Predicting continental origin from genomic information on Chromosome 1

Is Machine Learning racist?

Elena García Lara 2604289[1], **Stavros Giannoukakos 2566579**[1], **Alberto Gil Jimenez 2595259**[1], **and Aslı Küçükosmanoğlu 2506020**[1]

[1]MSc Bioinformatics and Systems Biology. VU University Amsterdam

ABSTRACT

Single nucleotide polymorphisms (SNP) store the genetic variability among populations. In this report, genomic information from humans was retrieved from "*The 1000 genomes project*" with the motivation of predicting the continental origin of an individual by only using his sequenced DNA. Three different Machine Learning algorithms (Random Forest, Support Vector Machines and Bayes classifier) have demonstrated to be able to perform these predictions by only using certain information contained in the chromosome 1 with good performances (accuracies of 0.87 $\pm$0.04, 0.93 $\pm$ 0.04 and 0.83 $\pm$ 0.04 respectively) and statistical significance. Moreover, the ensemble method has shown high predictive accuracy of 0.93 $\pm$0.04.

Introduction

Genomic variation in human

The deoxyribonucleic acid (DNA) sequence information is valuable for research, health and commercial uses[1,2]. Sequencing is the technique used to determine the DNA code. [a] Over the past decades, this technology has improved, allowing biologists to join the big-data club[3]. One single sequenced human genome occupies around 140 gigabytes. Thus, computer scientists are essential to retrieve the most of the information that the DNA sequence has to offer. In short, sequencing serves to differentiate individuals between each other based on genetic information, for example for disease prediction, therapy or genetic ethnicity tracking[3,4].

In this research, we focused on genetic ethnicity based on single nucleotide polymorphisms (SNPs). A SNP is a variation at a single position in the DNA sequence among individuals. The DNA sequence consists of a chain of four nucleotide bases: Adenine, Cytisine, Guanine and Thymine, represented by letters A, C, G and T, respectively. If more than 1% of a population carries a different nucleotide base at a specific position than the rest (reference allele), this variation will be classified as a SNP (alternative allele).

So far, SNPs have been widely studied, especially for human health. For instance, SNPs are helpful for prediction of an individual's response to certain drugs, susceptibility to environmental factors, or risk of particular diseases[5,6].

The 1000 genomes project

In the last project of "The 1000 genomes project", whole genome sequencing was performed on 2504 individuals from 26 populations located in East Asia, South Asia, Europe, America and Africa.[7]. Within this freely accessible database, we focused on the dataset of SNP variation.

The first chromosome alone has already 7 million reported SNPs, which we thought as sufficient to predict populations. Chromosome 1 is the largest human chromosome, containing approximately 8% of all human genetic information[8]. Furthermore, previous studies have shown differences between the populations of Africa, East-Asia and Europe based on the information of chromosome 1. Figure 1 shows the result of this study, where the differentiation measure (bottom) is the log-likelihood ratio test statistic of the the populations[8]. In this project, our main research question is to predict the continent of origin of a person based on the SNPs located on chromosome 1.

Machine Learning

Machine learning is a Compute Science branch devoted to automate analytical model building. Using algorithms that iteratively learn from data, it allows computers to find hidden insights without being explicitly programmed where to look.

[a] "A knowledge of sequences could contribute much to our understanding of living matter."[Frederick Sanger]

It has become a very important analysis tool for the ever-increasing biological data[9]. The whole genome sequence contains around 84.7 million SNPs; these variances are called features in Machine learning language (from now on we will call them features instead of variances). In our project, these features were analysed for 2504 individuals (”subjects” from now on). With a simple look one already realizes that machine learning is essential for the analysis. The machine learning tasks included data preprocessing, test and training set selection, algorithm election, training with tuned parameters and evaluation (summarized as a flowchart in figure 2).
We first selected a dataset containing SNPs of chromosome 1, reducing the initial sequence information down to 7 million features. The data contained the features 0|0 0|1 1|0 1|1, which needed to be transformed to numerical values. Subsequently we selected the first 400.000 features, assuming that this would be enough, as the first part of chromosome 1 already contains important information ([8]). Out of the cropped dataset, the 400 features with highest Information Gain (IG) were selected. Next we split the dataset into a test set, consisting 500 randomly selected subjects, and a training set, with the remaining 2004 subjects. Later, we created prediction models with Random forest, Support Vector Machine (SVM) and Naïve Bayes classifier. For these models we trained the data using parameter tuning and 10-fold cross-validation (CV). After the training, we evaluated the models and calculated the accuracy of all models individually. Finally these models were merged using majority voting to achieve the final model.
Let us see if machine learning is racist.

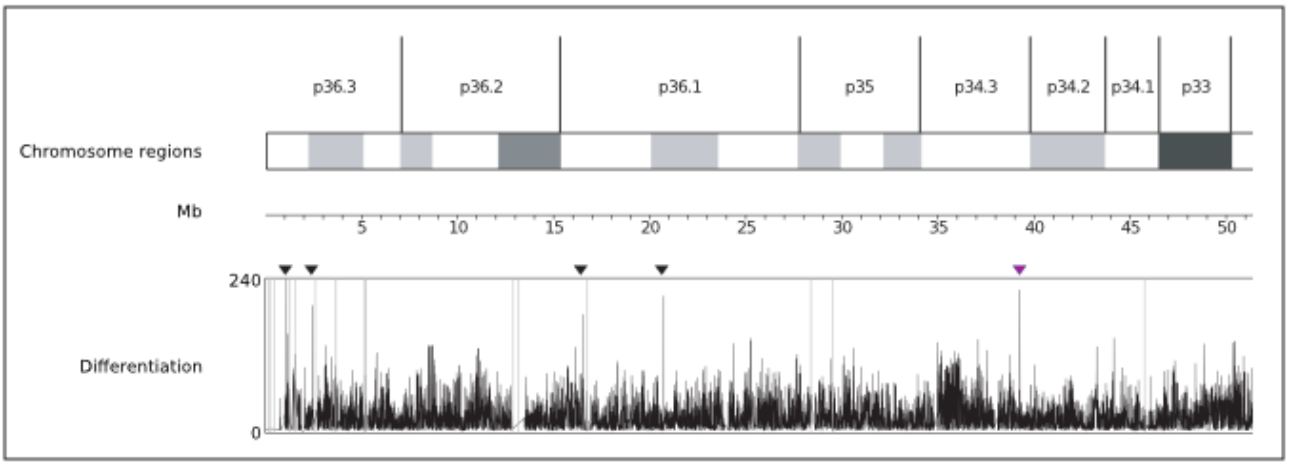

Figure 1. Differentiation peaks of SNP variances of only the first part of Chromosome 1, showing the log-likelihood ratio test statistic between the populations: Africa, East-Asia and Europe. Figure from another paper[8].

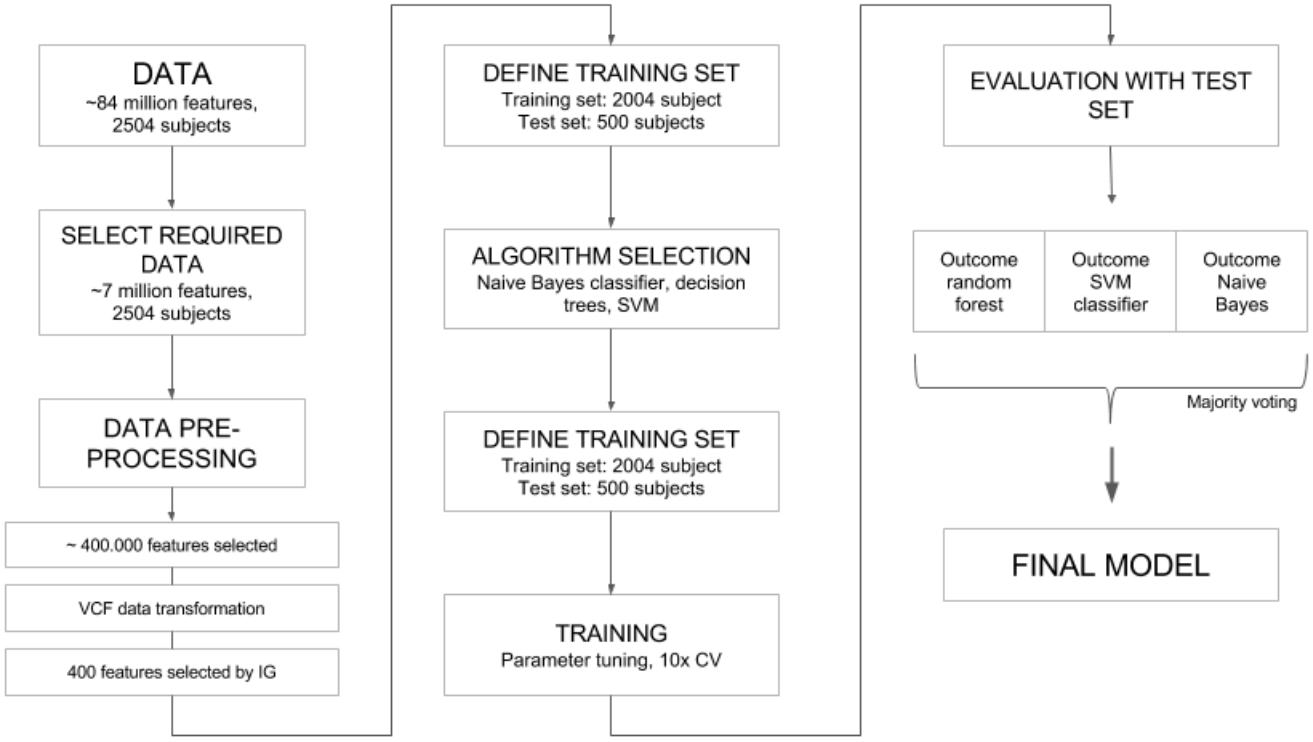

Figure 2. Flowchart Machine Learning (ML) tasks: We started with a dataset with 84 million features and 2504 subjects. We selected the required data containing 7million features and started the pre-processing: first selecting 400.000 features and changing the data into numeric values, then selecting 400 features with the highest information gain. Subsequently, we split the data into a test set (500 subjects) and training set (2004 subjects). With this data we build 3 different prediction models: Random Forest, Support Vector Machine (SVM) and Naïve Bayes classifier. These models were trained using parameter tuning and 10-fold cross-validation (CV) and evaluated with the test set. The outcome of the three models was finally combined using majority voting to create the final model

Data inspection and preparation

Original dataset and data filtering

The dataset used in this project was retrieved from International Genome Sample Resource, the largest public database of human variation and genotype data. More specifically, the dataset was retrieved from the *Phase 3* resource, which contains the genomes of 2504 subjects from 26 different countries and 5 different continents. A raw inspection of the continental variability in the whole database can be seen in figure 3, the data is split in the continents East Asia, South Asia, Europe, America and Africa. The class Africa has the most subjects, 26.4%, and America the least, 13.9 %.

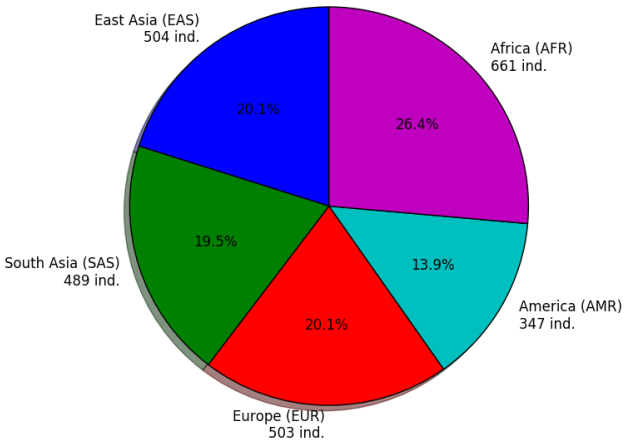

Figure 3. International Genome Sample Resource (phase 3) population distribution across continents. subjects were split in to the following continents: Africa (AFR), America(AMR), Europe (EUR),South Asia (SAS) and East Asia (EAS)

In the database, the genome is divided in chromosome files, which weigh up to ∼1 GB when compressed, and ∼ 80 GB when uncompressed. For the 23 pairs of chromosomes, the whole dataset size weighs around 2 TB; the DNA stores an extremely massive amount of information. Due to the huge amount of features present in the original dataset, the data preprocessing and model building would probably be hindered. Based on the biological background only the information contained on chromosome 1, was retrieved ([8]). The dataset was still immense (7 million features per subject). Therefore, we filtered this data by retrieving only the information of the first 400.000 SNPs.

Data preprocessing

The dataset was represented in Variant Call Format (VCF) format[10], which is a standard for storing gene sequence variations in Bioinformatics. Herein, for every SNP the genotype [b] is encoded as allele values separated by a pipe. Each position will either receive the value 0, if the allele matches with the reference allele, or 1, 2 or 3 corresponding to the alternative alleles listed in *ALT* (alternative base-pair list).

Humans have two copies of every chromosome, each genotype has two values for every SNP (Table 1). These values were transformed to discrete numerical values in which only three possible values (0, 0.5 and 1) were assigned. They could be interpreted as the variation strength observed at a certain SNP for an individual, where a higher number means stronger variation.

Table 1. Data transformation. In this data transformation, "1" value can also represent the numbers 2 and 3 (the alternative alleles listed in the *ALT*)

Possible Genotype	0|0	0|1	1|0	2|0	0|2	1|0	1|1	2|2
Assigned value	0	0.5	0.5	0.5	0.5	0.5	1	1

Feature selection

Initially the dataset suffered from curse of dimensionality: it had 200 times more SNP attributes (400.000) than subjects (2004). This could lead to problems in machine learning implementations, as the models should be trained with enough

[b] The genotype of a person is her complete heritable genetic identity

combinations of features to reach a high coverage[11]. Moreover, it has been reported that not all SNP are significant for making discriminating analysis among populations[12]. To reduce the dimensionality of our dataset, the IG of every SNP attribute was calculated to select only those with the highest IG (Supplementary data *IG_entropy.py*). Principal Component Analysis could have been used, but its implementation for our discrete values would have been more complex.

For every feature (SNP_i), IG was computed as follows:

$$IG(SNP_i|Continent) = H(SNP_i) - H(SNP_i|Continent) \tag{1}$$

where $H(SNP_i)$ represents the information entropy of the feature:

$$H(SNP_i) = -\sum_{J=\{0,0.5,1\}} p_j log_2 p_j \tag{2}$$

In the previous equation, $H(SNP_i|Continent)$ represents the conditional entropy:

$$H(SNP_i|Continent) = \sum_{y_i=\{EAS,SAS,EUR,AMR,AFR\}} p_{y_i} H(SNP_i|Continent = y_i) \tag{3}$$

And $H(SNP_i|Continent = y_i)$ is the specific conditional entropy:

$$H(SNP_i|Continent = y_i) = -p_{y_i} log_2(p_{y_i}) \tag{4}$$

In the previous equations, p_{y_i} denotes the probability of observing the y_i label (continent, in our case) for the specific subset being studied. Once computed the IG for every SNP, we observed how 4600 SNP ranked an IG higher than 1.5. In order to ensure stability of our built models, this high number of features was reduced, and only the first 400 SNP attributes with the highest IG were retrieved.

Test set and training set

In order to avoid an inducted bias in the ML models, 20% of the original dataset was randomly selected and separated for building an independent test set (500 subjects). The remaining dataset was used as a training set (2004 subjects), which was used for model building and CV. The performance of the different models was quantified by their predictions with the test set. As depicted in figures 4 and 5, both sets are heterogeneous and contain subjects from every different continent.

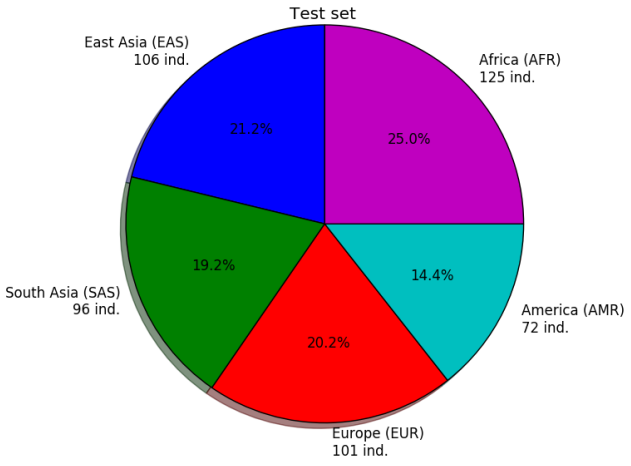

Figure 4. Test set population

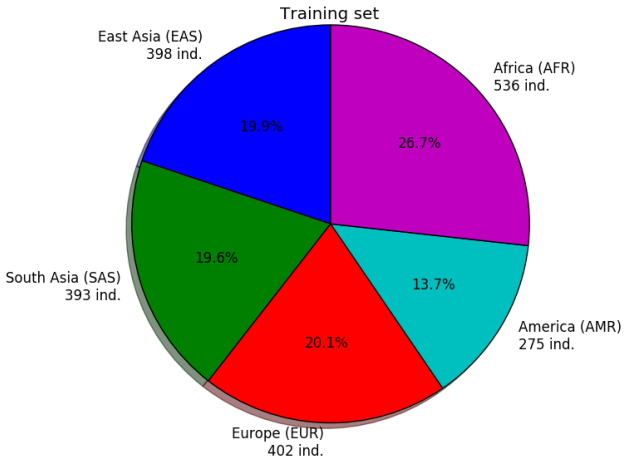

Figure 5. Training set population

Experimental set-up

Random Forest

Random forests are ensemble models created from several decision trees. Decision trees are a classification and regression method, based on non-parametric supervised learning[13]. The flowchart-like trees are built by sequential splitting of the data, using the feature with highest impurity. Two common impurity measures are Gini index: $H(X_m) = \sum_k p_{mk}(1 - p_{mk})$, and Entropy: $H(X_m) = -\sum_k p_{mk} log_2(p_{mk})$, where p is the probability of a class. Every time a node is generated, the impurity is recalculated for the remaining features. New splits occur until either all the final leaves have only one class or there are no more features left[14]. A model with only one decision tree is unstable and prone to overfitting. Random forest offers a good solution by boostrap aggregation or bagging[15]. In this case many trees are generated out of various sub-datasets, selected from the original with replacement, containing only certain features. The final random forest is the combination of the various trees, where the output is chosen by majority voting.

An original Python script was developed (Supplementary data, *random_forest.py*), using the functions *RandomForestClassifier*, *predict* and *predict_proba* from the free library "Scikit-learn"[16]. We performed CV for optimum parameter selection. The studied variables were: number of trees (from 1 to 150), criterion for quality of the split (gini impurity or entropy) and pruning (from 0 to maximum tree depth). They were changed using the parameters $n\_estimators = 100$, $criterion$ and $max\_depth = 10$ respectively, available in the $RandomForestClassifier$ function. Other parameters that could be tuned in the $RandomForestClassifier$ function did not report any significant difference in the performance and therefore are no longer discussed in this paper.

Support Vector Machine (SVM)

SVMs are algorithms based on statistical learning theory[17] and used in the field of machine learning. They are (usually) supervised learning models widely used in solving problems related to classification or regression analysis, learning and prediction models[18] .[18] . In more detail, a SVM model represents the given data as points in space and the algorithm classifies these points into groups (classes) according to principle of margin maximisation; maximising the minimum distance from the separating decision boundary to the nearest point. During SVM training, the algorithm is trying to find this optimal decision boundary, called hyperplane, that best divides the dataset into separate classes. This can be achieved by the help of the support vectors; data points closest to the hyperplane that are being used as critical beacons for the definition and standardisation of the margin(s)[19] (figure 6). Two (or more) classes will be separated by the calculated hyperplane with the best possible margin, giving the finest possible chance of classifying any new data correctly. SVMs can handle linear separable data (figure 6), but also non-linear. In case we are dealing with more complicated data, then a linear separation can be achieved (most likely) in a high dimensional space (Cover's theorem). The non-linearity can be accomplished by applying the kernel trick to handle the higher dimensional space separation[20] (figure 7).

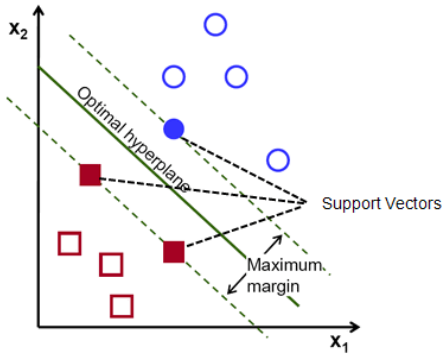

Figure 6. Illustration of linearly separable SVM

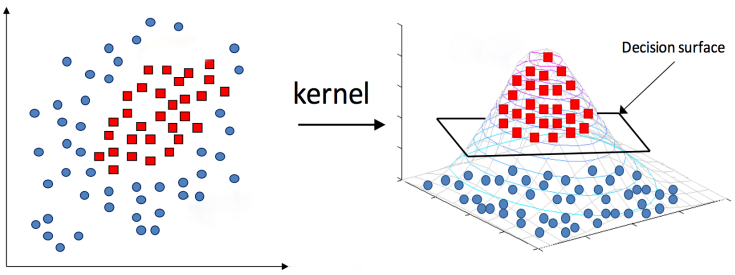

Figure 7. Visual representation of the Kernel Trick

The SVM algorithm recently was extended to handle multi-class classification problems by combining several binary classifiers. There are two basic methods that can perform the task, "one-against-all" and "one-against-one". In the first case,

n-SVM models will be constructed, where n is the number of the classes found in the data. Each model is then trained with all points of its own (only) class. In the latter case, $n(n-1)/2$ classifiers will be constructed, where each one is trained on data from two classes[21] For this project we will work on Python in combination with the SVM package from "Scikit-learn" (Supplementary data, *SVM.py*).

One vital step when SVM learning is the tuning of parameters. There are three major parameters to tune. The first one concerns the kernel function;there are 4 available functions[16]:

- Linear - $K(X,Y) = X^T Y$
- Polynomial - $K(X,Y) = (\gamma \cdot X^T Y + r)^d, \gamma > 0$
- RBF - $K(X,Y) = e^{-\gamma \cdot ||X-Y||^2}, \gamma > 0$
- Sigmoid - $K(X,Y) = \tanh(\gamma \cdot X^T Y + r)$

The other parameters influence the kernel function: C parameter, and γ parameter The hyperparameter optimisation was performed using the "Scikit-learn" standard function *GridSearchCV*[16] that compares every possible combination of our chosen input values. We tried all possible kernel functions, C: $[0.1, 1, 10, 50, 100, 1000]$ and gamma: $[1e^{-2}, 1e^{-3}, 1e^{-4}, 1e^{-5}]$.

Naïve Bayes classifier

A Bayes classifier is a probability classifier that can estimate the class of an input with Maximum a posteriori estimation (MAP). More specifically, a Naïve Bayes classifier makes these estimations by assuming that all the attributes are independent between them[22]. The classification rule for a i class (Y) that it follows, given that we have k attributes (X), is:

$$Pr(Y = y_i | X_1 = u_1, X_2, ..., X_k = u_k) = \frac{Pr(Y = y_i) \prod_{i=1}^{k} Pr(X_i | Y = y_k)}{Pr(X_1 = u_1, X_2 = u_2, ..., X_k = u_k)} \quad (5)$$

Due to the denominator is a constant, we can save computational time by ignoring it:

$$Pr(Y = y_i | X_1, X_2, ..., X_k) = Pr(Y = y_i) \prod_{i=1}^{k} Pr(X_i | Y = y_k) \quad (6)$$

The algorithm classifies each input given the following rule:

$$Y^{predicted} = argmax_v (Pr(Y = y_v | X_1, X_2, ..., X_k)) \quad (7)$$

The density estimator (probabilities of a certain SNP having a certain value for a certain population) were completely learnt from the training set as follows:

$$Pr(X_i = x_{ij} | Y = y_k) = \frac{D(X_i = x_{ij} and Y = y_k) + l}{D(Y = y_k) + lJ}, for j = 1, 2, ..., J \quad (8)$$

In which X_i, x_{ij}, Y and y_k represent the input SNP, its possible values (J possible values, J=3 in this case), the continent, and the specific origin continent (output/label), respectively. $D\{m\}$ represents the number of elements of "m". In the previous equation, the only adjustable parameter is known as the Dirichlet Weight (represented as l), and it will be optimized with CV. The prior probability for each label y_k was computed as follows:

$$Pr(Y = y_k) = \frac{D(Y = y_k) + l}{|D| + 2l} \quad (9)$$

In which $|D|$ is the number of training set subjects.

We implemented the Naïve Bayes classifier from scratch in *naive_bayes.py* (*Scripts* directory) already described by *Sambo et.al*[23] and *Hu et.al* [24], in which the algorithm was specifically applied to a SNP dataset with discrete-valued inputs (0, 0.5 and 1).

Cross-validation (CV)

A 10-fold CV was performed in our models to choose the best parameters. For this task, the training set was randomly split into 10 subsets (due to the almost equal proportion of continents in the training set, building them randomly was feasible). Then, during 10 iterations, a subset was left out (CV set) and a model was build on the remaining 9 sets. The misclassifications on the CV set were counted and summed in every iteration.

Stacking of methods

An ensemble method combines different classifiers, where every model contributes in the predictions. Firstly all classifiers will be trained with the same dataset and the ensemble classifier will be trained in order to make the final prediction by using the base-classifiers as additional inputs. Stacking has two main ways of combining results: weighted and unweighted majority voting. One of the advantages of ensemble methods is the unlikeness of all classifiers predicting the same wrong class. The optimal classification is achieved when, for each sample, a false prediction is made by the minority of the classifiers[25]. In our project we used three base classifiers: random forest (Cl1), SVM (Cl2) and Naïve Bayes classifier (Cl3). For every new $\bar{x}$-sample, $y = Cl1(\bar{x}), Cl2(\bar{x}), Cl2(\bar{x}) = ClM(\bar{x})$ will be calculated. The majority will decide the class of the new sample. For the ensemble of the methods we will use the "VotingClassifier" provided by "Scikit-learn" . The only tuning in this method is the way that the voting should be done. The accuracy and misclassifications of each model pointed us to higher weight SVM and then equally RF and NB. The tuning of the weight was performed with the trial and error method. The final weight that gave the higher accuracy were: $[RF : 0.001, SVM : 0.95, NB : 0.001]$.

Model validation and performance

True error estimation

Knowing the sampling error of our test set ($error_S(h)$) with size n, if it is assumed that it follows a Normal distribution (it is a fair assumption, given that $n > 30$), with a 99% confidence interval (in which z_n=2,58) it can be assured that the true error ($error_D(h)$) will fall in the following confidence interval[14]:

$$error_S(h) \pm 2,58\sqrt{\frac{error_S(h)(1-error_S(h))}{n}} \tag{10}$$

Receiver Operating Characteristic (ROC) plot

Five ROC plots were created for each model, one per continent class, because of the multi-classification approach. In order to do so, it was iteratively assessed positive a certain continent and the other continents were considered as negatives. The predicted probabilities of belonging to the "positive continent" were sorted by increased value, and the decision boundary was iteratively updated to measure the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The true positive rate, $TPR = \frac{TP}{TP+FN}$, was represented against the false positive rate, $FPR = \frac{FP}{FP+TN}$. The performance was then measured by numerically integrating the area under the curve (AUC) of every ROC plot and averaging the 5 AUC.

Confusion Matrix

For each method we also made confusion matrices (2D) where the rows show the instances in a predicted class and the columns show the instances in an actual class (or vice versa).

Permutation test

A permutation approach was used to asses the statistical significance of our models. The labels of the training set were randomly permuted 1000 times, generating 1000 scrambled-label files. With every file, a model was trained and the misclassifications in the test set estimations were counted, thus computing the chance distribution. Due to time limitation we were not able to perform 1000 times permutation for the SVM method, this was done only 50 times These new models represent situations where no link between SNPs and continent origin exists.

Results

Initially we built three different prediction models and evaluated them using the test set. The results are in the form of ROC-plot, confusion matrix and permutation histogram. First we discuss briefly the different models and then compare the results. The results of the ensemble model are discussed at the end.

For the Random Forest CV was performed on different parameters to assure the best model as explained in the methods section, the results of the CV are shown in the Figures refFig:Data1 and S2 Firstly, the number of trees that participate in the forest was set to 100 (S1). The node split criterion in the final model was Gini impurity, as it yielded a slightly better performance. Finally, the tree was pruned based on a maximum depth of 10 (S3, S4). Optimizing the hyperparameter for SVM like described in the methods we found the optimal parameters to be kernel: RBF, C:50, γ: 0.001. This is subsequently used to create the final model. SVM was optimally trained performing 10-fold Cross Validation, as reported in the learning curve figure S7. For the Naive Bayes classifier we performed CV to estimate the optimal Dirichlet weight parameter by evaluating the CV error for models ranging an l value from -100 to +100, as reported in figure S5. Once identified the range in which the parameter scored the smaller CV error, an additional scanning was performed with a smaller step, as shown in figure S6. The optimal Dirichlet weight parameter found was $l = -0.8$, reporting 299 misclassifications (out of 2004 estimations) in the CV.

Evaluating models

The performance of the Random Forest was assessed with a ROC plot (figure 8a). Most of the continents (East Asia, South Asia, Europe and Africa) show an AUC value higher than 0.99, whereas the AUC for America is slightly lower (0.939). The mean AUC is 0.984, much higher than 0.5. Accordingly, we can conclude that our model is much better than a random method. A similar conclusion is drawn from the confusion matrix (Figure 9a). Most of the continents are well predicted (0, 10, 10 or 4 false negatives out of 106, 96, 101 and 125 subjects respectively). However, in the case of America, more individuals were classified in the wrong continent (41) than in the real one (31). These suggest that these subjects were the most challenging ones to predict their continental origin. In overall, 435 out of 500 individuals were correctly classified, giving a test set error of 0.13 ± 0.04. A permutation approach was done to study the statistical significance of our model. In the figure 10c it is seen how the number of misclassifications of random models is much higher ($>$300) than the number of the proposed model (67): our model works better than a random classification.

The SVM method shows the lowest error of the three base models (0.07 ± 0.03). The figure 8b shows in East Asia an AUC value of 1.000, which suggests that there are no false positives. South Asia and Africa have a value higher than 0.99, while America has the lowest value of 0.946. The average of the AUC is 0,98, indicating that the model is much better than random. We could draw the same conclusion from the confusion matrix (figure 9b. Here we see that the SVM method classified everything correct for East Asia. South Asia and Africa have both 2 false negatives and Europe has 9. The worst predicted continent is America, in total 23 out of 72 subjects has been classified wrongly. Interestingly the model could not distinguish well between America and EUR as 18 of the false negatives were predicted as Europe which is comparable with the Random Forest method. To study the statistical significance of our model a permutation approach was done. We show in figure 10b the number of misclassifications of random models (green bars) and our model (blue), please note that for SVM we used 50 permutations in stead of 1000. But as we can see in the histogram 50 permutation show a clear difference between random models ($>$350) and our proposed model (75).

In the ROC plot for the Naïve Bayes (figure 8c), we can observe how the performance of the method varies depending of the continental origin of the test set subjects. Whereas for East Asia and African subjects the AUC gets closer to the one of a perfect method (0.988 and 0.981, respectively) for the case of South Asia and America subjects the performance is much lower, having an AUC of 0.815 and 0.786. The mean AUC for the 5 ROC plots is 0.90. Furthermore, in the confusion matrix attached in figure 9c these observations can be reinforced: the test set sample error is much lower for the case of East Asia (7 false negatives out of 124 subjects) and Africa (13 out of 96), whereas for the case of America subjects the number of false negatives (44) is higher than the number of the correctly predicted subjects (28). This coincides with the difficulties to predict the continental origin for individuals from America. Out of the 500 subjects in the test set 80 were misclassified, reporting a relatively accurate overall performance (test set error = 0.16 ± 0.04). In terms of the statistical significance of the built model, we can observe in figure 10c how the number of misclassifications (84) is much lower than the ones obtained by the permuted models, suggesting that our model works better than random models and ensuring its statistical significance.

From these results we can conclude that there is not much difference in AUC values between Random Forest and SVM (0.984 and 0.983, respectively). However the confusion matrix shows a better prediction of the continents for SVM than for Random Forest, SVM has in total 36 false negatives out of the 500 subjects, whereas Random Forest has 65 false negatives. Also SVM classifies 49 out 72 subjects correctly for AMR whereas Random forest only 27. The Naïve Bayes classifier has on average much a lower value for AUC (0.904) compared to Random forest and SVM.

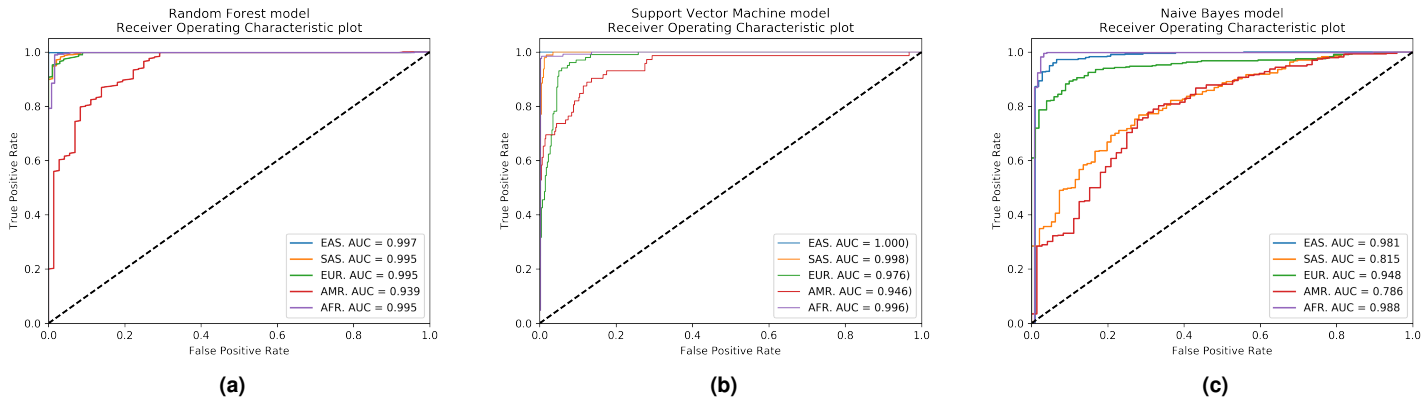

Figure 8. ROC plots with the Area under the curve (AUC) value of the three different prediction models for the different continents: East Asia (EAS), South Asia (SAS), Europe (EUR), America (AMR) and Africa (AFR) a. Random forest predictions have an AUC value of 0.99 for all continents except for AMR, which scores slightly lower (0.93). b. Support Vector Machine has an AUC value of 1 for EAS; SAS and AFR have a value of 0.99; EUR 0.98, and AMR has the lowest value with 0.94. c. Naïve Bayes shows the highest AUC value for AFR (0.99), followed by EAS (0.98), EUR (0.95), SAS (0.82) and EUR (0.79)

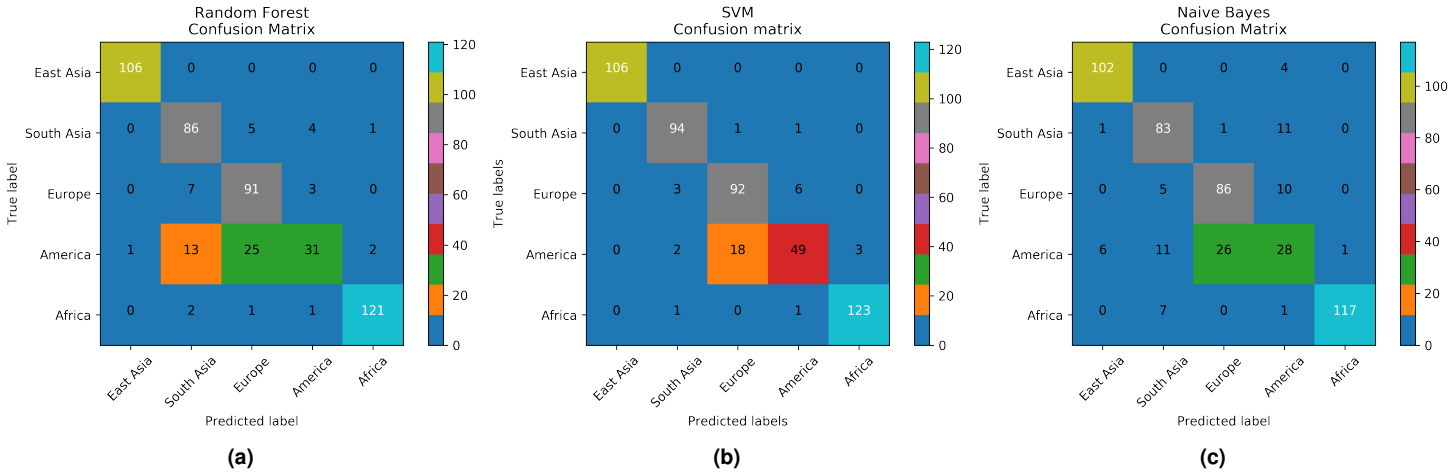

Figure 9. Confusion matrices with the values of the misclassifications for the different continents. The x-axis represents the predicted label and the y-axis the true labels a.Random forest: Except for America all the other continents have few false negatives. Most American false negatives fall in Europa and South Asia b. Support Vector Machine: Also in SVM there are almost none false negatives except for America. America cannot be separated from Europa. c. Naïve Bayes: Overall there are more false negatives in this method and, again, America was the worst at its prediction

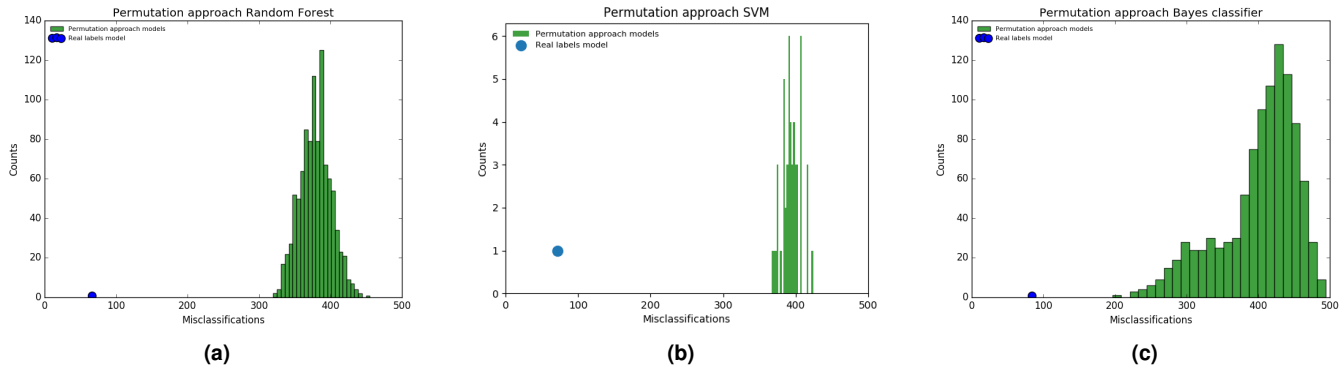

Figure 10. Permutation plots for the different methods. On the x-axis the number of misclassifications are represented and the y-axis the counts. The green bars show the results of the permuted data and the blue dot represents the result of our models. a.Random forest: There is a nice normally distributed plot for the randomised data and the green bars show much higher error than our model. b. Support Vector Machine: Similar results that of Random forest. The random data shows a normally distributed number of misclassifications, where as our model performs much better. c. Naïve Bayes: The randomised data is close to a normal distribution and the error is much higher than the model. For all the cases we can conclude that our model is not derived from randomised data

Ensemble model

An ensemble method was built based on the results of the three base models. The evaluation of this model was done in the same way. The ROC plot (figure 11a) shows a mean AUC of 0.98, which is similar to the ones of random forest and SVM models. In the prediction of European origin, the ensemble models fails to outperform the random forest. The confusion matrix (figure 11b) confirms the overall good performance. In short, the ensemble model is slightly better than any of the base models alone, especially for America, the continent that was most difficult to predict. However, the difference is very small when compared to the SVM or random forest models. A summary of all results can be looked up in the table 2.

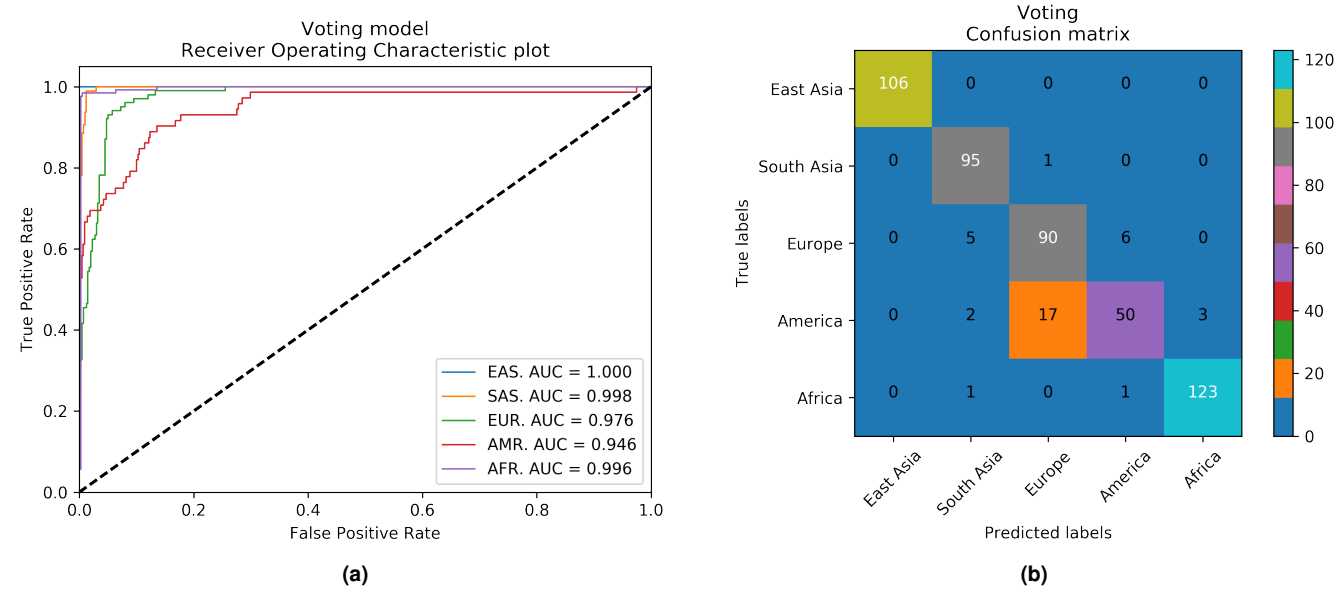

Figure 11. Results of the Ensemble method a. ROC Plot of the Ensemble Method. The calculated AUC are 0.946 for America, 0.076 for Europe and above 0.99 for the rest, reaching 1.00 in South Asia prediction. b. Confusion matrix of the Ensemble Method. In accordance with the ROC plot, the misclassifications are barely present. America again stands by the difficulty of its prediction (22 out of 72 subjects were wrongly classified).

Table 2. Results obtained by the different models

	Test set Misclassifications	Test set accuracy	Test set error	Average AUC
Random forest	65	0.87±0.04	0.13±0.04	0.98
SVM	72	0.93±0.04	0.07 ±0.03	0.98
Naïve Bayes classifier	80	0.83± 0.04	0.17±0.04	0.90
Ensemble of methods	74	0.93±0.04	0.07 ±0.03	0.98

Discussion

In this research we wanted to see if we could predict the continent of origin of an individual based on the SNPs located on chromosome 1. We made three different prediction models: random forest, SVM and Naïve Bayes, and in the end merged them into a final prediction model. Overall the base models performed well, with accuracies of 0.87, 0.93, 0.83 respectively). Furthermore, from the results of the ROC-plots (Figure 8 we concluded that all of our models perform better than a random model as their average AUC value is higher than 0.5 (0.98 for Random forest and SVM and 0.90 for Naïve Bayes). We achieved the same result from the permutation approach, where the random data has $>$200 misclassifications and our models have $<$100. The confusion tables (Figure 9 show that for most continents (East Asia, South Asia, Europe and Africa) the predictions are accurate, as the number misclassifications is low. However, in all the models America is not well classified, they cannot distinguish well between America and Europe. By merging the models with Ensemble voting we achieved an accuracy of 0.93, similar to that of the SVM model. From this we can conclude that a combination of the models is slightly improved for the accuracy.

The Naïve Bayes classifier worked well for our problem, but was the worst of all predictors. This can be mainly attributed to the biological characteristics of SNPs. They are known to be independent from each other, fulfilling the assumptions of a Naïve Bayes classifier. However, certain SNPs are dependent from other ones (linkage disequilibrium); this fact was neglected in our model[26]. The joint distribution was discarded, as it is more prone for overfitting and probably yielding worse results. Besides, our results clearly showed that America cannot be classified well, and were often classified as Europe. This makes sense as the data may contain individuals from North America (NA) and South America (SA), and NA was partly populated by European settlers. In the future the labels could be split into more labels, e.g. NA and SA. It would be interesting to see if this would actually improve the model or if America is such a mixed population that the predictions will not improve at all. The improvement in performance of the ensemble model is small. In the future an ensemble model based on weighted voting, could be made to improve even further the performance of our model. It was not done in this project due to lack of time. Moreover, the stacking ensemble should be trained on data unused on training the base learners, instead of the same training data as for the learners, as we did. This model could have been used with more SNPs as these methods can handle up to 10.000 attributes, but it is not worth the cost for the little putative improvement.

It is surprising that by only retrieving some features from a chromosome that only contains 8% of the total genetic information included in the human genome we were able to predict the continental origin of some subjects with a satisfactory accuracy.

References

1. Lehrach, H. DNA sequencing methods in human genetics and disease research. *F1000prime reports* **5**, 34 (2013).
2. Shendure, J. & Lieberman Aiden, E. The expanding scope of DNA sequencing. *Nat. biotechnology* **30**, 1084–94 (2012).
3. Vivien Marx. The big challenges of big data. *Nat.* **498** (2013).
4. Pettersson, E., Lundeberg, J. & Ahmadian, A. Generations of sequencing technologies. *Elsevier* (2009).
5. Dickson, S. P., Wang, K., Krantz, I., Hakonarson, H. & Goldstein, D. B. Rare Variants Create Synthetic Genome-Wide Associations. *PLoS Biol.* (2010).
6. Gabriel, S. B. *et al.* The Structure of Haplotype Blocks in the Human Genome. *Am. J. Hum. Genet. Methods Biol. Nat. Nat. N. E. Morton, Proc. Natl. Acad. Sci. U.S.A. J. Cell Sci. Cell Genes Dev. Cell Nat. Genet. C. Tease et al. Am. J. Hum. Genet. J. Cell Biol. Trends Genet.* **49**, 1186–529 (1985).
7. The 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nat.* **526** (2015).
8. Gregory, S. G. *et al.* The DNA sequence and biological annotation of human chromosome 1. *Nat.* (2006).
9. Tarca, A. L., Carey, V. J., Chen, X.-w., Romero, R. & Drăghici, S. Machine learning and its applications to biology. *PLoS computational biology* (2007).
10. Danecek, P. *et al.* The variant call format and VCFtools. *Bioinforma.* (2011).
11. Hughes, G. F. On the Mean Accuracy of Statistical Pattern Recognizers. *IEEE Transactions on Inf. Theory* (1968).
12. Baye, T. *et al.* Population structure analysis using rare and common functional variants. *BMC Proc.* (2011).
13. Breiman, L. *Classification and regression trees* (Chapman & Hall/CRC, 1998).
14. Flach, P. A. *Machine learning : the art and science of algorithms that make sense of data* (Cambridge University Press, 2012).
15. Guy, R. T., Santago, P. & Langefeld, C. D. Bootstrap aggregating of alternating decision trees to detect sets of SNPs that associate with disease. *Genet. epidemiology* **36**, 99–106 (2012).
16. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
17. Yiling Chen and Isaac G. Councill. An Introduction to Support Vector Machines A Review. *AI Mag.* **24** (2003).
18. Kausar, N., Samir, B. B., Abdullah, A., Ahmad, I. & Hussain, M. *A Review of Classification Approaches Using Support Vector Machine in Intrusion Detection* (ommunications in Computer and Information Science).
19. Burbidge, R. & Buxton, B. *An Introduction to Support Vector Machines for Data Mining* (Cambridge University Press New York).
20. Lee, Y.-J., Yeh, Y.-R. & Pao, H.-K. *An Introduction to Support Vector Machines* (Cambridge University Press New York).
21. Hsu, C. W. & Lin, C. J. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks* (2002).
22. Mitchell, T. M. T. M. *Machine Learning* (McGraw-Hill, 1997).
23. Sambo, F., Trifoglio, E., Di Camillo, B., Toffolo, G. M. & Cobelli, C. Bag of Naive Bayes: biomarker selection and classification from genome-wide SNP data. *BMC Bioinforma.* (2012).
24. Hu, P. *et al.* A Simple Algorithm for Population Classification. *Sci. Reports* (2016).
25. Thomas G. Dietterich. Ensemble Methods in Machine Learning. *Lect. Notes Comput. Sci.* **1857** (2000).
26. Ewens, W. J. Genetics of populations. *Am. journal human genetics* (1987).

Group members contribution

All authors participated in the design of the experiments, analysis of the results and revision of the manuscript.

Supplementary data

Scripts

The scripts created for this project can be accessed through the following link:
https://drive.google.com/open?id=0B0ukoXwKukmbLVpaV1BnWk1FS0U)
In brief, the scripts were used for the following purposes:

- **naive_bayes.py**: Naïve Bayes classifier
- **preprocess_data.py**: Data preprocessing
- **IG_entropy.py**: calculate the IG of every attribute
- **featureselection.py**: Select the features with the highest IG
- **random_forest.py**: Random Forest classifier
- **SVM.py**: Support Vector Machine classifier
- **voting.py**: Ensemble classifier

Parameter tuning of base models

As described in methods, cross-validation (CV) was performed in order to find the values for the best performance. Here is the CV results of the parameters that showed influence in the models' performance.

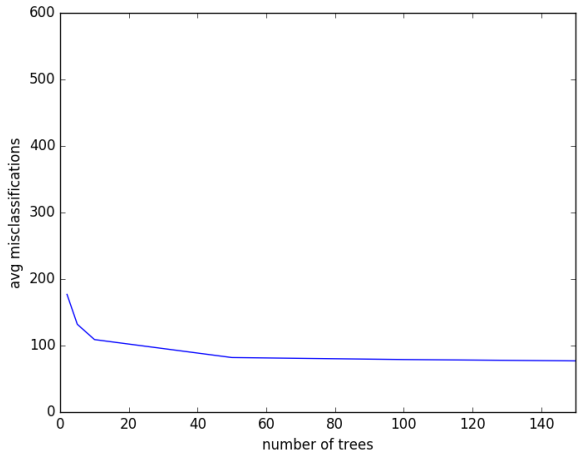

Figure S1. Cross-validation result for Random Forest classifier with number of trees ranging from 1 to 150. The x-axis shows the number of trees and the y-axis the average misclassifications. We can see from this figure that the misclassification is not changing much after 10 trees.

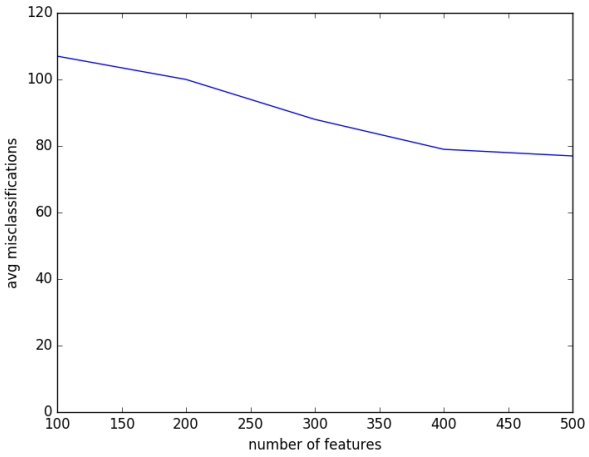

Figure S2. Cross-validation results for Random Forest classifier with training set features ranging from 100 to 500. The x-axis shows the number of features and the y-axis the average misclassifications. From this figure we see that the average misclassifications is stable after 400 features.

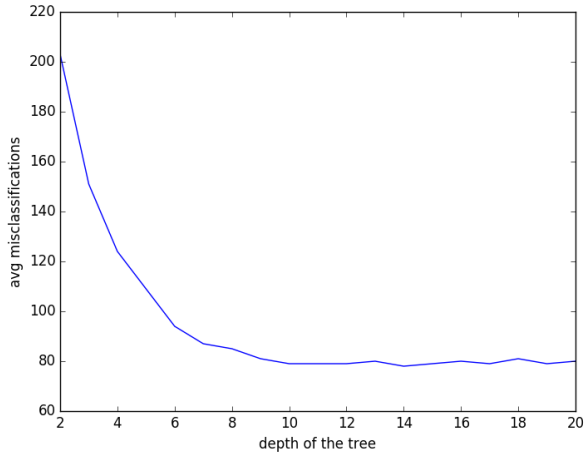

Figure S3. Cross-validation results for Random Forest classifier with maximum depth of tree ranging from 1 to 20, using Gini impurity criterion. The x-axis shows the depth of the tree and the y-axis the average misclassification. We can see in this figure that after the value 10 for depth of the tree the number of misclassification does not change.

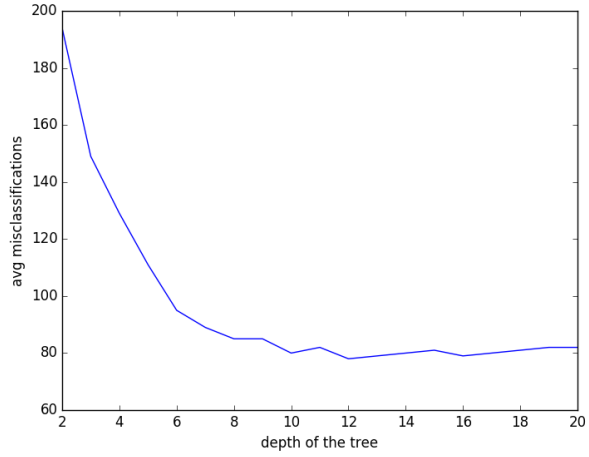

Figure S4. Cross-validation results for Random Forest classifier with maximum depth of tree ranging from 1 to 20, using information gain criterion. The x-axis shows the depth of the tree and the y-axis the average misclassification. Here we show that also with the information gain criterion we do not see a change in the average of misclassifications after the value 10 for depth of the tree.

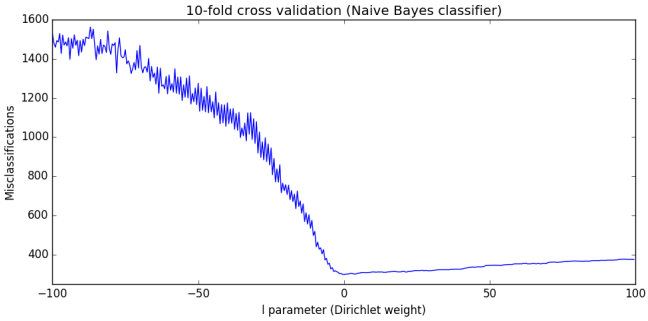

Figure S5. Cross-validation results for Naïve Bayes classifier with l ranging from -100 to +100. The x-axis shows the Dirichlet weight and the y-axis the number of misclassifications.

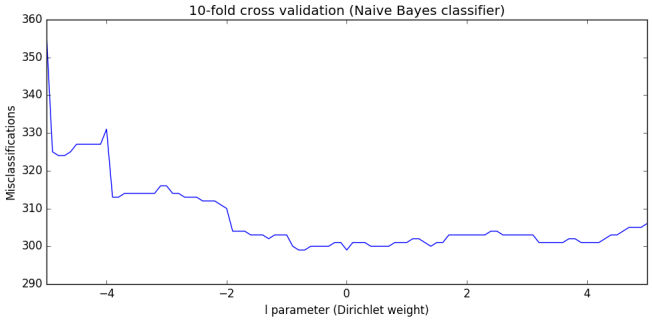

Figure S6. Cross-validation results for Naïve Bayes classifier with l ranging from -5 to +5. The x-axis shows the Dirichlet weight and the y-axis the number of misclassifications.

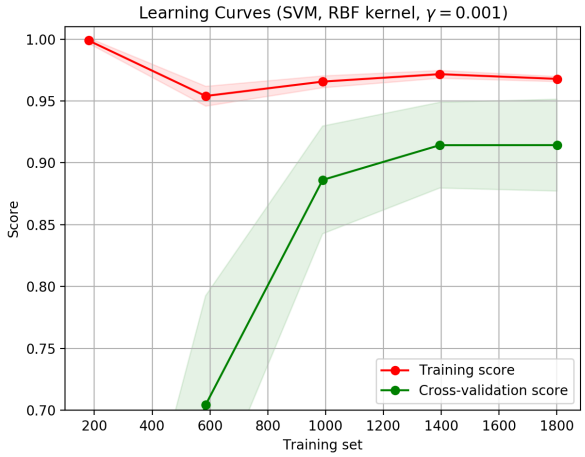

Figure S7. The learning curve of the Support Vector Machine (SVM) classifier is shown for the dataset. The training score is very high at the beginning and decreases and the cross-validation CV score is very low at the beginning and increases. We can observe that roughly after 3-fold CV the training score and the CV score are converging towards their optimal threshold.