Focus Article

Mining data with random forests: current options for real-world applications

Andreas Ziegler[1,2]* and Inke R. König[1]

Random Forests are fast, flexible, and represent a robust approach to mining high-dimensional data. They are an extension of classification and regression trees (CART). They perform well even in the presence of a large number of features and a small number of observations. In analogy to CART, random forests can deal with continuous outcome, categorical outcome, and time-to-event outcome with censoring. The tree-building process of random forests implicitly allows for interaction between features and high correlation between features. Approaches are available to measuring variable importance and reducing the number of features. Although random forests perform well in many applications, their theoretical properties are not fully understood. Recently, several articles have provided a better understanding of random forests, and we summarize these findings. We survey different versions of random forests, including random forests for classification, random forests for probability estimation, and random forests for estimating survival data. We discuss the consequences of (1) no selection, (2) random selection, and (3) a combination of deterministic and random selection of features for random forests. Finally, we review a backward elimination and a forward procedure, the determination of trees representing a forest, and the identification of important variables in a random forest. Finally, we provide a brief overview of different areas of application of random forests.

How to cite this article:
WIREs Data Mining Knowl Discov 2014, 4:55–63. doi: 10.1002/widm.1114

INTRODUCTION

Classification and regression trees (CART) have become popular after publication of the excellent textbook by Breiman and colleagues[1]; for a recent review see Ref 2. They have been extended in several ways so that today dichotomous, unordered categorical and ordered categorical, as well as continuous dependent variables can be handled in addition to survival times.[3] For dichotomous and categorical dependent variables CART cannot only be used for classification but also for the estimation of probabilities. The CART algorithm has the advantage of being simple to interpret, simple to implement, and simple to run. However, it has several disadvantages, and the most important is that CART is unstable, which means that small changes in the data can lead to substantial changes in the resulting tree.

This disadvantage of instability can be overcome by growing not only one tree from a dataset but an entire forest consisting of many trees.[4] The question is, however, how such a set of trees can be grown from a single dataset. One aim of this article is to provide an overview on the different approaches available, and they are described in the next section. They have in common that samples are first bootstrapped, but they differ in the way the bootstrap is performed.

*Correspondence to: ziegler@imbs.uni-luebeck.de

[1]Institut für Medizinische Biometrie und Statistik, Universität zu Lübeck, Universitätsklinikum Schleswig-Holstein, Campus Lübeck, Lübeck, Germany

[2]Zentrum für Klinische Studien, Universität zu Lübeck, Lübeck, Germany

AZ is statistical consultant for the Protagen AG, Dortmund. AZ is an Associate Editor of the Biometrical Journal, Statistics in Medicine, Methods of Information in Medicine. IRK has declared no conflicts of interest for this article.

Next, a tree is grown for every bootstrap sample, and the algorithms differ in how variables are selected for the tree building process. Finally, results from single trees are averaged to form a forest. This random forests approach thus is an extension of bootstrap aggregation, bagging for short.[5] As such, it inherits the properties of bagging, including the improved stability over CART.

Originally, random forests were proposed for classification problems, but they naturally extend to regression problems, and they have been extended to survival data[6,7] as well as the estimation of probabilities for multi-category dependent variables.[8] Another aim of this article is to describe differences in the random forest algorithms for the different outcome variables.

A major strength of random forests is its extensions, and one aim of the article is a summary of these. For example, as in CART,[1] the importance of variables can be quantified,[9] and the importance of variables can be used to substantially reduce the number of features used in the forest in high-dimensional problems with a large number of features. A disadvantage of random forests when compared with CART is that random forests are hard to interpret. Since a real forest consists in some kind of characteristic trees, such as oak trees or birches, a random forest might be described by some characteristic trees. To this end, we summarize the concepts developed so far for identifying representative trees of a forest.

RANDOMLY GENERATING MANY DATASETS FROM A SINGLE DATASET

Figure 1 displays the framework for all random forest algorithms; also see the reviews in Refs 10–17. In this section, only the first two steps shown in Figure 1 are described. This means that we consider different ideas for randomly creating datasets before and in the tree building process. To categorize several solutions existing in the literature, we first interpret the dataset as a two-dimensional array. The rows represent the n independent subjects and the columns contain the p features, also termed independent variables, exogenous variables, or covariates, plus the outcome variable, also termed outcome, dependent variable, endogenous variable, or target variable. In all three approaches, a bootstrap sample is drawn from the dataset. In current implementations, this means that subjects are repeatedly drawn with replacement. In general, the number of subjects to be drawn is n, and this means that on average about two thirds of the subjects are used per bootstrap draw and that on average about one third is out of bag (OOB).

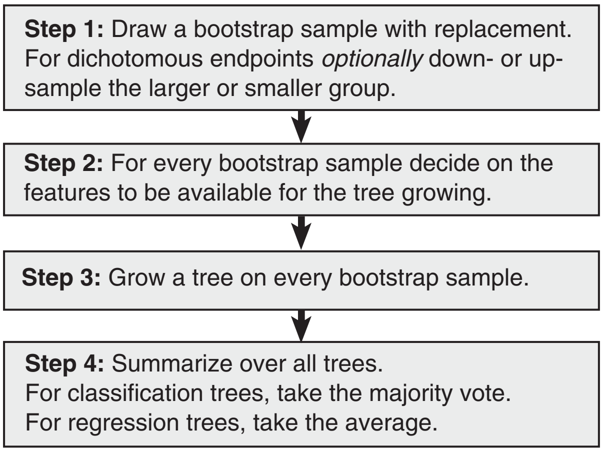

Step 1: Draw a bootstrap sample with replacement. For dichotomous endpoints optionally down- or up-sample the larger or smaller group.
Step 2: For every bootstrap sample decide on the features to be available for the tree growing.
Step 3: Grow a tree on every bootstrap sample.
Step 4: Summarize over all trees. For classification trees, take the majority vote. For regression trees, take the average.

FIGURE 1 | General steps in constructing a random forest.

For dichotomous endpoints, the procedure is commonly modified in case of unbalanced data.[4,18] For example, the ratio of cases and controls in the bootstrap sample may be fixed. More formally, this means that for n_{ca} and n_{co} cases and controls in the original sample, exactly n_{ca} and n_{co} cases and controls are drawn in the bootstrap sample. For substantially unbalanced data, a different strategy may be taken. Specifically, the smaller category may be up-sampled, and the larger one down-sampled to achieve an improved balance of cases and controls. Alternatively, bootstrapping a subsample without replacement may be performed, and it has been demonstrated by Mitchell[18] that choosing the same number of observations from each group, i.e., $n_{ca} = n_{co}$ provides the least unbiased estimate of the prediction error.

So far, different approaches have been described for generating bootstrap samples, i.e., to generate variability in the rows. Next, we consider ways to obtain different versions of the features, i.e., the columns of a datasets:

(a) Use all available features in the tree building process.

(b) Randomly draw a subset of features at every splitting node and use only this subset of features as input to determine the optimal split at the node in a single tree.

(c) The third approach is a combination of the two previous ones. Suppose the dataset consists in some low-dimensional important variables and other high-dimensional data. Low-dimensional data might be clinical variables which have already been proven to be important—typically just a handful but generally not more than 50. High-dimensional data could be data from a microarray, where currently up to 5,000,000 different features can be measured per subject.

Always make available the low-dimensional set of features for the tree building process. Randomly draw a subset from the high-dimensional set of features for the building of a single tree. The algorithm is available in the software package Random Jungle 2.0.[19]

The second approach of random feature input, which is the original random forest approach of Breiman[4] could be reduced:

(b*) Randomly draw a subset of features per bootstrap sample and use only this subset of features to grow a single tree.

Approach (a) is termed bagging trees in the literature. The advantage of (b), i.e., the random feature input approach is that it destroys correlation between trees. The default for the proportion of features to be randomly drawn is denoted as mtry in most implementations, and its default is $\lceil \sqrt{p} \rceil$, where $\lceil \cdot \rceil$ denotes the next largest integer.[19] It has, however, been observed that this number is not optimal,[20] and an alternative approach is to tune mtry so that the error rate is minimal; for details see Ref 19.

The effect of choosing $\lceil \sqrt{p} \rceil$ as default is that the proportion of features available for growing a specific tree decreases with an increasing number of available features. For example, if 25 features are available, the default of mtry selects 20% of the features, while only 10 and 1% of the features are used for growing a specific tree if 100 and 10,000 features are available. If only a small proportion of features is used in the tree growing, it is very likely that completely different or almost different features are selected for two different trees. Individual trees are thus very different. As in real-life committees, this amount of dissimilarity or lack of correlation among committee members provides the chance that there is variability in the votes and that a committee member making a wrong decision will be outvoted by the majority.

If features are not randomly selected as in (a), trees from a forest will be more similar to each other compared with (b) and (b*). This is of importance when the aim is the identification of trees representative for a forest, as will be described below. However, before we can discuss various specific aspects of random forests, we need to consider different approaches for growing a single tree in a random forest given one instance of a bootstrap dataset.

TREE GROWING PROCEDURES AND AVERAGING OVER TREES IN A RANDOM FOREST

In the previous section, we have described steps 1 and 2 shown in Figure 1. In step 3 of the general framework, trees are grown which can be done as in CART. The tree building process varies substantially between the different types of dependent variables, the split criterion used for growing trees, the size of the trees, and the approaches to summarize, i.e., aggregate the results from different trees. However, all procedures have in common that a parent node is always split in exactly two offspring nodes although modifications have been proposed and implemented for CART.[21] In the final step 4 of the general algorithm, results from the different trees are summarized, and the approaches vary here as well.

Random Forests for Classifying Dichotomous Dependent Variables

Random forests have originally been proposed to classify dichotomous-dependent variables, and the standard approach is as follows.[19] For every bootstrap sample, the tree is grown in step 3 by recursively splitting data into distinct subsets so that one parent node has two child nodes. Data are split so that the purity of the data, i.e., the separation of affected and unaffected subjects, in the child nodes is maximized. The standard measure for determining the best splitting feature together with its cutpoint is the Gini index.[19] Alternatives include the deviance, entropy-based information gain, or the area under the curve splitting criterion.[22]

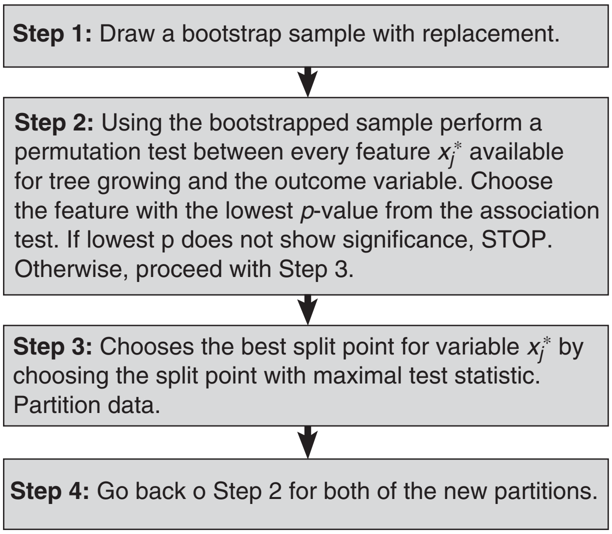
In the standard random forest algorithm, a tree is always grown to its largest extent, i.e., to purity. This means that a terminal node consists in either cases or controls. In the standard algorithm, trees are not pruned although pruning has been done in CART to avoid overfitting. The risk of overfitting in random forests is substantially reduced because of two aspects in the algorithm. The first is that samples are bootstrapped to grow the individual trees, and bootstrap aggregation reduces the risk overfitting, and the second is the even more important component that a small set of features is randomly selected per tree.

Finally, in step 4, results are summarized over trees by taking the majority vote over all trees.

To determine the prediction error, Breiman[4] first grew the random forest on 90% of the data which were randomly selected, and the other 10% of the data were put aside the random forest to obtain a test set error. An alternative approach might be to take the OOB samples of a tree and to drop down every subject in the tree, yielding a prediction of the case/control status. Averaged over all OOB samples from a tree, this approach provides important classification statistics, such as sensitivity, i.e., the proportion of correctly classified cases, and specificity, i.e., the proportion of correctly classified controls, or the prediction error, also termed hit rate, which is the overall proportion of correctly classified subjects. These error estimates can be averaged over all trees from a forest, yielding the mean prediction error. One reviewer pointed out that this error estimation approach matches one usually implemented in software packages, but it differs from Breiman's original definition of the OOB error. In addition, its variability can be estimated using the standard deviation of the error estimates per tree.

One alternative to the random forests is boosting stumps,[23] where stumps are grown instead of large trees. A stump is a tree with one split so that there is only one parent node and two daughter nodes. As in the standard random forest algorithm, the majority vote over all stumps is used for classification.[24] It is computationally faster than random forests but boosted stumps generally perform worse than random forests.[25]

Another important alternative is the conditional inference forest approach.[26] Conditional inference forests differ from random forests in several ways, and the approach consists in three main steps (Figure 2). At a split, the feature with the lowest p-value from a permutation test is selected. With this approach it is possible to address different scales of the features in a natural way. Furthermore, it allows for unbiased selection of the features because the feature is selected in one step and the best split is determined when the variable to split on has been selected.



Step 1: Draw a bootstrap sample with replacement.

Step 2: Using the bootstrapped sample perform a permutation test between every feature x_j^* available for tree growing and the outcome variable. Choose the feature with the lowest p-value from the association test. If lowest p does not show significance, STOP. Otherwise, proceed with Step 3.

Step 3: Chooses the best split point for variable x_j^* by choosing the split point with maximal test statistic. Partition data.

Step 4: Go back o Step 2 for both of the new partitions.

FIGURE 2 | General steps in constructing a conditional inference forest.

Variance, Consistency, Bias, and Rate of Convergence of Random Forests

Although random forests and its standard algorithm are very popular, the properties of random forests were not well investigated because of the complex model building process. Recently, Biau and colleagues[27–29] and Genuer[30] have obtained theoretical findings which complement each other.

Specifically, Biau and colleagues[27–29] studied the properties of random forests using algorithms which is closer to the standard algorithm than any other scheme which has been investigated analytically. The essential difference between Biau's[29] approach and the standard random forest algorithm is that Biau supposes to have a second sample at hand to preserve some independence between the training sample and the optimization procedure. It is unclear whether the results also hold for the standard approach but Biau clearly states that this 'analysis does not appear to be simple'.[29]

A first result is that random forests yield consistent estimates of the regression function.[27,29] Simply speaking this means that random forests estimate what they intend to estimate as long as the sample size is sufficiently large. Under suitable regularity conditions, Biau furthermore showed that the rate of convergence of random forests is higher than the standard minimax convergence rate of other parametric or nonparametric regression methods, such as logistic regression, if <54% of the available features are relevant—termed strong features—for modeling the dependent variable; for details see Ref 29. These results are important for high-dimensional data, and a random forest can therefore be considered a dimension reduction estimator.

Genuer[30] considered a random split random forest algorithm and first derived an upper bound for the variance of the random forest. With this upper bound, he next showed that even in the one-dimensional feature space, purely random forests perform better than CART because they reduce the variance. Furthermore, random forests are consistent in this setting and reach the usual minimax rate of convergence. Finally, even the bias of random forests can be quantified.[30] We stress that, in analogy to Breiman,[31] Genuer[30] considered the random split random forest, where splits are not done to maximize purity but by randomly splitting a feature. For applications, we do not recommend random splitting because it promotes splits near the edges. The tree growing process can be stopped early in the smaller group, and extremely unbalanced trees can result. This can only be corrected by taking into account the nodesize in the tree growing process,[32] which is rather complicated.

Random Forests for Probability Estimation

Further up, we have explained the fundamental ideas for constructing a random forest with dichotomous dependent variables, when the aim is classification. An alternative aim is the estimation of probabilities.[8] This kind of random forest can be considered a generalization of probability estimation trees, abbreviated PET. There are three fundamental differences between the random forests algorithms for classification and for probability estimation. Specifically, in step 3 trees are not grown to purity in probability estimation random forests. However, the results of Biau[29] also hold when terminal nodes are pure. In this case, more trees should be grown for probability estimation. Furthermore, the findings of Biau[29] indicate that the convergence rate is higher if trees are not grown to purity. In software packages, the default is to grow trees to the greatest extent possible with the restriction that terminal nodes generally require a minimum terminal nodesize, and we have used a default of 10%. Alternatively, the optimal terminal nodesize can be tuned.

A second important difference to the standard random forest procedure is that for probability random forests, the split criterion is the mean square error (MSE), which is the standard split criterion for regression trees. At a node, the feature is selected minimizing the MSE over all randomly selected features.

The third difference is in how trees are summarized. For a new subject, the probability that he/she belongs to the case group is determined as follows. First, the proportion of cases is determined in each terminal node and tree. The new subject is now dropped down a tree until its final node, and the proportion of cases in this terminal node is determined. The probability estimate is the average of the proportion of cases over all trees.

This approach can be easily extended to the multi-category case.[8]

Regression Random Forests

For a continuous dependent variable, the approach is analogous to the one described in the previous section for probability estimation of dichotomous dependent variables.[19]

Random Forests for the Analysis of Survival Data

Random survival forests were introduced by Ishwaran and colleagues.[7] The default splitting rule of survival forests splits tree nodes by maximization of the log-rank test statistic.[32] One alternative is the use of a standardized log-rank statistic.[32] In general, trees are grown to purity or, as in random probability forests, some restriction is used on the terminal nodesize so that the number of unique deaths does not drop below the minimum admissible number per node.[32]

In random survival forests, an ensemble cumulative hazard estimate is finally calculated over all trees in the forest.

SPECIAL OPTIONS

In the previous section, we described the construction of random forests. The construction principle allows the simple calculation of the importance of variables, which, in turn, can be used to select the most important variables. This will be detailed in the following sections. In the final part of this section, we describe the identification of representative trees.

Variable Importance

In classification problems, a variable is important if it has a large effect on the classification accuracy. An intuitive approach therefore is as follows. Grow a tree as described above and determine the prediction error in the OOB samples. If variable x_j is included in the tree, randomly permute this predictor variable, thus breaking the association of the feature x_j with the dependent variable. Determine the prediction error in the OOB samples again, and determine the difference in prediction accuracy between the original bootstrap samples and the bootstrap sample after permuting variable x_j. This difference averaged over all trees is often used as a measure of variable importance, and this measure is termed permutation importance. However, various definitions of importance have been implemented in software packages,[19] including Breiman's[4] suggestion to use the OOB error before and after permutation as measure of importance.

This simple approach to the importance has been criticized; for a summary see Ref 9. Specifically, Strobl and colleagues[33] showed that correlated variables were preferred at early splits in a tree and had larger variable importance. To overcome this bias toward continuous and correlated variables, they proposed a conditional permutation importance. The fundamental idea of the conditional importance is to permute not over all subjects but only within a group of subjects with similar variable structure. The conclusions of Strobl and colleagues were contradicted by Nicodemus and Malley,[34] who reported that random forests prefer uncorrelated variables over all splits performed in building all trees in the forest for the Gini index as splitting rule, and the permutation importance was biased. In a third study, Meng and colleagues[35] showed that the stronger the association with the response, the stronger the effect that the predictor correlation has on the performance of random forests.

The contradictions in the conclusions of the different individual studies were investigated by Nicodemus and colleagues.[9] In conclusion, they showed that the conditional permutation importance is preferable in studies with a small number of features. However, in screening studies with a large number of features, such as global gene expression studies or genome-wide association studies, the original permutation importance may be better suited because, in this case, correlation is usually a consequence of physical proximity or biological relatedness in a pathway and may thus be helpful.

Other variable importance measures are available, such as the Gini importance or the scaled importance.[19] However, the Gini importance has been shown to be biased when the number of categories differs between predictor variables or when categories have different frequencies in case of identical numbers of categories,[36] and the scaled importance depends on the scale of the problem.

One important property of the random forest importance measures is that variables of great importance may be highly correlated. For example, when the functional outcome of stroke is studied, both left arm paresis and left leg paresis are important features. They are highly correlated, and in standard logistic regression only one of the two would be kept after feature, i.e., variable selection. However, in random forests both variables are kept as important.

Variable Selection

A different aim is to identify a small number of variables sufficient for a good prediction of the response variable. Díaz-Uriarte and Alvarez de Andrés[37] have proposed a simple backward selection method based on the permutation importance, and its basic idea is as follows. First, a random forest is grown, and the importance is determined for every variable. Unimportant variables which can be ignored will have a low variable importance. Thus, a fixed proportion of the least important variables is dropped next. We now return to step 1, and a new random forest is grown with all variables but the dropped ones. Then, the same proportion of the least important variables is dropped once again. Díaz-Uriarte and Alvarez de Andrés[37] emphasized that in the next steps the variable importance was not recalculated from the fewer available variables because the recalculation might introduce bias. Instead, they dropped a fixed proportion of variables using the variable importance determined from all available variables. Finally, the random forest was chosen which had the lowest OOB error.

One question is how large the proportion of dropped variables should be. Díaz-Uriarte and Alvarez de Andrés[37] concluded that the parameter fraction dropped can be adjusted to modify the resolution of the number of variable selected. Smaller fractions dropped lead to finer resolution in the examination of the number of genes, but to higher computing times. In our own applications, we therefore include a dependency of this proportion on the number of available features. For example, after random forest run k, the least important $1/(1+k)$ features are dropped. As a result 50% of the features are dropped after the first run, while only an additional 20% are dropped after the fourth run. Of course, these percentages depend on the number of features available for the first run, and the $1/(1+k)$ is only used for high throughput studies, i.e., $p \gg n$ problems.

Several alternative variable strategies have been proposed in the literature for use with random forests,[38–41] and one interesting alternative is the approach of Genuer et al.[38] They considered a two-step procedure, where they grew several random forests and calculated the standard deviation of the variable importance for each variable over the random forests. Only those variables with a high standard deviation of the variable importance were retained. In the second step, these remaining variables were included in a forward selection procedure as follows, where a random forest was run in each forward selection step. The first random forest included only the most important variable. The second most variable was added to the random forest if it decreased the error substantially; for details see Ref 38. The evaluation of the error needs to be made carefully with this approach because ranking and variable selection are done on the same set of observations.

Representative Trees

One of the strengths of a single CART is that it is simple to interpret: The relevant characteristics are the features included in the tree; the earlier a variable appears in a tree, the more important it is. Furthermore, differences between the characteristics lead to different forecasts. With forests, this simplicity is lost because many trees, i.e., a forest, have to be considered simultaneously. An important question therefore is whether all trees from a forest are required or whether some trees represent some kind of 'species' of trees and whether a small set of trees is sufficient.[42]

To this end, we consider two trees from the forest and drop a subject down both trees. We can easily determine whether the tree classifications coincide or differ. This can be repeated for all subjects, and the trees are more similar, the higher the proportion of agreements in the classifications between them. Finally, this can be done for all pairs of trees from a forest, and the trees are most similar that have the greatest proportion of agreement. In fact, a cluster analysis could be performed over all pairwise distance measures to estimate the averaged distances between trees from the forest.

If the aim of the study is not classification but estimation, such as probabilities, a similar approach can be taken; for technical details see Ref 43. An important aspect is whether features used for the tree construction are randomly selected or whether all features are used. Specifically, Banerjee et al.[43] did not employ the random feature input approach but used all features for tree construction. As described in detail in the section on randomly generating many datasets from a single dataset, we expect substantial more variability between trees if features are selected at random, and this diversity will increase with the number of features available.

Although it is appealing to define similarity of tree through similarity of classification results, probability estimates or regression estimates, an alternative approach would be to define the similarity of trees through the similarity of features used in the tree growing process. However, the authors are not aware of any published work which has considered this idea in some detail.

A different approach has been followed by Zhang and Wang,[42] who investigated methods to reduce the number of trees coming from a forest. They aimed to construct the smallest random forest that has the same predictive accuracy as the large random forest. To this end, Zhang and Wang[42] looked at the predictions of a random forest using all trees from the forest and compared it to the predictions when one tree was left out. The tree with the least changes in the predictions when left out is the least important one, and it is eliminated. Other measures of similarity to identify the smallest random forest are possible.[42]

CONCLUSION

Random forests have well developed over the past years. They are widely accepted as one machine learning approach among others for a wide variety of tasks. Random forests have been applied to substantially different areas of application, such as the analysis of gene expression microarrays,[13] genome-wide association studies and the analysis of gene–gene interactions, for a review see Ref 44, or the prediction of protein–protein interactions.[45] Other areas of application include credit scoring,[46] weather forecasting,[47] or land cover classification.[48]

This great variety of areas of application is easily explained by the great flexibility of random forests. Furthermore, several simple to use implementations are available, although users should be aware that different options in random forest software packages can yield substantial differences in results. Furthermore, even the same options used in different software packages can yield differences.[19]

One advantage of random forests over other machine learning approaches is its computational speed. Additional pros include the availability of importance measures and convenient feature selection procedures. The latter also allows a combination of random forests with other machine learning methods. Specifically, the most important features might be selected by use of random forests in the first step. With the lower number of features, other, more computer time consuming machine learning approaches might be used in the second step. However, users should always keep in mind the risk of overfitting when features are selected and machines are trained on a single dataset. The aspect of model validation should not be ignored.

REFERENCES

1. Breiman L, Friedman J, Olshen RA, Stone CJ. *Classification and Regression Trees*. Chapman & Hall/CRC: Boca Raton, FL; 1984.
2. Loh W-Y. Classification and regression trees. *WIREs Data Mining Knowl Discov* 2011, 1:14–23.
3. Gordon L, Olshen RA. Tree-structured survival analysis. *Cancer Treat Rep* 1985, 69:1065–1069.
4. Breiman L. Random forests. *Mach Learn* 2001, 45:5–32.
5. Breiman L. Bagging predictors. *Mach Learn* 1996, 24:123–140.
6. Hothorn T, Bühlmann P, Dudoit S, Molinaro A, van der Laan MJ. Survival ensembles. *Biostatistics* 2006, 7:355–373.

7. Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS. Random survival forests. *Ann App Statist* 2008, 2:841–860.
8. Kruppa J, Liu Y, Biau G, Kohler M, König IR, Malley JD, Ziegler A. Probability estimation with machine learning methods for dichotomous and multi-category outcome: theory. *Biom J,* In press.
9. Nicodemus KK, Malley JD, Strobl C, Ziegler A. The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC Bioinformatics* 2010, 11:110.
10. Boulesteix A-L, Janitza S, Kruppa J, König IR. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *WIREs Data Mining Knowl Discov* 2012, 2:493–507.
11. Chen CC, Schwender H, Keith J, Nunkesser R, Mengersen K, Macrossan P. Methods for identifying SNP interactions: a review on variations of Logic Regression, Random Forest and Bayesian logistic regression. *IEEE/ACM Trans Comput Biol Bioinform* 2011, 8:1580–1591.
12. Chen X, Ishwaran H. Random forests for genomic data analysis. *Genomics* 2012, 99:323–329.
13. Cutler A, Stevens JR. Random forests for microarrays. *Meth Enzymol* 2006, 411:422–432.
14. Sun YV. Multigenic modeling of complex disease by random forests. *Adv Genet* 2010, 72:73–99.
15. Strobl C, Malley J, Tutz G. An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychol Methods* 2009, 14:323–348.
16. Touw WG, Bayjanov JR, Overmars L, Backus L, Boekhorst J, Wels M, van Hijum SA. Data mining in the life sciences with random forest: a walk in the park or lost in the jungle? *Brief Bioinform* 2013, 14:315–326.
17. Verikas A, Gelzinis A, Bacauskiene M. Mining data with random forests: a survey and results of new tests. *Pattern Recogn* 2011, 44:330–349.
18. Mitchell MW. Bias of the random forest out-of-bag (OOB) error for certain input parameters. *Open J Statist* 2011, 1:205–211.
19. Schwarz DF, König IR, Ziegler A. On safari to Random Jungle: a fast implementation of Random Forests for high-dimensional data. *Bioinformatics* 2010, 26:1752–1758.
20. Genuer R, Poggi JM, Tuleau C. Random Forests: some methodological insights. arXiv: 0811.3619. 2008. Available at: http://hal.inria.fr/inria-00340725/en/;
21. Al Ghoson AM. Decision tree induction & clustering techniques in SAS Enterprise Miner, SPSS Clementine, and IBM Intelligent Miner – a comparative analysis. *Int J Manag Inf Syst* 2010, 14:57–70.
22. Rokach L, Maimon O. Decision tress. In: Maimon O, Rokach L, eds. *The Data Mining and Knowledge Discovery Handbook*. New York: Springer; 2005, 165–192.
23. Schapire RE. A brief introduction to boosting. In: Dean TL, ed. *IJCAI-99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, vol. 2. San Francisco, CA: Morgan Kaufmann; 1999, 1401–1406.
24. Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting. *Ann Stat* 2000, 28:337–407.
25. Caruana R, Niculescu-Mizil A. An empirical comparison of supervised learning algorithms. In: Cohen W, Moore A, eds. *Proceedings of the 23rd International Conference on Machine Learning*. New York: Association for Computing Machinery; 2006, 161–168.
26. Hothorn T, Hornik K, Zeileis A. Unbiased recursive partitioning: a conditional inference framework. *J Comput Graph Stat* 2006, 15:651–674.
27. Biau G, Devroye L, Lugosi G. Consistency of random forests and other averaging classifiers. *J Mach Learn Res* 2008, 9:2039–2057.
28. Biau G, Devroye L. On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. *J Multivariate Anal* 2010, 101:2499–2518.
29. Biau G. Analysis of a random forests model. *J Mach Learn Res* 2012, 13:1063–1095.
30. Genuer R. Variance reduction in purely random forests. *J Nonparametric Stat* 2012, 24:543–562.
31. Breiman L. Some infinity theory for predictor ensembles. 2000. Available at: http://digitalassets.lib.berkeley.edu/sdtr/ucb/text/579.pdf.
32. Ishwaran H, Kogalur UB. Random survival forests for R. *R-News* 2007, 7:25–31.
33. Strobl C, Boulesteix AL, Zeileis A, Hothorn T. Bias in random forest variable importance measures: illustrations, sources and a solution. *BMC Bioinformatics* 2007, 8:25.
34. Nicodemus KK, Malley JD. Predictor correlation impacts machine learning algorithms: implications for genomic studies. *Bioinformatics* 2009, 25:1884–1890.
35. Meng YA, Yu Y, Cupples LA, Farrer LA, Lunetta KL. Performance of random forest when SNPs are in linkage disequilibrium. *BMC Bioinformatics* 2009, 10:78.
36. Boulesteix AL, Bender A, Lorenzo Bermejo J, Strobl C. Random forest Gini importance favours SNPs with large minor allele frequency: impact, sources and recommendations. *Brief Bioinform* 2012, 13:292–304.
37. Díaz-Uriarte R, Alvarez de Andrés S. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics* 2006, 7:3.
38. Genuer R, Poggi JM, Tuleau-Malot C. Variable selection using random forests. *Pattern Recogn Lett* 2010, 31:2225–2236.

39. Kursa MB. Robustness of the random forest-based gene selection methods. 2013. Available at: http://arxiv.org/abs/1305.4525.

40. Rodin AS, Litvinenko A, Klos K, Morrison AC, Woodage T, Coresh J, Boerwinkle E. Use of wrapper algorithms coupled with a random forests classifier for variable selection in large-scale genomic association studies. *J Comput Biol* 2009, 16:1705–1718.

41. Sandri M, Zuccolotto P. Variable selection using random forests. In: Zani S, Cerioli A, Riani M, Vichi M, eds. *Proceedings of the Meeting of the Classification and Data Analysis Group (CLADAG) of the Italian Statistical Society, University of Parma, June 6–8, 2005*. Heidelberg, Germany: Springer; 2006, 263–270.

42. Zhang H, Wang M. Search for the smallest random forest. *Stat. its interface* 2009, 2:381.

43. Banerjee M, Ding Y, Noone AM. Identifying representative trees from ensembles. *Stat Med* 2012, 31:1601–1616.

44. Kruppa J, Ziegler A, König IR. Risk estimation and risk prediction using machine learning methods. *Hum Genet* 2012, 131:1639–1654.

45. Lin HF, Juo SH, Cheng R. Comparison of the power between microsatellite and single-nucleotide polymorphism markers for linkage and linkage disequilibrium mapping of an electrophysiological phenotype. *BMC Genet* 2005, 6:S7.

46. Brown I, Mues C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Exp Syst Appl* 2012, 39: 3446–3453.

47. Deloncle A, Berk R, D'Andrea F, Ghil M. Weather regime prediction using statistical learning. *J Atmos Sci* 2007, 64:1619–1635.

48. Pal M. Random forest classifier for remote sensing classification. *Int J Remote Sens* 2005, 26: 217–222.