ML models are shown. **A.** The species assignment based on the LDA model with the combined variable sets (morphological and molecular) is shown. Correctly assigned individual observations are displayed in gray and misclassified observations are shown in magenta. The three oak tree species are indicated by the distinct plotting symbols. The LDA model decision boundaries are represented as green lines. The centers (class means) of each species partition are displayed as black circles. **B** and **C**. The predictors importance based on the NN and RF models with the combined variable sets (morphological and molecular), respectively, is shown.

According to the results of the RF model with the combined variable sets (morphological and molecular), the molecular PCo1 predictor and the morphological PR (petiole ratio) and PV (percentage of venation) predictors appear to have the highest importance in classifying the oak tree species (Figure 3C). According to the NN model with the combined variable sets (morphological and molecular) however, PCo1 and PR have only moderate importance, whereas PCo2 and PCo3 are the predictors the influence the most the species classification (Figure 3B). Noteworthy, all ML models result in a high misclassification rate of the observations assigned to the *Q. petraea* species. This is likely related to the underrepresentation of these species in this data set (Figure 2B).

3.3 ML-based Species Prediction for the Unlabelled Observations

The best three ML models according to the 10-fold CV (and OOB) misclassification rate, are the LDA, NN and RF models using the combined variable set (morphological and molecular). These selected ML models were used to predict the species for the unlabelled data set consisting of 550 observations.

The probability of belonging to one of the three oak tree species for each of the LDA, NN and RF models is shown in the Figure 4A.

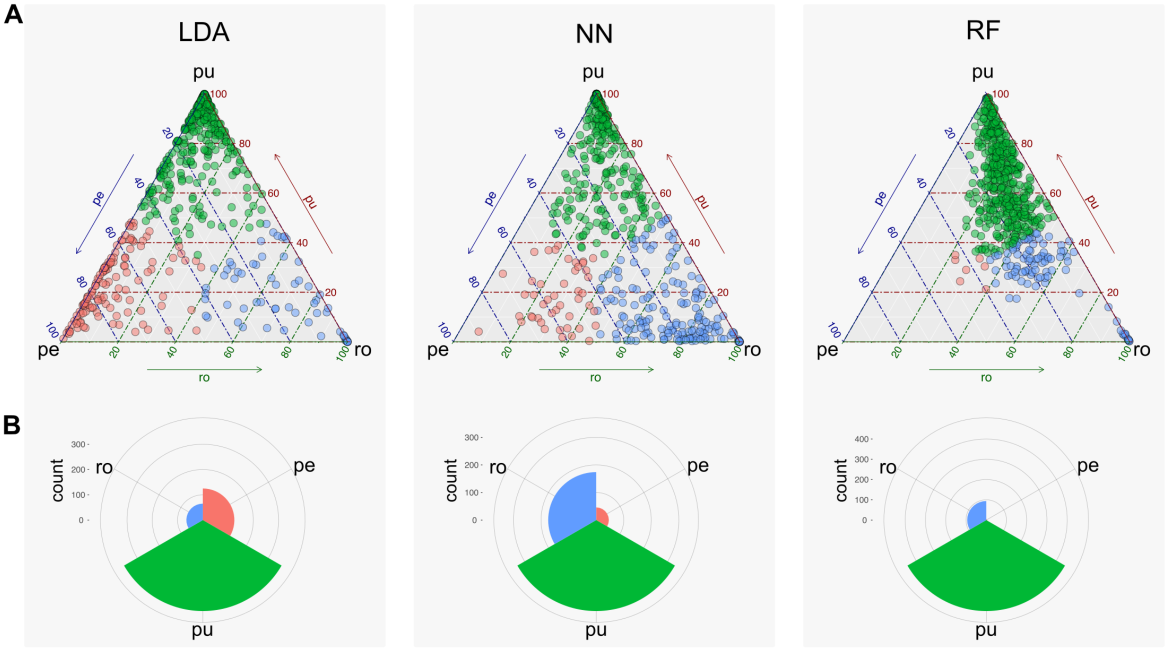


Figure 4. ML-based species prediction for the unlabelled observations. **A.** The ternary plots show the probabilities of the 550 previously unlabelled observations to belong to one of the three species, according to the prediction based on the LDA, NN and RF models with the combined variable set (morphological and molecular). The three oak tree species are represented in distinct colors. Every observation on the ternary plots is displayed as a circle and its location represents a different composition of the probabilities of belonging to one of the three oak tree species. Observations located at the corners of the triangles have high probabilities to belong to the respective species whereas the probability of belonging to one of the two remaining species is low, hence their classification is specific. Observations located near the center of the triangles have similar probability of belonging to either of the three species, hence their classification is ambiguous. **B.** The pie charts show the proportion of the predicted species for the previously unlabelled observations based on the LDA, NN and RF models with the combined variable set (morphological and molecular).

For the LDA model with the combined variable set (morphological and molecular), the *Q. pubescens* species is predicted with the highest probability, whereas the remaining two species (*Q. petraea* and *Q. robur*) show classification probabilities that could either place them to their respective class or misclassify them as *Q. pubescens* species. The classification predicted according to the NN model with the combined variable set (morphological and molecular) is most unstable for the *Q. petraea* species. The RF model with the combined variable set (morphological and molecular) has most likely the lowest predicting capacity. Species classified as *Q. petraea* or *Q. robur* could most likely belong to *Q. pubescens* species according to the small differences in their classification probabilities.

The LDA model with the combined variable set (morphological and molecular) predicts that most of the observation belong to the *Q. pubescens* species (360), followed by the *Q. petraea* species with 125 and *Q. robur* with 65 individuals, respectively (Figure 4B). Similar to the LDA model, the NN model with the combined variable set (morphological and molecular) classifies the majority of the unlabelled observations as *Q. pubescens* (330), followed by 174 trees assigned to *Q. robur* and 46 trees assigned to *Q. petraea*. Consistently, the RF model with the combined variable set (morphological and molecular) classifies the majority of the unlabelled observations as *Q. pubescens* (449), whereas only 94 and 7 observations are assigned to *Q. robur* and *Q. petraea*, respectively (Figure 4B).

Finally, the species composition of the 71 oak tree populations can be inspected in two scenarios: based on the species assignment derived from the trichome data (819 labelled observations) and based on the ML models species predictions (species assignment for the unlabelled 550 observations). Both scenarios are displayed based on their longitude and latitude information, onto the map of Switzerland (Figure 5).

The predictions based on the LDA and NN models with the combined variable set (morphological and molecular) appear to capture best the population composition of the labelled data set. These two models successfully predict observations to belong to the *Q. petraea* species, whereas the RF model with the combined variable set (morphological and molecular) show only modest performance in predicting this underrepresented species (Figure 5). The RF model relies mainly on the PCo1 predictor, and in turn this shows a distinct distribution of values in the data set used to train the ML model compared to the unlabelled data set used for predicting the species. Therefore, the LDA and NN models, for which other predictors in addition to the PCo1 are of importance (Figure 3), are most likely more appropriate models given this data set, for predicting the species assignment for new observations.

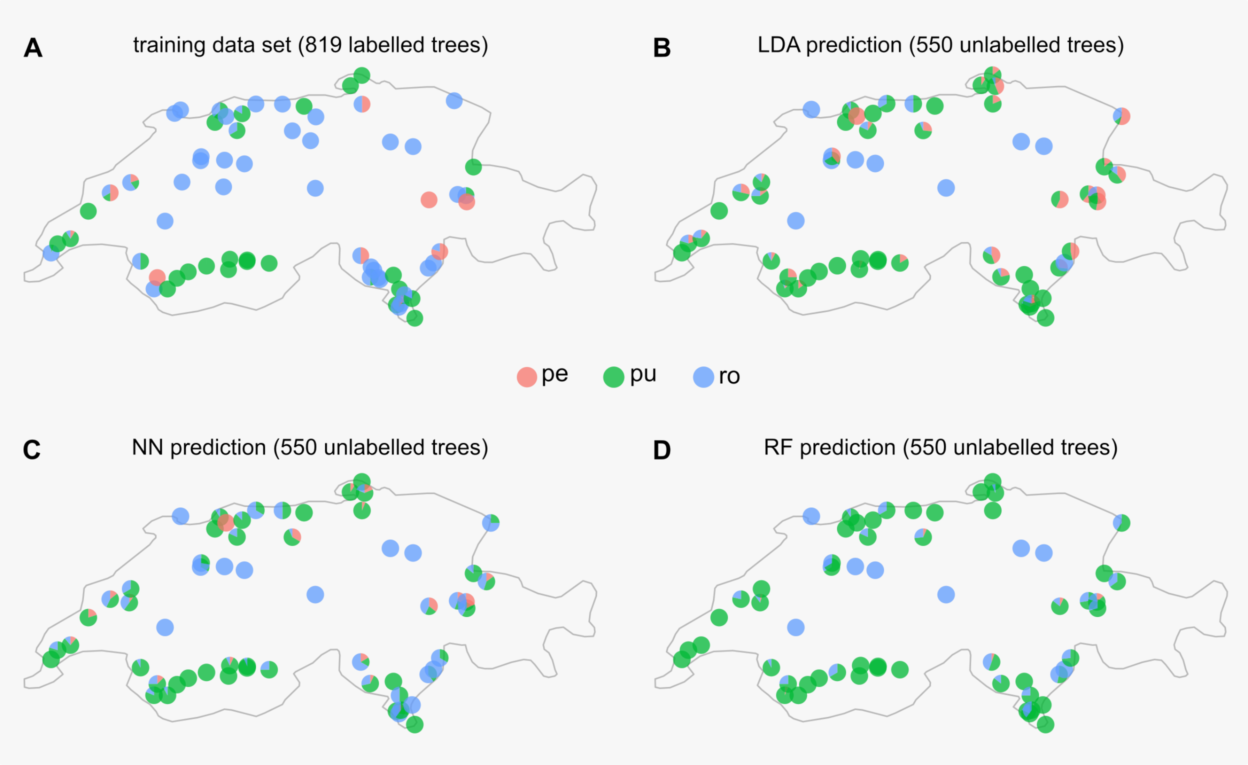

A
training data set (819 labelled trees)
B
LDA prediction (550 unlabelled trees)
pe
pu
ro
C
NN prediction (550 unlabelled trees)
D
RF prediction (550 unlabelled trees)

Figure 5. The species composition and the geographical location of the 71 oak tree populations. The four panels show the species proportions based on the ML-training data set (819 labelled observations) and based on the species predictions for the 550 previously unlabelled observation according to the three best ML models with the combined variable set (morphological and molecular). Each pie chart corresponds to an oak tree population and it displays proportionally the species composition. The three oak tree species are indicated by distinct colors. Pure populations appear in single colors, whereas mixed populations (mixed colors) reflect the proportional species compositions.

4 Conclusion

Two critical observations have emerged during this study, which should be taken into account when interpreting the oak tree species predictions based on the ML models used here. First, assuming the labelling of the training data set based on the trichome profiles, one of the species, *Q. petraea*, is underrepresented (Figure 2B). This could further lead to underrepresentation of individual observations of this species in the 10-fold CV blocks. A possibility to overcome this is to stratify the three species in the CV sampling, insuring a more balanced species representation across the CV blocks. Second, the training data set (labelled observations) does not appear to comprehensively capture the profile of the unlabelled data set, for which predictions should be made (Figure 2C). This could lead to predicting classes based on features of the predictors for which the ML models have not been trained. The second aspect cannot be overcome analytically and it is rather likely due to the experimental sampling setup. For example, the labelled and unlabelled data sets contains each of a subset of the 71 oak trees populations which are overrepresented in only one of the data sets (Supplementary Figure 3). Future ML analyses of such data sets would benefit from a balanced class distribution and from a comparable distribution of the predictors values in the labelled training data set and in the unlabelled data set used for predictions.

Overall, the findings of this study are in line with the results reported by Rellstab et al. [1] and support the main fact that molecular markers have a potential to assist and possibly replace the more difficult to obtain and evaluate morphological predictors for accurate species assignment in hybridising European white oak trees. The most accurate ML models in this study are those considering a combination of the morphological and molecular variables, predicting correctly the species in up to 90 % of the cases. In conclusion, I show here that ML techniques can in principle be useful in identifying oak tree species based on molecular and morphological characters. Specifically, should the leaf trichome data be available for individual oak trees with unresolved species assignment, either an LDA or NN algorithm can be used to reliably assign the species these oak trees belong to.

References

1. **Rellstab C, Buehler A, Graf R, Folly C, Gugerli F. (2016).** Using joint multivariate analyses of leaf morphology and molecular-genetic markers for taxon identification in three hybridizing European white oak species (Quercus spp.). *Annals of Forest Science*; 73:669–79. doi:10.1007/s13595-016-0552-7.

2. **Kremer A, Dupouey JL, Deans JD, *et al.* (2002).** Leaf morphological differentiation between *Quercus robur* and *Quercus petraea* is stable across western European mixed oak stands. *Annals of Forest Science*; 59:777–87. doi:10.1051/forest:2002065.

3. **James G, Witten D, Hastie T, Tibshirani R. (2013).** An Introduction to Statistical Learning. *New York, NY: Springer New York*; doi:10.1007/978-1-4614-7138-7.

4. **Hastie T, Tibshirani R, Friedman JH (2009).** The elements of statistical learning data mining, inference, and prediction. *New York, NY: Springer New York;* doi:10.1007/978-0-387-84858-7.

5. **R Core Team (2017).** R: A Language and Environment for Statistical Computing.

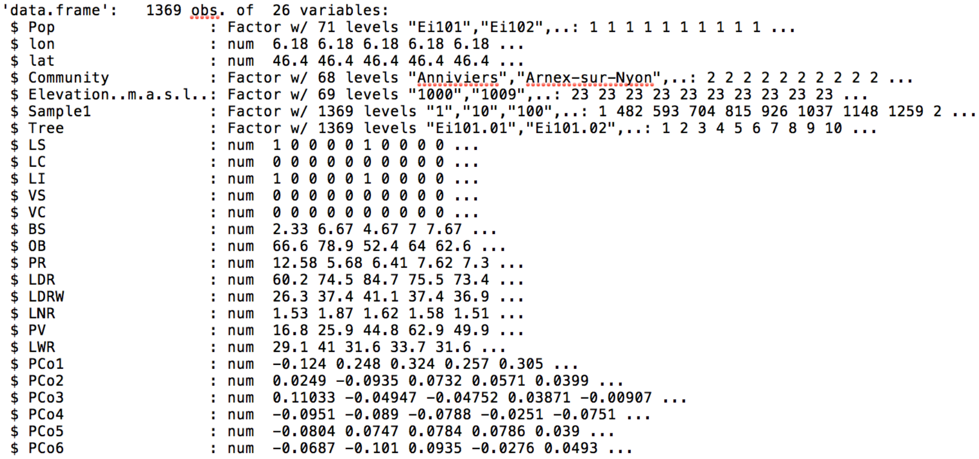
6. **RStudio Team (2015).** RStudio: Integrated Development for R. RStudio, Inc., Boston, MA URL http://www.rstudio.com/.

7. **Bühlmann P, Mächler M. (2008).** Computational Statistics. http://stat.ethz.ch/education/semesters/ss2012/CompStat/sk.pdf.

Supplementary Material

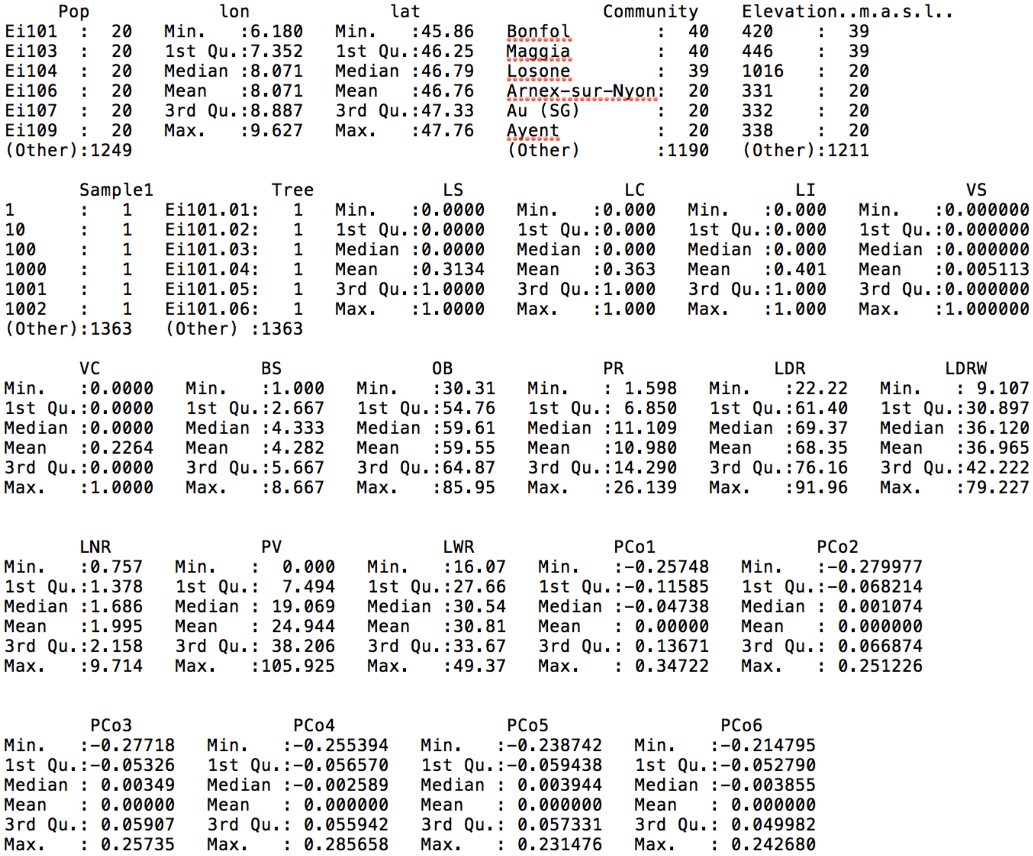
Supplementary Table 1. Summary of the original data set. **A.** The data structure: name and type of variables. **B.** Summary statistics of the variables.

A.



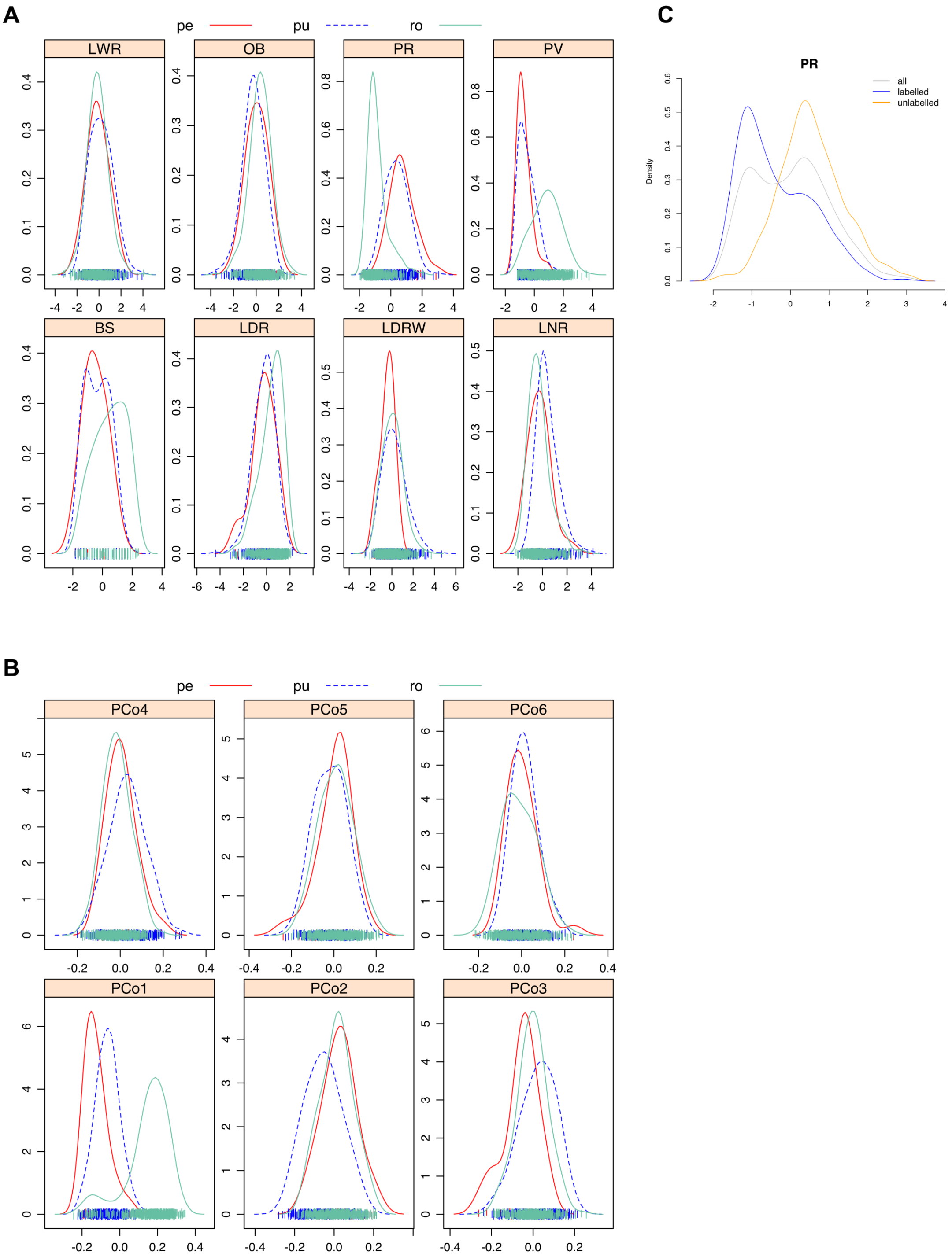
```
'data.frame':	1369 obs. of  26 variables:
 $ Pop                 : Factor w/ 71 levels "Ei101","Ei102",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ lon                 : num  6.18 6.18 6.18 6.18 6.18 ...
 $ lat                 : num  46.4 46.4 46.4 46.4 46.4 ...
 $ Community           : Factor w/ 68 levels "Anniviers","Arnex-sur-Nyon",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ Elevation..m.a.s.l..: Factor w/ 69 levels "1000","1009",..: 23 23 23 23 23 23 23 23 23 23 ...
 $ Sample1             : Factor w/ 1369 levels "1","10","100",..: 1 482 593 704 815 926 1037 1148 1259 2 ...
 $ Tree                : Factor w/ 1369 levels "Ei101.01","Ei101.02",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ LS                  : num  1 0 0 0 0 1 0 0 0 0 ...
 $ LC                  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ LI                  : num  1 0 0 0 0 1 0 0 0 0 ...
 $ VS                  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ VC                  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ BS                  : num  2.33 6.67 4.67 7 7.67 ...
 $ OB                  : num  66.6 78.9 52.4 64 62.6 ...
 $ PR                  : num  12.58 5.68 6.41 7.62 7.3 ...
 $ LDR                 : num  60.2 74.5 84.7 75.5 73.4 ...
 $ LDRW                : num  26.3 37.4 41.1 37.4 36.9 ...
 $ LNR                 : num  1.53 1.87 1.62 1.58 1.51 ...
 $ PV                  : num  16.8 25.9 44.8 62.9 49.9 ...
 $ LWR                 : num  29.1 41 31.6 33.7 31.6 ...
 $ PCo1                : num  -0.124 0.248 0.324 0.257 0.305 ...
 $ PCo2                : num  0.0249 -0.0935 0.0732 0.0571 0.0399 ...
 $ PCo3                : num  0.11033 -0.04947 -0.04752 0.03871 -0.00907 ...
 $ PCo4                : num  -0.0951 -0.089 -0.0788 -0.0251 -0.0751 ...
 $ PCo5                : num  -0.0804 0.0747 0.0784 0.0786 0.039 ...
 $ PCo6                : num  -0.0687 -0.101 0.0935 -0.0276 0.0493 ...
```

B.

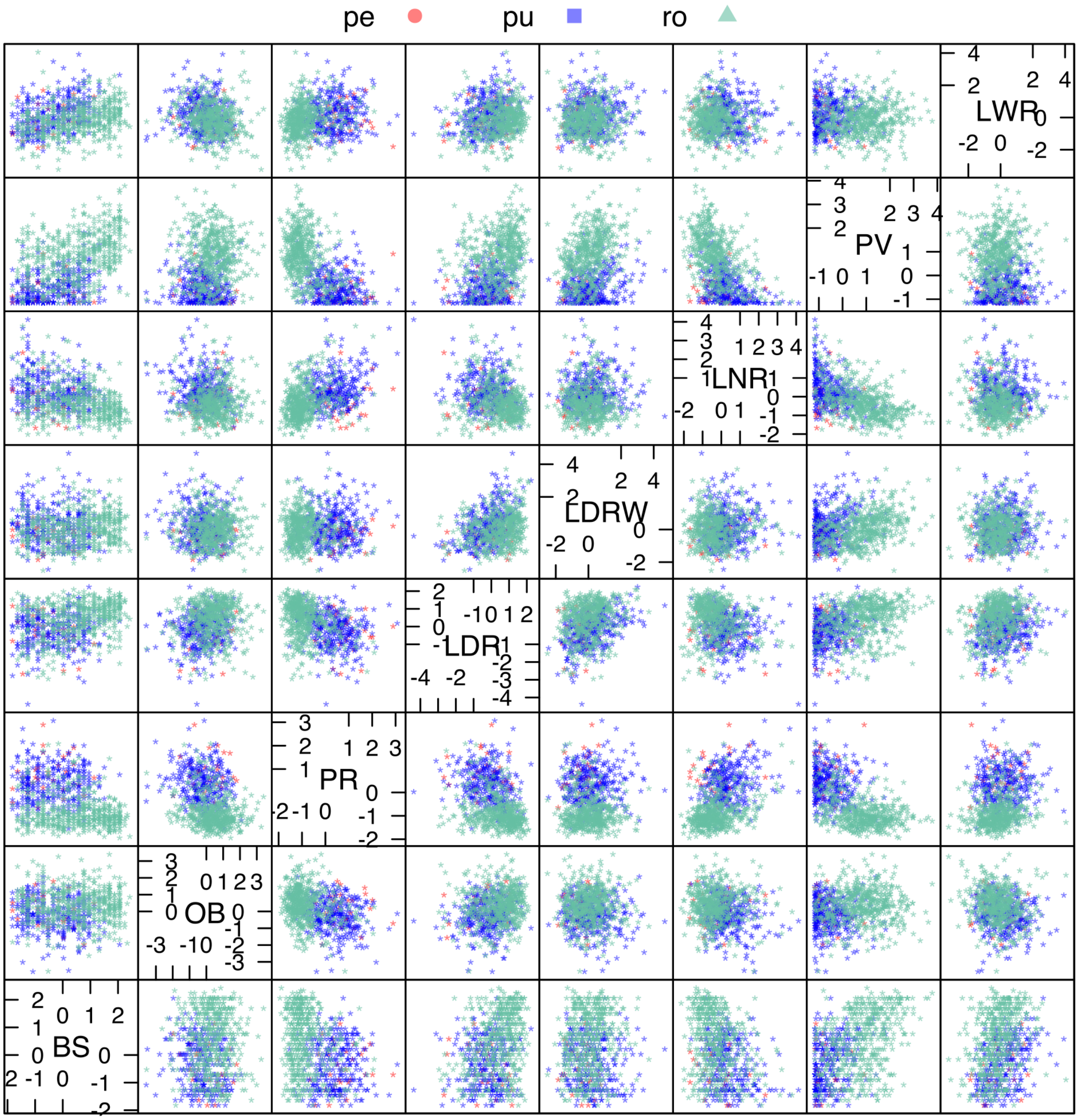


```
      Pop            lon             lat                     Community    Elevation..m.a.s.l..
 Ei101  :  20   Min.   :6.180   Min.   :45.86   Bonfol        :  40   420    :  39            
 Ei103  :  20   1st Qu.:7.352   1st Qu.:46.25   Maggia        :  40   446    :  39            
 Ei104  :  20   Median :8.071   Median :46.79   Losone        :  39   1016   :  20            
 Ei106  :  20   Mean   :8.071   Mean   :46.76   Arnex-sur-Nyon:  20   331    :  20            
 Ei107  :  20   3rd Qu.:8.887   3rd Qu.:47.33   Au (SG)       :  20   332    :  20            
 Ei109  :  20   Max.   :9.627   Max.   :47.76   Ayent         :  20   338    :  20            
 (Other):1249                                   (Other)       :1190   (Other):1211            

    Sample1           Tree            LS               LC              LI               VS          
 1      :   1   Ei101.01:   1   Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   :0.000000  
 10     :   1   Ei101.02:   1   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000000  
 100    :   1   Ei101.03:   1   Median :0.0000   Median :0.000   Median :0.000   Median :0.000000  
 1000   :   1   Ei101.04:   1   Mean   :0.3134   Mean   :0.363   Mean   :0.401   Mean   :0.005113  
 1001   :   1   Ei101.05:   1   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:0.000000  
 1002   :   1   Ei101.06:   1   Max.   :1.0000   Max.   :1.000   Max.   :1.000   Max.   :1.000000  
 (Other):1363   (Other) :1363                                                                       

       VC               BS              OB              PR              LDR             LDRW       
 Min.   :0.0000   Min.   :1.000   Min.   :30.31   Min.   : 1.598   Min.   :22.22   Min.   : 9.107  
 1st Qu.:0.0000   1st Qu.:2.667   1st Qu.:54.76   1st Qu.: 6.850   1st Qu.:61.40   1st Qu.:30.897  
 Median :0.0000   Median :4.333   Median :59.61   Median :11.109   Median :69.37   Median :36.120  
 Mean   :0.2264   Mean   :4.282   Mean   :59.55   Mean   :10.980   Mean   :68.35   Mean   :36.965  
 3rd Qu.:0.0000   3rd Qu.:5.667   3rd Qu.:64.87   3rd Qu.:14.290   3rd Qu.:76.16   3rd Qu.:42.222  
 Max.   :1.0000   Max.   :8.667   Max.   :85.95   Max.   :26.139   Max.   :91.96   Max.   :79.227  


      LNR              PV               LWR             PCo1               PCo2          
 Min.   :0.757   Min.   :  0.000   Min.   :16.07   Min.   :-0.25748   Min.   :-0.279977  
 1st Qu.:1.378   1st Qu.:  7.494   1st Qu.:27.66   1st Qu.:-0.11585   1st Qu.:-0.068214  
 Median :1.686   Median : 19.069   Median :30.54   Median :-0.04738   Median : 0.001074  
 Mean   :1.995   Mean   : 24.944   Mean   :30.81   Mean   : 0.00000   Mean   : 0.000000  
 3rd Qu.:2.158   3rd Qu.: 38.206   3rd Qu.:33.67   3rd Qu.: 0.13671   3rd Qu.: 0.066874  
 Max.   :9.714   Max.   :105.925   Max.   :49.37   Max.   : 0.34722   Max.   : 0.251226  


       PCo3               PCo4                PCo5                PCo6          
 Min.   :-0.27718   Min.   :-0.255394   Min.   :-0.238742   Min.   :-0.214795  
 1st Qu.:-0.05326   1st Qu.:-0.056570   1st Qu.:-0.059438   1st Qu.:-0.052790  
 Median : 0.00349   Median :-0.002589   Median : 0.003944   Median :-0.003855  
 Mean   : 0.00000   Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.000000  
 3rd Qu.: 0.05907   3rd Qu.: 0.055942   3rd Qu.: 0.057331   3rd Qu.: 0.049982  
 Max.   : 0.25735   Max.   : 0.285658   Max.   : 0.231476   Max.   : 0.242680  
```

Supplementary Figure 1. The distribution of the morphological **(A)** and molecular **(B)** variable values in the labelled data set (819 observations). The distinct distribution of the PR morphological variable in the labelled, unlabelled and complete data set is shown **(C)**.

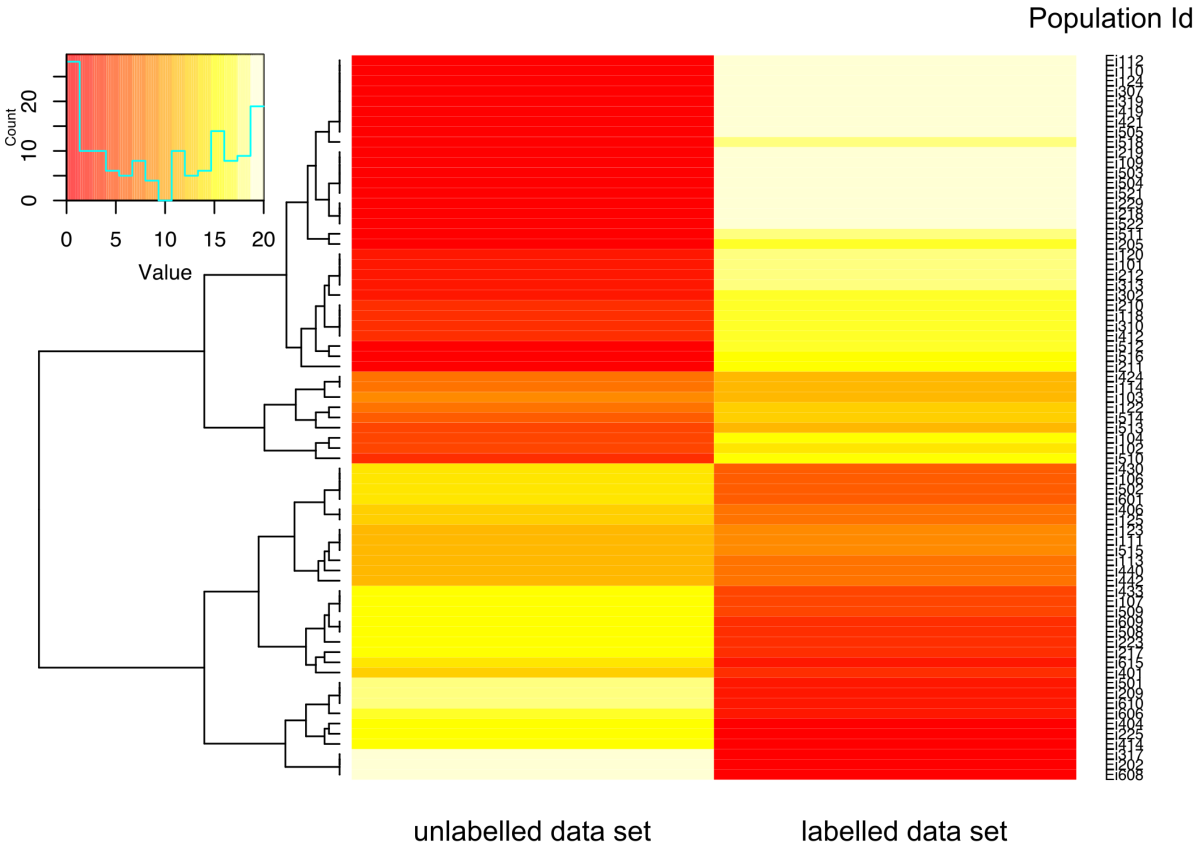


Supplementary Figure 2. The scatter plot matrix of the morphological predictors. The three oak tree species are displayed in distinct colors. Several morphological variables appear to have an influence on the oak tree species delineation (e.g., PR and BS).



Supplementary Figure 3.

The distinct representation of the 71 oak tree populations in the labelled and unlabelled data sets is displayed as heatmap. Due to the field sampling experimental design, each population consisted of 20 individual oak trees. The number of observations (individual oak trees) for each population in the labelled and unlabelled data set is indicated by the color scale: red no or low number of observations (0 to 5) and yellow for moderate to high number of observations (15 to 20). Some population are absent from either of the labelled or unlabelled data sets (red sectors in the heatmap).



Appendix 1. The R script for the data analysis presented in this thesis.

Install and load required R packages:

```
ipak <- function(pkg){
  new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
  if (length(new.pkg))
    install.packages(new.pkg, dependencies = TRUE)
  sapply(pkg, require, character.only = TRUE)
}

packages <- c("ggplot2","ggmap","rgdal","rgeos","maptools","dplyr","tidyr","k
nitr","tmap",
              "maps","caret","SparseM","xlsx","AppliedPredictiveModeling","cl
ass","nnet",
              "rpart","rpart.plot","data.table","timeDate","lubridate","rando
mForest","MASS","ranger",
              "ggtern","qmap","ggforce","scatterpie","RColorBrewer","corrplot
","klaR","NeuralNetTools","gplots")

ipak(packages)

lapply(packages, library,character.only=TRUE)
```

Data preprocessing

The Latitude and Longitude columns are swapped in the original file (verified on map). This is addressed below.

```
q.param <- read.xlsx2("/.../quercus_parameters.xlsx",1)
q.param[,4:22] <- apply(q.param[,4:22],2,function(x)as.numeric(as.character(x
)))

q.geo <- read.xlsx2("/.../quercus_populations.xlsx",1)
q.geo$Population <- factor(paste0("Ei",q.geo$Population))
colnames(q.geo)[1:3] <- c("Pop","lon","lat")
q.geo$lon <- as.numeric(as.character(q.geo$lon))
q.geo$lat <- as.numeric(as.character(q.geo$lat))

q.dat.full <- merge(q.geo,q.param,by="Pop")
```

Exploratory data analysis

The variables used for defining species assignment rules are excluded

```
Lfeat <- c("BS","OB","PR","LDR","LDRW","LNR","PV","LWR")
PCload <- paste0("PCo",1:6)
GEOloc <- "geo"
Elev <- "Elevation..m.a.s.l.."

# plot the predictor values ranges (full data set of 1,369 observations)
svg("/.../Plot.svg")

par(mfrow=c(2,3),oma=c(0,0,2,0))
for (i in colnames(q.dat.full[,PCload])){
  boxplot(q.dat.full[,i],main=i)
}
title("Full data set 1,369 obs", outer=TRUE)

par(mfrow=c(2,4),oma=c(0,0,2,0))
for (i in colnames(q.dat.full[,Lfeat])){
  boxplot(q.dat.full[,i],main=i)
}
title("Full data set 1,369 obs", outer=TRUE)
dev.off()

# data transformation
q.dat.full$LNR <- log2(q.dat.full$LNR)

# data scaling; only Lfeat
colnames(q.dat.full[,Lfeat])
q.dat.full[,Lfeat] <- scale(q.dat.full[ ,Lfeat])
```

Define the classes or species based on trichome profiles + exclusively stellate hair on the lamina: LS/LC/LI=1/0/0 --> *Q. petraea* + exclusively clustered hair on the lamina: LS/LC/LI=0/1/0 --> *Q. pubescens* + no hair on the lamina: LS/LC/LI=0/0/0 --> *Q. robur*

```
pe <- paste0(1,0,0)
pu <- paste0(0,1,0)
ro <- paste0(0,0,0)

for (i in 1:nrow(q.dat.full)){
  q.dat.full[i,"hairP"] <- paste(unlist(q.dat.full[i,c("LS","LC","LI")]),coll
apse="")
}
```

```
q.dat.full$Species <- ifelse(q.dat.full$hairP == pe,"pe",
                        ifelse(q.dat.full$hairP == pu,"pu",
                          ifelse(q.dat.full$hairP == ro,"ro","Undef")))

table(q.dat.full$Species)

# define distribution of the trees into geographical regions (e.g. N-W, S-W,
N-E and S-E)
# limits are chosen basedon visual inspection of the plot displaying species
in longitude vs. latitude space

q.dat.full$geolon <- cut(q.dat.full$lon, c(-Inf, 8.071, Inf),
                                 labels=c("West", "East"))

q.dat.full$geolat <- cut(q.dat.full$lat, c(-Inf, 46.6964, Inf),
                                 labels=c("South", "North"))
table(q.dat.full$geolon)
table(q.dat.full$geolat)

q.dat.full$geo <- paste0(q.dat.full$geolat,"_",q.dat.full$geolon)
table(q.dat.full$geo)
sum(table(q.dat.full$geo))
```

Plot the distribution of predictor values in the scaled, full data set (1,369 observations)

```
svg("/.../Plot.svg")
par(mfrow=c(2,3),oma=c(0,0,2,0))
for (i in colnames(q.dat.full[,PCload])){
  plot(density(q.dat.full[,i]),main=i)
}
title("Scaled full data set 1,369 obs", outer=TRUE)

par(mfrow=c(2,4),oma=c(0,0,2,0))
for (i in colnames(q.dat.full[,Lfeat])){
  plot(density(q.dat.full[,i]),main=i)
}
title("Scaled full data set 1,369 obs", outer=TRUE)
dev.off()
```

Split the data into a labelled data set (for which species information was deduced from the trichome profiles data) and an unlabelled data set.

```
# labelled data set
q.dat <- q.dat.full[q.dat.full$Species != "Undef",]
q.dat$Species <- as.factor(q.dat$Species)
q.dat$geo <- as.factor(q.dat$geo)
table(q.dat$Species)

# unlabelled data set
newd <- q.dat.full[q.dat.full$Species == "Undef",]
```

```
newd$Species <- as.factor(newd$Species)
newd$geo <- as.factor(newd$geo)
```

Verify if there are potential sampling differences for the labelled and unlabelled data sets

```
pop.lb <- as.matrix(table(q.dat$Pop))
pop.unlb <- as.matrix(table(newd$Pop))
pop.lbunlb <- merge(pop.lb,pop.unlb,by="row.names")
rownames(pop.lbunlb) <- pop.lbunlb[,1]
pop.lbunlb[,1] <- NULL
colnames(pop.lbunlb) <- c("pop_lb","pop_unlb")

svg("/.../Plot.svg")
heatmap.2(as.matrix(pop.lbunlb),cexCol = 0.75,trace = "none",cexRow = 0.65)
dev.off()
```

Plot predictor values distribution in the labelled and unlabelled data sets (and in the full data set of 1,369 observations)

```
svg("/.../Plot.svg")
par(mfrow=c(1,1))
plot(density(q.dat.full$PCo1),col="gray",ylim=c(0,6.5),main="PCo1",cex.main=0
.65,xlab=NA)
lines(density(q.dat$PCo1),col="blue")
lines(density(newd$PCo1),col="orange")
legend(0.05,6,c("all","labelled","unlabelled"),col=c("gray","blue","orange"),
cex=0.5,lty = 1,bty = "n")
dev.off()

svg("/.../Plot.svg")
plot(density(q.dat.full$PR),col="gray",ylim=c(0,0.65),main="PR",cex.main=0.65
,xlab=NA)
lines(density(q.dat$PR),col="blue")
lines(density(newd$PR),col="orange")
legend(0.5,0.6,c("all","labelled","unlabelled"),col=c("gray","blue","orange")
,cex=0.5,lty = 1,bty = "n")
dev.off()
```

Plot the species proportion in the labelled data set, the scatter plot matrices for the predictors and the predictors correlogram

```
# species proportion
svg("/.../Plot.svg")
print(ggplot(q.dat, aes(x = Species,fill=Species)) +
  geom_bar(width = 1) +
  coord_polar())
dev.off()

# PCo1,2,3 scatter plot matrix
svg("/.../Plot.svg")
transparentTheme(trans = .5)
```

```
featurePlot(x = q.dat[,PCload[1:3]],
            y = q.dat$Species,
            plot = "pairs",pch="*",
            auto.key = list(columns = 3))
dev.off()

# morphological predictors scatter plot matrix
svg("/.../Plot.svg")
transparentTheme(trans = .5)
featurePlot(x = q.dat[,Lfeat],
            y = q.dat$Species,
            plot = "pairs",pch="*",
            auto.key = list(columns = 3))
dev.off()

# molecular predictor values distribution
svg("/.../Plot.svg")
transparentTheme(trans = .9)
featurePlot(x = q.dat[,PCload],
            y = q.dat$Species,
            plot = "density",
            scales = list(x = list(relation="free"),
                          y = list(relation="free")),
            adjust = 1.5,
            pch = "|",
            layout = c(3, 2),
            auto.key = list(columns = 3))
dev.off()

# morphological predictor values distribution
svg("/.../Plot.svg")
transparentTheme(trans = .9)
featurePlot(x = q.dat[,Lfeat],
            y = q.dat$Species,
            plot = "density",
            scales = list(x = list(relation="free"),
                          y = list(relation="free")),
            adjust = 1.5,
            pch = "|",
            layout = c(4, 2),
            auto.key = list(columns = 3))
dev.off()
```

Machine learning

```
# use the labelled data set (819 observations) to train and test the ML model
s
q.dat.s <- subset(q.dat,select=c(c(PCload,Lfeat),"Species"))
table(q.dat.s$Species)

# plot the predictors correlations
p.corr <- cor(q.dat.s[,-15])
col1 <- colorRampPalette(c("#67001F", "#B2182B", "#D6604D", "#F4A582", "#FDDB
C7",
            "#FFFFFF", "#D1E5F0", "#92C5DE", "#4393C3", "#2166AC", "#053061")
)

svg(".../Plot.svg")
corrplot(p.corr,method="ellipse",type="lower", diag = FALSE, col=col1(35),tl.
cex = 0.75,tl.col = "gray")
dev.off()

# ML with Cross-validation
attach(q.dat.s)
source("http://stat.ethz.ch/Teaching/WBL/Source-WBL-5/03.RCodes/dm-serie2.R")

## Funktionen f??r Klassifikationsvergleiche mit CV
# CVtest <- function(fitfn, predfn, data, k = 10, verbose=TRUE, ...)
# {
#   n <- nrow(data)
#   stopifnot(is.numeric(n), n >= 1, 1 <= k, k <= n)
#   ii <- sample(n)
#   res <- numeric(n)
#   j1 <- 1                            ## Start des ersten Blocks
#   if(verbose) cat("fold ")
#   for (i in 1:k) {
#     j2 <- (i*n) %/% k                ## Ende des i-ten Blocks
#     j <- ii[j1:j2]                   ## Indizes der Test-Beob im Fold i
#     fitted.model <- fitfn(data = data[-j,], ...)
#     if(verbose) { cat(i, ""); flush.console() }
#     res[j] <- predfn(fitted.model, newdata = data[j,])
#     j1 <- j2 + 1                     ## Start des (i+1)-ten Blocks
#   }; if(verbose) cat("\n")
#   res
# }
#
# con <- function(...)
# {
#   print(tab <- table(...), zero.print = ".")
#   t0 <- tab ; diag(t0) <- 0
```

```r
#   cat("error rate = ",
#       round(100*sum(t0)/length(list(...)[[1]]), 2), "%\n")
#   invisible(tab)
# }
#


# KNN
q.dat.s.tr <- q.dat.s[,-15]
n <- nrow(q.dat.s.tr[,-15])

set.seed(909)
ii <- sample(n)
res.knn <- numeric(n)
k <- 10  #  k-fold for cross-validation

for(KK in 1:4) { # KK is the number of nearest neighbors
  cat("KNN (K = ", KK,"):\n-----------\n",sep="")
  j1 <- 1                            # Start of the 1^st CV block
  for (i in 1:k) {
    j2 <- (i*n) %/% k                # Endof the i^th CV block
    j <- ii[j1:j2]                   # Indices of the test data set at CV fold 
i
    res.knn[j] <- knn(train = q.dat.s.tr[-j,], test = q.dat.s.tr[j,],
                      cl = q.dat.s$Species[-j], k = KK)
    cat(i, ""); flush.console()
    j1 <- j2 + 1                     # Start of the (i+1)^th CV block
  }; cat("\n")
  con(true = q.dat.s$Species, predicted = res.knn)
}



# Logistic regression for multiple classes
set.seed(909)
res.multinom <- CVtest(function(...) multinom(Species ~ ., ...),
                       function(obj, ...) predict(obj, type = "class", ...),
                       data = q.dat.s, maxit = 1000, trace = FALSE)

con(true = q.dat.s$Species, "CV-predicted" = res.multinom)

# LDA
set.seed(909)
res.lda <- CVtest(function(...) lda(Species ~ ., ...),
                  function(obj, ...) predict(obj, ...)$class,
                  data = q.dat.s)

con(true = q.dat.s$Species, "CV-predicted" = res.lda)
```

```r
r.lda <- lda(Species ~ ., data=q.dat.s)
datPred<-data.frame(Species=predict(r.lda)$class,predict(r.lda)$x)
datPred$pch.s <- ifelse(datPred$Species == "pe",1,
                        ifelse(datPred$Species == "pu",2,
                          ifelse(datPred$Species == "ro",0,NA)))

svg("/.../Plot.svg")
partimat(Species~LD2+LD1,data=datPred,method="lda",
         col.correct="gray",
         col.wrong="orange",
         imageplot=FALSE,
         gs=datPred$pch.s)
dev.off()

# QDA
set.seed(909)
res.qda <- CVtest(function(...) qda(Species ~ ., ...),
                  function(obj, ...) predict(obj, ...)$class,
                  data = q.dat.s)

con(true = q.dat.s$Species, "CV-predicted" = res.qda)

# Neural networks
set.seed(909)
res.nn <- CVtest(function(...)
                 nnet(Species ~ ., size = 6, decay = 0.1, trace=FALSE, ...),
                 function(obj, ...) predict(obj, type = "class", ...),
                 data = q.dat.s, maxit = 500)

con(true = q.dat.s$Species, "CV-predicted" = res.nn)

r.nnet <- nnet(Species ~ ., size = 6, decay = 0.1, trace=FALSE, data=q.dat.s,
maxit=500)

svg("/.../Plot.svg")
olden(r.nnet)
dev.off()

# recursive partition and regression trees (rpart)
set.seed(909)
res.rpart <- CVtest(function(...) rpart(Species ~ ., ...),
                    function(obj, ...) predict(obj, type = "class", ...),
                    data=q.dat.s)

con("true" = q.dat.s$Species, "CV-predicted" = res.rpart)

# Radom Forests
# OOB
set.seed(909)
```

```r
rf.rf <- randomForest(Species ~ ., data=q.dat.s,importance=TRUE,proximity=TRU
E)

rf.rf$confusion
con(true = q.dat.s$Species, predicted = predict(rf.rf))

svg("/.../Plot.svg")
varImpPlot(rf.rf, sort = TRUE, main="Variable Importance", n.var=5,pch=18,col
="gray")
dev.off()

# 10-fold CV
set.seed(909)
res.rf <- CVtest(function(...) randomForest(Species ~ ., ...),
                     function(obj, ...) predict(obj, type = "response", ...),
                     data = q.dat.s)
con(true = q.dat.s$Species, predicted = res.rf)


# ML prediction of the observations in the unlabelled data set based on the b
est three ML models (according to their misclassification rate)

bestML <- r.lda
newPR <- as.data.frame(predict(bestML, newdata=newd)$post)

bestML <- rf.rf
newPR <- as.data.frame(predict(bestML, newdata=newd,type="prob"))

bestML <- r.nnet
newPR <- as.data.frame(predict(bestML, newdata=newd,type="raw"))

SpeciesPR <- character(nrow(newPR))
for (i in 1:nrow(newPR)){
  SpeciesPR[i] <- names(which.max(newPR[i,]))
}

newPR$SpeciesPR <- SpeciesPR
newd$SpeciesPR <- SpeciesPR

table(newPR$SpeciesPR)

# ternary plot
svg("/.../Plot.svg")
tern.p <- ggtern(data = newPR, aes(x = pe, y = pu, z = ro)) +
               geom_point(aes(fill = SpeciesPR),
                          size = 4,
                          shape = 21,
                          color = "black",alpha=0.5) +
               ggtitle("LDA prediction") +
               labs(fill = "Species") +
               theme_rgbg() +
               theme(legend.position      = c(0,1),
                     legend.justification = c(0,1))

print(tern.p)
dev.off()

# Plot thespecies proportion in the predicted (unlabelled) data set (550 obse
rvations)
svg("/.../Plot.svg")
print(ggplot(newd, aes(x = SpeciesPR,fill=SpeciesPR)) +
  geom_bar(width = 1) +
  coord_polar())
dev.off()
```

Plot the ML predictions on the map of Switzerland

```
# Proportion of species (known and ML-predicted) within each of the 71 popula
tions

pops <- as.data.frame.matrix(table(as.character(q.dat$Pop),q.dat$Species))

for (i in row.names(pops)){
  pops[i,"lat"] <- unique(q.dat[which(q.dat[,"Pop"] == i),"lat"])
  pops[i,"long"] <- unique(q.dat[which(q.dat[,"Pop"] == i),"lon"])
}

pops$region <- factor(1:nrow(pops))

# plot the populations and their species proportions (based on the training d
ata set) onto the map
world_map <- map_data('world')
CHmap <- subset(world_map, world_map$region=="Switzerland")

svg("/.../Plot.svg")
p <- ggplot(CHmap, aes(long, lat)) + geom_map(map=CHmap,aes(map_id=region), f
ill=NA,color="black") +
      coord_quickmap()

print(p + geom_scatterpie(aes(x=long, y=lat, group=region), data=pops,cols=c(
"pe","pu","ro"),color=NA, alpha=.75) +
coord_fixed())

dev.off()

# plot the populations and their species proportions (ML-predicted) onto the
map
```

```r
popsPR <- as.data.frame.matrix(table(as.character(newd$Pop),newd$SpeciesPR))

for (i in row.names(popsPR)){
  popsPR[i,"lat"] <- unique(newd[which(newd[,"Pop"] == i),"lat"])
  popsPR[i,"long"] <- unique(newd[which(newd[,"Pop"] == i),"lon"])
}

popsPR$region <- factor(1:nrow(popsPR))

world_map <- map_data('world')
CHmap <- subset(world_map, world_map$region=="Switzerland")

svg("/.../Plot.svg")
p <- ggplot(CHmap, aes(long, lat)) + geom_map(map=CHmap,aes(map_id=region), f
ill=NA,color="black") +
      coord_quickmap()

print(p + geom_scatterpie(aes(x=long, y=lat, group=region), data=popsPR, cols
=c("pe","pu","ro"),color=NA, alpha=.75) +
coord_fixed())
dev.off()
```

R session information

```r
sessionInfo()
```

```
## R version 3.3.3 (2017-03-06)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X Yosemite 10.10.5
##
## locale:
## [1] C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] backports_1.0.5      magrittr_1.5         rprojroot_1.2
##  [4] tools_3.3.3          htmltools_0.3.5      yaml_2.1.14
##  [7] Rcpp_0.12.9          stringi_1.1.2        rmarkdown_1.4.0.9000
## [10] knitr_1.15.1         stringr_1.2.0        digest_0.6.12
## [13] evaluate_0.10
```