LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Anne-Laure Boulesteix, Silke Janitza, Jochen Kruppa, Inke R. König

Overview of Random Forest Methodology and Practical Guidance with Emphasis on Computational Biology and Bioinformatics

Technical Report Number 129, 2012
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Overview of Random Forest Methodology and Practical Guidance with Emphasis on Computational Biology and Bioinformatics

Anne-Laure Boulesteix[1*] Silke Janitza[1]
Jochen Kruppa[2] Inke R. König[2]

July 25th 2012
pre-review version of a manuscript accepted for publication in
WIREs Data Mining & Knowledge Discovery

[1] Institut für Medizinische Informatik, Biometrie and Epidemiologie. Ludwig-Maximilians-Universität München, Germany.

[2] Institut für Medizinische Biometrie und Statistik, Unversität zu Lübeck, Universitätsklinikum Schleswig-Holstein, Campus Lübeck, Germany.

Abstract

The Random Forest (RF) algorithm by Leo Breiman has become a standard data analysis tool in bioinformatics. It has shown excellent performance in settings where the number of variables is much larger than the number of observations, can cope with complex interaction structures as well as highly correlated variables and returns measures of variable importance. This paper synthesizes ten years of RF development with emphasis on applications to bioinformatics and computational biology. Special attention is given to practical aspects such as the selection of parameters, available RF implementations, and important pitfalls and biases of RF and its variable importance measures (VIMs). The paper surveys recent developments of the methodology relevant to bioinformatics as well as some representative examples of RF applications in this context and possible directions for future research.

*Corresponding author. Email: boulesteix@ibe.med.uni-muenchen.de.

1 Introduction

In only ten years, the Random Forest (RF) [6] algorithm has evolved to a standard data analysis tool in bioinformatics. By "bioinformatics", we mean the application of computer science and information technology to the field of biology and medicine. RF methodology is used to address two main classes of problems: to construct a prediction rule for a supervised learning problem and to assess and rank variables with respect to their ability to predict the response. The latter is done by considering the so-called variable importance measures (VIMs) that are automatically computed for each predictor within the random forest algorithm. In particular, RF VIMs are believed to successfully identify predictors involved in interactions, i.e. predictors which can predict the response only in association with one or several other predictor(s). After sensible validation, the resulting prediction rule can then be applied, for instance, in clinical practice [42]. As far as these two tasks (prediction and predictor assessment) are concerned, RF offers specific features that makes it attractive for bioinformatics applications. It can cope with high-dimensional data (the so-called "$n \ll p$ curse") and can even be applied in difficult settings with highly correlated predictors. It is not based on a particular stochastic model and can also capture non-linear association patterns between predictors and response. It does not require the user to specify a model underlying the data. Considering the complexity of modern high-throughput "omics" data, these features are usually considered as important advantages in this context.

This paper synthesizes ten years of RF development with emphasis on bioinformatics and computational biology. Special attention is given to practical aspects such as the selection of parameters in the RF algorithm to provide helpful guidelines for applications. Essential pitfalls and shortcomings of RF and its VIMs are discussed as well as alternative approaches to circumvent these problems. For more theoretical details and reviews covering other aspects of RF, we refer to the literature. For example, Malley et al. [48] depict the theory in a broad context, Goldstein et al. [32] describe in detail the RF algorithm and its applications to genetic epidemiology, Chen et al. [18] give an extensive overview of applications of recursive partitioning to bioinformatics, and Verikas et al. [76] survey RF applications and their performance in comparison with other methods in a more general context. This paper

is structured as follows. After a short overview of the main RF variants, available implementations of RF and parameter choice issues are briefly reviewed. The paper then surveys recent developments of the methodology in bioinformatics as well as some representative examples of RF applications in this context.

2 Random forest variants and parameters

2.1 Random forests and conditional inference forests

RF is a classification and regression method based on the *aggregation* of a large number of decision trees. Specifically, it is an *ensemble* of trees constructed from a training data set and internally validated to yield a prediction of the response given the predictors for future observations. There are several variants of RF which are characterized by 1) the way each individual tree is constructed, 2) the procedure used to generate the modified data sets on which each individual tree is constructed, 3) the way the predictions of each individual tree are aggregated to produce a unique consensus prediction.

The general functioning of the RF algorithm is depicted in Figure 1. In the original RF method suggested by Breiman et al. [8], each tree is a standard Classification or Regression Tree (CART) that uses the so-called Decrease of Gini Impurity (DGI) as a splitting criterion and selects the splitting predictor from a randomly selected subset of predictors (the subset is different at each split). Each tree is constructed from a bootstrap sample drawn with replacement from the original data set, and the predictions of all trees are finally aggregated through majority voting. This version of RF is implemented in most of the available software described below.

An important feature of RF is its out-of-bag (OOB) error. Each observation is an OOB observation for some of the trees, i.e. it was not used to construct them and can thus be considered as an internal validation data set for these trees. The OOB error of the RF is simply the average error frequency obtained when the observations from the data set are predicted using the trees for which they are OOB. Through this internal validation, the error estimation is less optimistic and usually considered as a good estimator of the error expected for independent data.

Although this is by far the most widely applied version, this standard

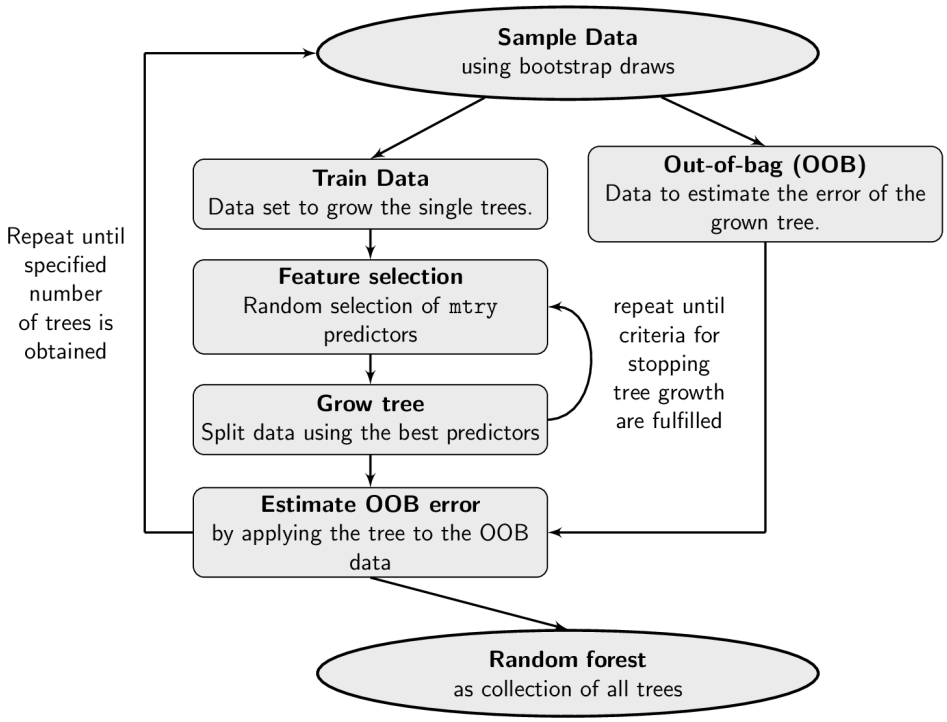

Figure 1: Random Forest Algorithm

RF method has an important pitfall. In the split selection process, predictors may be favored or disfavored depending on their scale of measurement or, in the case of categorical predictors, on their number of categories. This is described below in more detail. The alternative class of decision trees developed by Hothorn et al. [36] and Strobl et al. [68] addresses this issue through the principle of conditional hypothesis testing. The forests built based on these trees are correspondingly denoted as conditional inference forests (CIF). At each split, each candidate predictor is globally tested for its association with the response and a p-value is computed. This p-value is conditional, which means that it reflects the probability to obtain such a high (or a higher) association with the response given the marginal distributions of the response and of the considered predictor. Hence, in CIF splitting is based on an essentially unbiased splitting criterion that automatically adjusts for different marginal distributions of the predictors and thus does not share the above pitfall. In addition to standard regression and classification problems, the CIF methodology also directly addresses the case of censored survival response variables.

2.2 Gini importance vs. permutation importance

The standard RF computes two different VIMs for each predictor: the Gini VIM and the permutation VIM, see Goldstein et al. [32] for a detailed overview. In a few words, the Gini VIM of a predictor of interest is the sum of the DGI criteria of the splits that are based on this predictor, scaled by the total number of trees in the forest. An "important" predictor is often selected for splitting and yields a high DGI when selected, leading to a high Gini VIM. In contrast, the permutation VIM is directly based on the prediction accuracy rather than on the splitting criterion. It is defined as the difference between the OOB error resulting from a data set obtained through random permutation of the predictor of interest and the OOB error resulting from the original data set. Permutation of an "important" predictor is expected to increase the OOB error, leading to a high permutation VIM.

While the permutation VIM is more frequently used in practice, the question of the choice of the VIM type and the properties of these VIMs are still subjects of current research. The CIF algorithm, that does not use the decrease of Gini impurity as a splitting criterion, computes only the permutation VIM. If all predictors are non-informative to the prediction problem at hand, they are expected to have equally low VIMs. Any pattern that deviates from this indicates a systematic bias. Unfortunately, VIMs derived from standard RF and – to a lesser extent – from CIF are (sometimes strongly) biased in many scenarios. Due to a bias, a non-informative predictor with positively biased VIM may seemingly outperform a moderately informative predictor with negative bias. Hence, systematic biases should be avoided whenever possible, because they may lead to erroneous rankings of the predictors.

Biases of the Gini VIM

The perhaps most obvious bias primarily affects the Gini VIM in RF and is related to the number of candidate splits in predictors. A categorical predictor with K categories yields $2^{K-1} - 1$ possible splits, while a metric predictor without ties yields $n - 1$ candidate splits (with n denoting the sample size). The more candidate splits, the more likely it is that at least one of them yields a good splitting criterion – by chance. Hence, RF selects predictors with many categories or metric predictors more often in the tree

building process than predictors with few categories [68]. This so-called "selection bias" transfers directly into a "Gini VIM bias", since the Gini VIM grows with its occurrence of a predictor in the trees. Moreover, even if there were no selection bias (i.e. if all predictors were selected equally frequently, for instance because only one candidate predictor is considered at each split), the Gini VIM would be biased since it is directly computed from the Gini criterion itself, which is on average larger for predictors with more categories.

The selection bias at work in RF, however, does not lead to a bias of the permutation VIM. The reason for this is that the permutation VIM is based on the decrease of accuracy resulting from permutation for OOB observations. Even if non-informative predictors with many candidate splits are selected more often due to the selection bias, they have no chance to improve the average OOB accuracy, and thus do not receive higher VIMs. The higher frequency of selection of predictors with many candidate splits, however, results in a higher variance of the permutation VIM. Finally, let us point out that CIF uses an unbiased splitting criterion and avoids both the systematic bias and the increased variance for predictors with many candidate splits.

A similar bias is also observed in the case of predictors with the same number of categories but different category sizes [55, 4]. In genetic epidemiology, non-informative single nucleotide polymorphisms (SNPs) with large minor allele frequency (MAF) are systematically favored by the Gini VIM over non-informative SNPs with small MAF, potentially yielding misleading rankings of the candidate SNPs. The use of the permutation VIM, that is much less affected by this type of bias, is thus recommended in the case of SNPs with very different MAFs. Correlation between predictors may also induce a bias [57]. If all predictors are non-informative, predictors that are highly correlated with some of the other predictors tend to receive smaller VIMs than uncorrelated predictors. This effect affects both permutation and Gini VIM, but is particularly pronounced for the Gini VIM [57].

Cases where Gini VIM may be preferred

The bias affecting the Gini VIM is related to the type of the predictors. In a case where all predictors are continuous without ties and mutually uncorrelated, the Gini VIM is not expected to be biased. It can even identify

informative predictors more accurately than permutation VIM in specific cases. The first case where the permutation VIM may partly fail is when the response class is a categorical variable with strongly unbalanced categories. This may happen, e.g., when much more controls than cases are considered in an epidemiological study. In this case, the majority class (the control class in our example) is predicted for almost all terminal nodes, no matter whether the predictors are permuted or not. Hence, the OOB error is not expected to be strongly affected by permutation, and permutation VIMs are expected to approximate zero for all predictors and to be unreliable. A discussion on how to handle unbalanced data is given below. The second case where Gini VIM is expected to yield better results is when the signal-to-noise ratio is low (see [32] and references therein). This may be related to the higher instability of the permutation VIM [13].

2.3 Parameters in bioinformatics applications

This section describes the main parameters in RF and CIF and gives tentative recommendations for their choice in bioinformatics applications.

2.3.1 Number of trees

The number of trees in the forest should quite generally increase with the number of candidate predictors, so that each predictor has enough opportunities to be selected. If we have, say, 20,000 predictors (for instance gene expressions) in the data set, the number of trees should by no way be set to the default value of 500. Roughly speaking, it should be chosen such that two independent runs of the "random algorithm" yield very similar results. It is recommended to try several increasing values and to stop increasing as soon as the measures of interest (such as prediction error or VIM) stabilize. Note that a smaller number of trees might yield the same prediction accuracy as a larger number but less reliable VIMs [32]. To conclude, note that the number of trees is not a real parameter in the sense that a larger value always yields more reliable results than a smaller one.

Number of candidate predictors

In contrast, the number of candidate predictors considered at each split is a real parameter in the sense that its optimal value depends on the data

at hand. On the one hand, the default value $\sqrt{p}$ for classification and $p/3$ for regression (with p as the total number of predictors) recommended by Breiman [6] might be too small, especially in the presence of a large number of noise predictors. That is because, in this case, it often happens that all $\sqrt{p}$ resp. $p/3$ randomly selected predictors are non-informative, yielding inaccurate trees. On the other hand, in a scenario with many informative predictors with different strengths, a small value might "give a chance" to predictors with moderate effects that would otherwise have been masked by stronger predictors. If the value is small, predictors with moderate effects sometimes happen to be the best out of the selected candidate predictors and may contribute to prediction.

2.3.2 Size of the trees

The parameters controlling the size of the trees should also be seen as tuning parameters, but their influence on the results is expected to be lower than the influence of the number of parameters selected at each split. Moreover, they are not known to introduce a systematic VIM bias in favor of a particular type of predictors. There are several parameters that can be used to control the size of trees, for example the minimal size that a node should have to be split, the maximal number of layers or a threshold value for the splitting criterion.

2.3.3 Size of terminal nodes

Although they are also related to tree size, the parameters controlling the minimal size of the terminal nodes are treated separately because they may introduce any systematic bias, especially in the context of genetic association studies. A large value may prevent the selection of those categorical predictors that have, say, a large and a small category. That is because the small category would yield a too small terminal node. Even if it is selected as the best predictor according to the splitting criterion, such a predictor would be excluded because it yields a terminal node smaller than the prespecified size. Our advice is to set this parameter to a small value and to rather control the size of the trees using the parameters discussed in the previous section.

2.3.4 Resampling scheme

A RF option that is often ignored is the resampling scheme of the observations on which a tree is built. Trees are built based on bootstrap samples drawn with or without replacement. Strobl et al. [68] show that the option with replacement leads to a VIM bias in favor of predictors with many categories even if the trees are built using an unbiased criterion. Sampling without replacement eliminates this bias. Since there is to our knowledge no inconvenience in the use of subsampling instead of bootstrap sampling, we recommend to systematically use sampling without replacement. The size of the subsamples is then an additional parameter, which can for example be set to 0.632 in analogy with the average proportion of observations included in a bootstrap sample drawn with replacement [68].

2.3.5 Summary

Except for the number of trees that should be as large as computationally feasible and sampling without replacement, the other parameters can be selected based on the OOB error frequency, as suggested by Goldstein et al. [32]. RF are built using different parameter values (or combinations of parameter values) successively, and for each RF the OOB error frequency is computed. The (combination of) parameter value(s) yielding the lowest error is then selected. However, it needs to be kept in mind that this tuning of parameters increases the necessity of externally validating the resulting prediction rule [43].

3 Implementations and example code

3.1 Implementations

A brief overview of available RF implementations is given in Table 1, while more details can be found in Table 2. In addition, a variant of RF handling censored survival outcome as response is available in the R package **randomSurvivalForest** [38]. In some implementations, RF is one tool among many others, which can be a drawback. The documentation and the available tuning parameters may be very sparse with the consequence that users with limited programming knowledge have no clear insight into the framework and capability of the offered RF application. A summary

Name	RF only	MT	System	Code
ALGLIB [3]	no	no	Win/Unix	C++
cforest function in R package party [37]	no	yes	all	C++/S
FastRandomForest [71]	yes	yes	all*	Java
Orange [22]	no	no	Win/Unix/Mac	C++/Python
PARF - Parallel RF Algorithm [75]	yes	yes	all*	F90
Random forest [9]	yes	no	all*	F77
Randomforest-matlab [39]	yes	no	all*	C/C++/
randomForest-R package [45]	no	yes	all	C++/S
Random Jungle [64]	yes	yes†	Win/Unix	C++
RT-Rank [53]	yes	yes	Unix*	C++/Python
Waffles [28]	no	no	Win/Unix/Mac	C++
Weka 3 [33]	no	no	all*	Java

Table 1: Overview of random forest implementations. RF only - indicates whether this is a program only for RF analysis (yes) or part of a broader software package (no), MT - Multithreading ability, *provided that a compiler is available; †only for UNIX machines available

of important arguments for the R packages **randomForest** and **cforest** is shown in Tables 3 and 4.

Table 2: Features and short descriptions of random forest implementations listed in Table 1.

Name	Description	Main features
ALGLIB	Portable open source analysis and data processing library including random decision forest variants as modifications of RF. Until now only classification is possible.	Standard tuning parameters are available (`NTrees` equals `ntree` and `NFeatures` equals `mtry`). Moreover the size of the part of the training can be controlled. Further options are limited.
cforest function in R package party	Implements the CIF methodology, i.e. uses conditional inference trees as base learners; strongly differs from other RF implementations.	Many tuning parameters (see Table 4).
FastRandomForest	Re-implementation of RF in Weka environment to achieve speed and memory optimization.	Add-on to Weka 3 for fast RF implementation adding multithreading to RF and improving speed and memory usage. Only classification so far.
Orange	Open source data visualization software with a GUI. Different data analysis tools can be selected by drag and drop of a widget tool approach.	Many available tuning parameters, e.g.: number of trees, number of features, and parameters controlling the tree size. By now only classification is available.
PARF	Command line open source RF implementation for multiple threading. Linkage with gnuplot is also provided enabling visualization of the generated outcome.	Many tuning parameters. Options to control the growing of the forest, the analysis of the training data, and the data classification and regression.

Name	Description	Main features
Random forest	Original code by Breiman and Cutler. All other RF implementations refer to this original source.	Many tuning parameters. Slow F77 code. Newer implementation offering multithreading. Classification and regression possible.
Randomforest-matlab	MATLAB and stand-alone implementation of Andy Liaw's R package **randomForest**.	Classification and regression is practicable and nearly all tuning parameters like in the corresponding R package are available.
randomForest-R package	Based on the original code by Breiman and Cutler; implements variable importances and proximity measures.	Many tuning parameters (see Table 3).
Random Jungle	Implements all features of the reference implementation **randomForest** such as various tuning parameters, prediction of new data sets using previously grown forests, sample proximities and imputation. Additionally implements backward variable elimination.	Different VIMs, conditional inference forests, prediction and different types of CART. User-defined tuning parameters. Special version allowing the analysis of genomic data in a memory sparing way.
RT-Rank	Open source project for various machine learning algorithms including gradient boosting, RF and IGBRT (Initialized Gradient Boosted Regression Trees) as a novel approach.	Originated from the "Yahoo Learning to Rank Challenge". Only standard tuning parameters (e.g. number of trees and number of candidate splitting predictors).

Parameter	Acronym	Default (classification resp. regression)
No. of trees	`ntree`	500
No. of candidate predictors	`mtry`	$\sqrt{p}$ resp. $p/3$
Maximum no. of terminal nodes	`maxnodes`	not restricted
Minimum size of terminal nodes	`nodesize`	1 resp. 5
Resampling scheme	`replace`	`TRUE`

Table 3: Important arguments to the function `randomForest` from the R package **randomForest**.

Name	Description	Main features
Waffles	Licensed under the GNU Lesser General Public License, uses a command line interface and additionally offers a graphical wizard tool; can be compiled across many platforms and provides many supervised learning methods, data transformation etc.	Includes the regression and classification algorithm by Breiman with slight adjustments by the developer.
Weka 3	Collection of machine learning algorithms selectable from a GUI. Contains many data tools for clustering, classification and visualization. For the usage of RF the extension FastRandomForest is recommended.	Only classification trees (regression trees not yet provided). Few usable tuning parameters. Difficult access to the RF documentation.

Parameter	Acronym	Default
No. of trees	`ntree`	500
No. of candidate predictors	`mtry`	5
P-value threshold	`mincriterion`	0.95
Minimum size of node to be split	`minsplit`	20
Maximal no. of layers	`maxdepth`	not restricted
Minimum size of terminal nodes	`minbucket`	7
Resampling scheme	`replace`	`TRUE`

Table 4: Important arguments to the function `cforest` from the R package **party** .

3.2 Example code

The following RF example consists of two parts: an example code using the R package **randomForest** and an example code using the Random Jungle implementation [64]. The authors assume that the reader is familiar with R including the installation of additional packages and the general data processing. Readers are referred to the web project Quick-R (`http://www.statmethods.net/`) for a brief insight to the R statistical software. The "Breast Cancer Wisconsin (Original) Data Set" [49] from the UCI repository (`http://archive.ics.uci.edu/ml/`) is used as an example data set. It includes $n = 699$ observations and nine predictors. The response variable (`class`) is binary (benign versus malignant).

3.2.1 Example code 1: RF in R package randomForest

The `randomForest` call automatically distinguishes between a classification and a regression RF based on the type of the response variable. A response of type `factor` leads to a classification RF while a `numeric` response leads to a regression RF. See Figure 2 for the visualized results of RF.

```
library(randomForest)
cancerDfRaw <- read.table("http://archive.ics.uci.edu/ml/machine-
   learning-databases/breast-cancer-wisconsin/breast-cancer-
   wisconsin.data", sep = ",", header = FALSE)
names(cancerDfRaw) <- c("ID", "clumpThickness", "uniSize",
   "uniShape", "adhesion", "cellSize", "nucleiBare",
   "chromatin", "nucleiNormal", "mitoses", "class")
cancerDf <- cancerDfRaw[,-1]   # remove ID
```

```
## do classification
cancerDf$class <- as.factor(cancerDf$class)
classRFCancer <- randomForest(class~.,
                              data=cancerDf, mtry=3, ntree=500)
## do regression (not recommended)
cancerDf$class <- as.numeric(cancerDf$class)
regRFCancer <- randomForest(class~.,
                            data=cancerDf, mtry=3, ntree=500)
## get importance measurements
impClass <- as.data.frame(classRFCancer$importance)
impReg <- as.data.frame(regRFCancer$importance)
```

3.2.2 Example code 2: Random Jungle

The Random Jungle example is plugged into the R environment to provide better data handling. A compiled version of Random Jungle can be downloaded from the project page `http://randomjungle.de` for several operating systems. The help pages of Random Jungle give a full overview of the available features and can be called using `rjungle -h`. In the following example we use again the data set prepared as in example code 1. The following code can be used to perform the same analysis.

```
write.table(cancerDf, file = rjungleInFile,   # get rid of index
            row.names = FALSE, quote = FALSE) # and quotes
rjungle <- file.path("to/rjungle/executable")
rjungleCMD <- paste(rjungle,
                    "--file", rjungleInFile,
                    "--treetype 1", # 1 = classification
                                    # 3 = regression
                    "--ntree 500",  # number of grown trees
                    "--mtry 3",     # number of used features
                    "-v",           # verbose; nicer output
                    "-D class",     # response variable name
                    "--outprefix", rjungleOut)
try(system(rjungleCMD)) # send command string to system and
                        # handle error-recovery
```

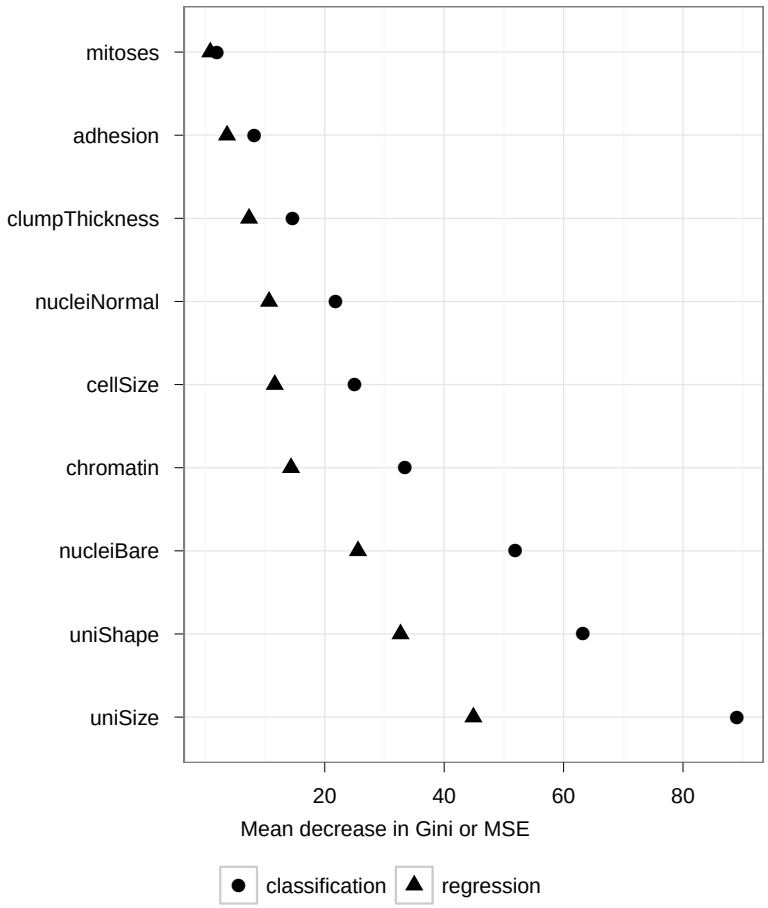

Figure 2: Variable importance of the example data set. The plot includes the ordered results of `impClass` and `impReg` of example 1. Mean decrease in Gini refers to classification, MSE to the regression mode. Note that both modes occasional deliver the same order of the variables and range of values.

4 Recent developments in bioinformatics

4.1 Dealing with correlated predictors

The problem of correlated predictors and how they are/should be handled by RFs has given place to a large body of literature in the last few years. While correlation between predictors does not usually have much influence on prediction accuracy, VIMs can be strongly affected. In some applications it might make sense to circumvent this issue at the data level by selecting one or few representative predictor(s) out of a block of strongly correlated predictors, a procedure referred as 'LD pruning' in genetic epidemiology. However, the results typically depend on the sample size, and to reduce the data set to strictly independent variables is not desired. Then, in most applications there will be some residual correlation that has to be handled at the algorithmic level.

In the context of SNP data analysis, Nicodemus and Malley [57] point out that the Gini VIM systematically favors uncorrelated SNPs over strongly correlated SNPs even if all SNPs are non-informative. They consequently recommend the use of the permutation VIM. Nicodemus et al. [58] explore the behavior of the permutation VIM in the presence of correlated predictors in an extensive simulation study based on data generated from the logistic regression model. They conclude that predictors highly correlated with influential predictors but not having an own direct effect on the response are ranked higher than uncorrelated predictors and thus may be difficult to distinguish from truly influential predictors. This may either be seen as an advantage (if all these correlated predictors are potentially interesting) or as an inconvenience (if one is interested in the conditional effect of a particular predictor in a multivariate modeling perspective). Strobl et al. [67] take the second perspective and modify the permutation VIM such that the effect of a predictor is adjusted for other predictors through a computationally intensive conditional permutation procedure, while Meng et al. [51] take the opposite point of view and suggest to scale the VIM by the number of trees in which the corresponding predictor is used for splitting instead of scaling by the total number of trees. The latter procedure tends to increase the VI of highly correlated predictors that act as surrogates of each other and appear in the trees less often than if taken individually.

4.2 Testing variable importance

VIMs provide a ranking of predictors. However, in the standard form they say nothing about the significance of top-ranked predictors. VIMs always output a ranking - even if all predictors are useless to the prediction problem. Several attempts have been made in the literature to construct statistical tests for variable importance of similar nature as tests performed in the regression framework. Breiman and Cutler [7] suggest a straightforward testing approach based on a Z-score computed as the permutation VIM divided by $\hat{\sigma}/\sqrt{\texttt{ntree}}$, where $\hat{\sigma}$ stands for the standard deviation of the VIMs over the trees and `ntree` is the number of trees. However, Strobl and Zeileis [69] demonstrate in an extensive simulation study that the power of this straightforward test strongly depends on the `ntree` parameter and on the sample size, and that its power is zero for large sample sizes and small `ntree` – a very undesirable feature for a statistical test. A fundamental problem of this test is that its null-hypothesis is not clearly stated.

Complex permutation-based testing approaches are discussed by Wang et al. [77] and Altmann et al. [1]. In the latter paper, usual VIMs – no matter if biased or not – are calculated for each predictor using the original data set. The null distribution of the VIM is derived empirically by computing VIMs for a large number of data sets obtained by randomly permuting the response. The p-value is then computed as the fraction of permuted data sets yielding a more extreme VI. This method was originally developed to correct biased VIMs, but it can also be applied to any VIM for testing purposes or for variable selection. A similar permutation strategy is applied by Wang et al. [77] to an alternative VIM defined as the *maximal* conditional χ^2 statistic over all nodes of the forest that use the considered predictor. Note that in case of a very large number of predictors, e.g. in genome-wide association studies, permutation testing is computationally demanding and may require the use of parallel computing techniques.

4.3 Handling unbalanced data

Like many other machine learning algorithms the standard RF may perform poorly in case of extremely unbalanced class distributions. Here, the prediction accuracy for the majority gets a higher priority than the prediction accuracy for the minority class. In an extreme case where the minority class

makes up, e.g., only 5%, it could happen that the majority class is always predicted by the RF, yielding a prediction error of 5%. Unbalance of class distributions can be handled at different levels: at the data level and at the algorithmic level. At the data level over-sampling the minority class and/or down-sampling the majority class, respectively, have been considered in several papers to balance the class distribution. For example, Chen et al. [16] suggest a method denoted as 'Balanced RFs'. Each tree is built based on a data set combining a bootstrap sample from the minority class and a random sample from the majority class of the same size. At the algorithmic level a common approach for handling unbalanced data is cost-sensitive learning, where misclassification of the minority class is assigned a higher cost. Chen et al. [16] introduce a variant of the RF method based on this idea, the so called 'Weighted RF'. A weight is specified for each class and used for the computation of the Gini criterion and in the voting procedure.

4.4 Predictors involved in interactions

VIMs computed from RF turn out to identify SNPs involved in interactions (epistasis) as top-ranking with better accuracy than many other methods including logistic regression. This good performance is documented in several independent comparison studies implementing different simulation settings [26, 54, 73, 17]. In these studies, however, standard VIMs (either Gini or permutation) are used to rank the SNPs. The performance is thus essentially limited by the fact that a predictor must have at least a moderate main effect to be selected for splitting. Interacting predictors that both have no main effect thus have poor chance to receive a high VIM. A further drawback of RF in this context is that, although interaction effects are implicitly taken into account by RF, the standard VIM does not provide any information about the nature of potential interactions, i.e. whether predictors have an effect in combination with other predictors and if yes with which. The original Fortran code of RF implements a specific VIM for assessing *pairs* of variables, but the developers of the code state that caution is required when interpreting the results, and this VIM fails to identify true interactions in the wide simulation by Chen et al. [17].

A simple graphical method which might help to identify predictors involved in interactions consists in plotting the RF VIMs (which may also capture interaction effects) against a standard univariate statistic, see e.g.

[61]. Predictors having an effect on the response only in combination with other predictors are expected to be ranked higher by the RF VIM than with univariate statistics. Tang et al. [74] propose a specific VIM-based method for detecting gene-gene interactions which could easily be generalized to the detection of any interacting predictors. The procedure consists in computing VIMs of all SNPs i) based on the original data set and ii) after random permutation of some of the SNPs. A SNP that interacts with permuted SNPs is expected to have a lower VI after permutation, because permutation destroys both the main effect of the permuted SNPs and their interactions with other SNPs. In contrast, Bureau et al. [11] suggest to permute values of possibly interacting predictors together when calculating the permutation VIM. The resulting VIMs contain the combined effect and might be helpful for exploring interaction structures. Finally, some authors apply a two-stage approach [40, 50]. In the first step, a subset of potentially interesting predictors is extracted using RF. In the second step specific analyses are performed on this subset to identify interactions using so-called B statistics based on Bayesian factors [40] or Bayesian network analyses [50].

4.5 Random forests and variable selection

When used as a prediction method, the random forest algorithm is sometimes embedded into complex model selection approaches. Recursive variable selection methods constructing a random forest at each iteration have been proposed by Svetnik et al. [72] in the context of Quantitative Structure-Activity Relationship (QSAR) modeling and by Díaz-Uriarte, R. and De Andrès [24] for gene expression data analysis. At each iteration, the subset of considered predictors is updated by eliminating a certain fraction of predictors with the lowest VIM. The optimal subset is then the subset yielding the smallest error frequency [24] or the smallest area under the curve [14]. An alternative variable selection approach based on a nested collection of random forests is described in Genuer et al. [29]. Again, it needs to be emphasized that the resulting model with selected variables needs to be externally validated.

5 RF applications in bioinformatics: some examples

In this section we give a few examples of bioinformatics applications of RF. In most of these applications, the true relationship between response and predictors is complex and the predictors are strongly correlated, hence the attractiveness of RFs. Most studies do not apply one single method but several methods because each method has its own strengths and weaknesses and a combination of those will provide best insight into complex diseases [34].

A major field of application of RFs is genetic epidemiology, specifically large-scale genetic association studies. The response is typically a phenotype of interest, either categorical (e.g. diseased/healthy) or quantitative. The predictors are genetic markers, often SNPs that can be seen as predictors with two or three categories. RFs yield both a prediction tool and a ranking of the SNPs with respect to their classification ability. They have been considered in tens of bioinformatics papers [81, 62, 80, 2, 44] and biomedical applications [12, 46, 10, 56, 78, 15, 70, 47] including genome-wide studies [63, 79, 31, 64, 51]. In the application of RFs to genome-wide association data, the focus has been on different features of the algorithm. Whereas some used RFs to identify candidate regions similar to standard analyses [31], others focused on the detection of gene-gene interactions [46]. In a third group of applications, the resulting genetic regions are not of interest in themselves; instead, a prediction model is built using hundreds of SNPs at a time [19]. Although all of these approaches are very promising, validation of the results is still mostly lacking [41]. As a consequence, if regions were identified that had not been detected using standard approaches, this is yet difficult to interpret.

Other applications include prediction of patient outcome from high-dimensional gene expression data [24, 66, 5] or proteomic mass spectra classification [30, 52], where patients are instances and their outcome is the response to be predicted. Another class of applications deals with the prediction of molecule properties based on sequence information, e.g. the prediction of replication capacity based on HIV-1 sequence variation [65], prediction of C-to-U edited sites in plant mitochondrial DNA based on surrounding nucleotides [20], or the assessment of the relation between rifampin

resistance and amino acid sequence [21]. In these applications, instances are molecules and the response to be predicted is a property of interest. An early overview on the use of RFs in QSAR modeling is given in Svetnik et al. [72]. A further field where RFs have been successfully applied is ecology. Garzón et al. [27], Evans and Cushman [25], Cutler et al. [23] and Hernandez et al. [35] predict the presence of a species from climatic and topographic variables and Peters et al. [60] show that RF performs well in the prediction of vegetation types from environmental variables. Perdiguero-Alonso et al. [59] used RF to classify fish populations based on parasites as a marker for population assignment.

6 Conclusion

RF has become a major analysis tool in many fields of bioinformatics and will most probably remain relevant in the future due to its high flexibility, its in-built variable importance measures, and its attractive and understandable principle. RF has raised much enthusiasm in various fields of application and generated a vast amount of computational literature in the last ten years. However, RF approaches still have to face a number of challenges. They produce "odd unexpected results" in some specific cases, e.g. a bias depending on the type of the predictor. It is likely that further biases and problems will be discovered in the next years. The advantage of RF - absence of a specific underlying stochastic model - is also an inconvenience in the sense that i) it is difficult to understand what exactly happens in this deep jungle, and ii) RF does not fit in the statistical framework we are used to (including p-values, confidence intervals, etc). Both issues might be better understood in the future through consideration of the algorithm from a statistical point of view, possibly including the formulation of the method in terms of parameters and tests. Additional practical aspects could be addressed in future research such as the challenge of "reproducibility" – in a broad sense. RF involves several random components: the bootstrap samples/or subsamples on which each tree is built, and the random subset of candidate predictors considered at each split. Is it possible to reproduce exactly the same forest using another implementation? How stable are the results obtained in different runs? How sensitive is RF against small changes of the parameter values? How should we choose parameter values or, in case

of OOB-based tuning, how should we define the candidate parameter values? In a nutshell, RF most often yields very satisfying results, but how "random" are its results? These issues will have to be addressed for RF to be used beyond explorative studies.

References

[1] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer. Permutation importance: a corrected feature importance measure. Bioinformatics, 26(10):1340–1347, 2010.

[2] D. Amaratunga, J. Cabrera, and Y. S. Lee. Enriched random forests. Bioinformatics, 24(18):2010–2014, 2008.

[3] Sergey Bochkanov and Vladimir Bystritsky. ALGLIB - a cross-platform numerical analysis and data processing library. ALGLIB Project, 2011.

[4] A. L. Boulesteix, A. Bender, J. Lorenzo Bermejo, and C. Strobl. Random forest Gini importance favours SNPs with large minor allele frequency: assessment, sources and recommendations. Briefings in Bioinformatics, doi:10.1093/bib/bbr053, 2011.

[5] A. L. Boulesteix, C. Porzelius, and M. Daumer. Microarray-based classification and clinical predictors: On combined classifiers and additional predictive value. Bioinformatics, 24(15):1698–1706, 2008.

[6] L. Breiman. Random forests. Machine learning, 45(1):5–32, 2001.

[7] L. Breiman and A. Cutler. Random forests – classification manual. `http://www.math.usu.edu/\~adele/forests/cc_graphics.htm`, 2008.

[8] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees. Chapman & Hall, 1984.

[9] Leo Breiman and Adele Cutler. Random Forests - original implementation. `http://www.stat.berkeley.edu/~breiman/RandomForests/`, 2004.

[10] F. B. Briggs, B. A. Goldstein, J. L. McCauley, R. L. Zuvich, P. L. De Jager, J. D. Rioux, A. J. Ivinson, A. Compston, D. A. Hafler, S. L.

Hauser, et al. Variation within DNA repair pathway genes and risk of multiple sclerosis. American Journal of Epidemiology, 172(2):217–224, 2010.

[11] A. Bureau, J. Dupuis, K. Falls, K. L. Lunetta, B. Hayward, T. P. Keith, and P. Van Eerdewegh. Identifying SNPs predictive of phenotype using random forests. Genetic Epidemiology, 28(2):171–182, 2005.

[12] S. Cabras, M. E. Castellanos, G. Biino, I. Persico, A. Sassu, L. Casula, S. Del Giacco, F. Bertolino, M. Pirastu, and N. Pirastu. A strategy analysis for genetic association studies with known inbreeding. BMC Genetics, 12:63, 2011.

[13] M. L. Calle and V. Urrea. Letter to the editor: Stability of random forest importance measures. Briefings in Bioinformatics, 12(1):86–89, 2011.

[14] M. L. Calle, V. Urrea, A. L. Boulesteix, and N. Malats. AUC-RF: A new strategy for genomic profiling with random forest. Human Heredity, 72(2):121–132, 2011.

[15] J. S. Chang, R. F. Yeh, J. K. Wiencke, J. L. Wiemels, I. Smirnov, A. R. Pico, T. Tihan, J. Patoka, R. Miike, J. D. Sison, T. Rice, and M. R. Wrensch. Pathway analysis of single-nucleotide polymorphisms potentially associated with glioblastoma multiforme susceptibility using random forests. Cancer Epidemiology Biomarkers & Prevention, 17(6):1368–1373, 2008.

[16] C. Chen, A. Liaw, and L. Breiman. Using random forest to learn imbalanced data. Technical report, University of California, Berkeley, 2004. `http://www.stat.berkeley.edu/tech-reports/666.pdf`.

[17] C. C. Chen, H. Schwender, J. Keith, R. Nunkesser, K. Mengersen, and Macrossan P. Methods for identifying SNP interactions: A review on variations of logic regression, random forest and Bayesian logistic regression. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 8(6):1580–1591, 2011.

[18] X. Chen, M. Wang, and H. Zhang. The use of classification trees for bioinformatics. Data Mining and Knowledge Discovery, 1(1):55–63, 2011.

[19] E. Cosgun, N. A. Limdi, and C. W. Duarte. High-dimensional pharmacogenetic prediction of a continuous trait using machine learning techniques with application to warfarin dose prediction in african americans. Bioinformatics, 27(10):1384–1389, 2011.

[20] M. P. Cummings and D. S. Myers. Simple statistical models predict C-to-U edited sites in plant mitochondrial RNA. BMC Bioinformatics, 5:132, 2004.

[21] M. P. Cummings and M. R. Segal. Few amino acid positions in rpoB are associated with most of the rifampin resistance in Mycobacterium tuberculosis. BMC Bioinformatics, 5:137, 2004.

[22] T. Curk, J. Demsar, Q. Xu, G. Leban, U. Petrovic, I. Bratko, G. Shaulsky, and B. Zupan. Microarray data mining with visual programming. Bioinformatics, 21(3):396–398, 2005.

[23] D. R. Cutler, T. C. Edwards, K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler. Random forests for classification in ecology. Ecology, 88(11):2783–2792, 2007.

[24] R. Díaz-Uriarte and S. A. De Andrès. Gene selection and classification of microarray data using random forest. BMC Bioinformatics, 7:3, 2006.

[25] J. S. Evans and S. A. Cushman. Gradient modeling of conifer species using random forests. Landscape Ecology, 24(5):673–683, 2009.

[26] M. García-Magariños, I. López-de Ullibarri, R. Cao, and A. Salas. Evaluating the ability of tree-based methods and logistic regression for the detection of SNP-SNP interaction. Annals of Human Genetics, 73(3):360–369, 2009.

[27] M. B. Garzón, R. Blazek, M. Neteler, R. S. De Dios, H. S. Ollero, and C. Furlanello. Predicting habitat suitability with machine learning models: the potential area of Pinus sylvestris L. in the Iberian Peninsula. Ecological Modelling, 197(3-4):383–393, 2006.

[28] Mike Gashler. Waffles - a collection of command-line tools for researchers in machine learning, data mining, and related fields. Brigham Young University, 2011.

[29] R. Genuer, J. M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. Pattern Recognition Letters, 31(14):2225–2236, 2010.

[30] P. Geurts, M. Fillet, D. De Seny, M. A. Meuwis, M. Malaise, M. P. Merville, and L. Wehenkel. Proteomic mass spectra classification using decision tree based ensemble methods. Bioinformatics, 21(14):3138–3145, 2005.

[31] B. A. Goldstein, A. E. Hubbard, A. Cutler, and L. F. Barcellos. An application of random forests to a genome-wide association dataset: Methodological considerations & new findings. BMC Genetics, 11:49, 2010.

[32] B. A. Goldstein, E. C. Polley, and F. B. S. Briggs. Random forests for genetic association studies. Statistical Applications in Genetics and Molecular Biology, 10(1):32, 2011.

[33] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA Data Mining Software: An update. SIGKDD Explorations, 11(1):10–18, 2009.

[34] A. G. Heidema, J. M. A. Boer, N. Nagelkerke, E. C. M. Mariman, D. L. Van der A, and E. J. M. Feskens. The challenge for genetic epidemiologists: how to analyze large numbers of SNPs in relation to complex diseases. BMC Genetics, 7:23, 2006.

[35] P. A. Hernandez, I. Franke, S. K. Herzog, V. Pacheco, L. Paniagua, H. L. Quintana, A. Soto, J. J. Swenson, C. Tovar, T. H. Valqui, J. Vargas, and B. E. Young. Predicting species distributions in poorly-studied landscapes. Biodiversity and Conservation, 17(6):1353–1366, 2008.

[36] T. Hothorn, K. Hornik, and A. Zeileis. Unbiased recursive partitioning: A conditional inference framework. Journal of Computational and Graphical Statistics, 15(3):651–674, 2006.

[37] Torsten Hothorn, Peter Bühlmann, Sandrine Dudoit, Annette Molinaro, and Mark Van Der Laan. Survival ensembles. Biostatistics, 7(3):355–373, 2006.

[38] H. Ishwaran and U. B. Kogalur. Random survival forests for R. R News, 7(2):25–31, 2007.

[39] Abhishek Jaiantilal. randomforest-matlab: Random Forest (regression, classification and clustering) implementation for MATLAB (and Standalone). http://code.google.com/p/randomforest-matlab/, 2010.

[40] R. Jiang, W. Tang, X. Wu, and W. Fu. A random forest approach to the detection of epistatic interactions in case-control studies. BMC Bioinformatics, 10(Suppl 1):S65, 2009.

[41] I. R. König. Validation in genetic association studies. Briefings in Bioinformatics, 12(3):253–258, 2011.

[42] I. R. König, J. D. Malley, S. Pajevic, C. Weimar, H. C. Diener, and A. Ziegler. Patient-centered yes/no prognosis using learning machines. International Journal of Data Mining and Bioinformatics, 2(4):289–341, 2008.

[43] I. R. König, J. D. Malley, C. Weimar, H. C. Diener, and A. Ziegler. Practical experiences on the necessity of external validation. Statistics in Medicine, 26(30):5499–5511, 2007.

[44] S. S. F. Lee, L. Sun, R. Kustra, and S. B. Bull. EM-random forest and new measures of variable importance for multi-locus quantitative trait linkage analysis. Bioinformatics, 24(14):1603–1610, 2008.

[45] Andy Liaw and Matthew Wiener. Classification and regression by randomForest. R News, 2(3):18–22, 2002.

[46] C. Liu, H. H. Ackerman, and J. P. Carulli. A genome-wide screen of gene–gene interactions for rheumatoid arthritis susceptibility. Human Genetics, 129(5):473–485, 2011.

[47] K. L. Lunetta, L. B. Hayward, J. Segal, and P. Van Eerdewegh. Screening large-scale association study data: exploiting interactions using random forests. BMC Genetics, 5:32, 2004.

[48] J. D. Malley, K. G. Malley, and S. Pajevic. Statistical Learning for Biomedical Data. Cambridge University Press, 2011.

[49] O. L. Mangasarian and W. H. Wolberg. Cancer diagnosis via linear programming. SIAM News, 23(5):1 – 18, 1990.

[50] Y. Meng, Q. Yang, K. T. Cuenco, L. A. Cupples, A. L. DeStefano, and K. L. Lunetta. Two-stage approach for identifying single-nucleotide polymorphisms associated with rheumatoid arthritis using random forests and Bayesian networks. BMC Proceedings, 1(Suppl 1):S56, 2007.

[51] Y. A. Meng, Y. Yu, L. A. Cupples, L. A. Farrer, and K. L. Lunetta. Performance of random forest when SNPs are in linkage disequilibrium. BMC Bioinformatics, 10:78, 2009.

[52] B. H. Menze, B. M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, and F. A. Hamprecht. A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. BMC Bioinformatics, 10:213, 2009.

[53] Ananth Mohan, Zheng Chen, and Kilian Q. Weinberger. Web-search ranking with initialized gradient boosted regression trees. Journal of Machine Learning Research, Workshop and Conference Proceedings, 14:77–89, 2011.

[54] A. M. Molinaro, N. J. Carriero, R. Bjornson, P. Hartge, N. Rothman, and N. Chatterjee. Power of data mining methods to detect genetic associations and interactions. Human Heredity, 72(2):85–97, 2011.

[55] K. K. Nicodemus. Letter to the editor: On the stability and ranking of predictors from random forest variable importance measures. Briefings in Bioinformatics, 12(4):369–373, 2011.

[56] K. K. Nicodemus, J. H. Callicott, R. G. Higier, A. Luna, D. C. Nixon, B. K. Lipska, R. Vakkalanka, I. Giegling, D. Rujescu, D. S. Clair, P. Muhlia, Y. Y. Shugart, and D. R. Weinberger. Evidence of statistical epistasis between DISC1, CIT and NDEL1 impacting risk for schizophrenia: biological validation with functional neuroimaging. Human Genetics, 127(4):441–452, 2010.

[57] K. K. Nicodemus and J. D. Malley. Predictor correlation impacts machine learning algorithms: implications for genomic studies. Bioinformatics, 25(15):1884–1890, 2009.

[58] K. K. Nicodemus, J. D. Malley, C. Strobl, and A. Ziegler. The behaviour of random forest permutation-based variable importance measures under predictor correlation. BMC Bioinformatics, 11:110, 2010.

[59] D. Perdiguero-Alonso, F. E. Montero, A. Kostadinova, J. A. Raga, and J. Barrett. Random forests, a novel approach for discrimination of fish populations using parasites as biological tags. International Journal for Parasitology, 38(12):1425–1434, 2008.

[60] J. Peters, B. De Baets, N. E. C. Verhoest, R. Samson, S. Degroeve, P. De Becker, and W. Huybrechts. Random forests as a tool for ecohydrological distribution modelling. Ecological Modelling, 207(2-4):304–318, 2007.

[61] W. Rodenburg, A. G. Heidema, J. M. A. Boer, I. M. J. Bovee-Oudenhoven, E. J. M. Feskens, E. C. M. Mariman, and J. Keijer. A framework to identify physiological responses in microarray-based gene expression studies: selection and interpretation of biologically relevant genes. Physiological Genomics, 33(1):78–90, 2008.

[62] A. S. Rodin, A. Litvinenko, K. Klos, A. C. Morrison, T. Woodage, J. Coresh, and E. Boerwinkle. Use of wrapper algorithms coupled with a random forests classifier for variable selection in large-scale genomic association studies. Journal of Computational Biology, 16(12):1705–1718, 2009.

[63] U. Roshan, S. Chikkagoudar, Z. Wei, K. Wang, and H. Hakonarson. Ranking causal variants and associated regions in genome-wide association studies by the support vector machine and random forest. Nucleic Acids Research, 39(9):e62, 2011.

[64] Daniel F. Schwarz, Inke R. König, and Andreas Ziegler. On safari to random jungle: a fast implementation of random forests for high-dimensional data. Bioinformatics, 26(14):1752–1758, 2010.

[65] M. R. Segal, J. D. Barbour, and R. M. Grant. Relating HIV-1 sequence variation to replication capacity via trees and forests. Statistical Applications in Genetics and Molecular Biology, 3(1):2, 2004.

[66] A. Statnikov, L. Wang, and C. F. Aliferis. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinformatics, 9:319, 2008.

[67] C. Strobl, A. L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. BMC Bioinformatics, 9:307, 2008.

[68] C. Strobl, A. L. Boulesteix, A. Zeileis, and T. Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics, 8:25, 2007.

[69] C. Strobl and A. Zeileis. Danger: High power! - exploring the statistical properties of a test for random forest variable importance. In COMPSTAT 2008 - Proceedings in Computational Statistics, volume II, pages 59–66, Heidelberg, Germany, 2008. Physica-Verlag.

[70] Y. V. Sun, Z. Cai, K. Desai, R. Lawrance, R. Leff, A. Jawaid, S. L. R. Kardia, and H. Yang. Classification of rheumatoid arthritis status with candidate gene and genome-wide single-nucleotide polymorphisms using random forests. BMC Proceedings, 1(Suppl 1):S62, 2007.

[71] Fran Supek. FastRandomForest – an efficient, multithreaded implementation of the Random Forest classifier for Java. integrates into Weka. Centre for Informatics and Computing of Ruder Boskovic Institute. `http://code.google.com/p/fast-random-forest/`, 2009.

[72] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston. Random forest: a classification and regression tool for compound classification and QSAR modeling. Journal of Chemical Information and Computer Sciences, 43(6):1947–1958, 2003.

[73] S. Szymczak, J. M. Biernacka, O. Cordell, H. J.and González-Recio, I. R. König, H. Zhang, and Y. V. Sun. Machine learning in genome-wide association studies. Genetic Epidemiology, 33:S51–S57, 2009.

[74] R. Tang, J. P. Sinnwell, J. Li, D. N. Rider, M. De Andrade, and J. M. Biernacka. Identification of genes and haplotypes that predict rheumatoid arthritis using random forests. BMC Proceedings, 3(Suppl 7):S68, 2009.

[75] Goran Topić and Tomislav Šmuc. PARF - Parallel Random Forest Algorithm. Centre for Informatics and Computing of Ruder Boskovic Institute. http://www.irb.hr/en/research/projects/it/2004/2004-111/, 2011.

[76] A. Verikas, A. Gelzinis, and M. Bacauskiene. Mining data with random forests: A survey and results of new tests. Pattern Recognition, 44(2):330–349, 2011.

[77] M. Wang, X. Chen, and H. Zhang. Maximal conditional chi-square importance in random forests. Bioinformatics, 26(6):831–837, 2010.

[78] M. Wang, X. Chen, M. Zhang, W. Zhu, K. Cho, and H. Zhang. Detecting significant single-nucleotide polymorphisms in a rheumatoid arthritis study using random forests. BMC Proceedings, 3(Suppl 7):S69, 2009.

[79] M. Xu, K. G. Tantisira, A. Wu, A. A. Litonjua, J. Chu, B. E. Himes, A. Damask, and S. T. Weiss. Genome wide association study to predict severe asthma exacerbations in children using random forests classifiers. BMC Medical Genetics, 12:90, 2011.

[80] W. W. Yang and C. C. Gu. Selection of important variables by statistical learning in genome-wide association analysis. BMC Proceedings, 3(Suppl 7):S70, 2009.

[81] W. Zhang, Y. Xiong, M. Zhao, H. Zou, X. Ye, and J. Liu. Prediction of conformational B-cell epitopes from 3D structures by random forest with a distance-based feature. BMC Bioinformatics, 12:341, 2011.