Overview

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] and Inke R. König[2]

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 return measures of variable importance. This paper synthesizes 10 years of RF development with emphasis on applications to bioinformatics and computational biology. Special attention is paid 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.

How to cite this article:
WIREs Data Mining Knowl Discov 2012, 2: 493–507 doi: 10.1002/widm.1072

INTRODUCTION

Random forest (RF) methodology is used to address two main classes of problems: to construct a prediction rule in 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 RF 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.[1] 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 nonlinear 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.

Several reviews on RF have been published in the last few years, each focussing on a particular aspect of the methodology or a particular field of application. For example, Malley et al.[2] depict the theory of RF in a broad context, Goldstein et al.[3] describe in detail the RF algorithm and its applications to genetic epidemiology, Chen et al.[4] give an extensive overview of applications of recursive partitioning to

*Correspondence to: boulesteix@ibe.med.uni-muenchen.de

[1] Institut für Medizinische Informationsverarbeitung, Biometrie und Epidemiologie, Ludwig-Maximilians-Universität München, München, Germany

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

DOI: 10.1002/widm.1072

bioinformatics, and Verikas et al.[5] survey RF applications and their performance in comparison with other methods in a more general context.

The present paper reviews practical aspects of RF methodology with examples borrowed from computational biology/medicine and bioinformatics. Essential shortcomings and biases of RF and its VIMs are discussed as well as alternative approaches to circumvent these problems are presented. After a short introductory overview of the main RF variants that summarizes previously published methodological reviews [e.g., Ref 3], we address the selection of important parameters in the RF algorithm and give advice to avoid common pitfalls and biases. The next section reviews existing software and discusses implementation issues to provide helpful guidelines for applications. Finally, the last section addresses special challenges that are particularly relevant to bioinformatics applications and provides practical advice to cope with them when considering state-of-the-art methods from the literature.

RANDOM FOREST VARIANTS AND PARAMETERS

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, and (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, 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 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 problem is described in more detail in the following sections, including a discussion of the notion of bias. The alternative class of decision trees developed by Hothorn et al.[7] and Strobl et al.[8] 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.

Gini Importance versus Permutation Importance

The standard RF computes two different VIMs for each predictor: the Gini VIM and the permutation VIM (see Ref 3 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

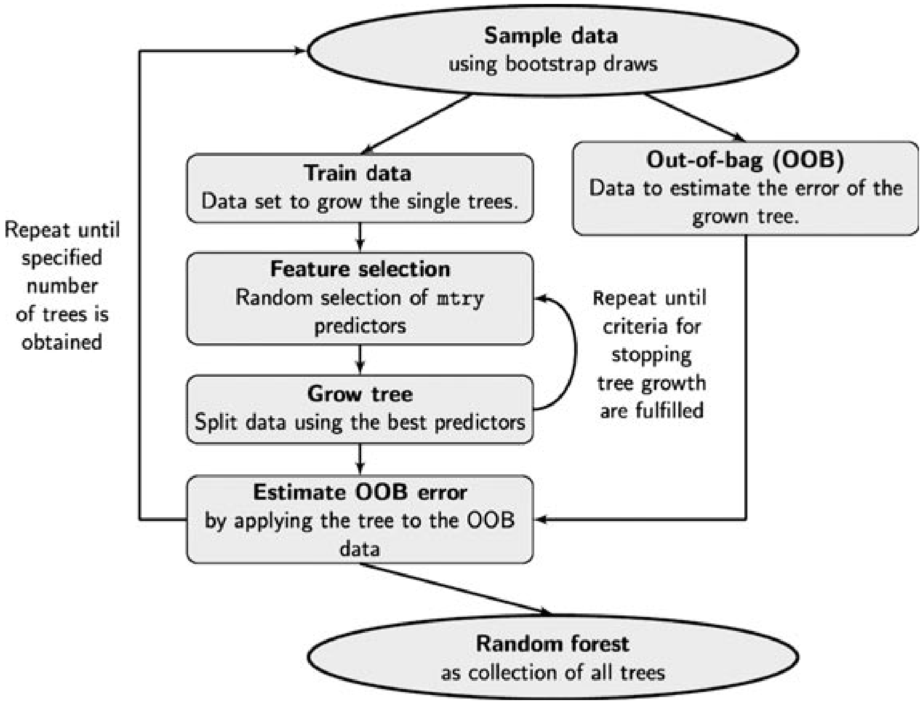
Sample data
using bootstrap draws
Train data
Data set to grow the single trees.
Out-of-bag (OOB)
Data to estimate the error of the grown tree.
Repeat until specified number of trees is obtained
Feature selection
Random selection of mtry predictors
Repeat until criteria for stopping tree growth are fulfilled
Grow tree
Split data using the best predictors
Estimate OOB error
by applying the tree to the OOB data
Random forest
as collection of all trees

FIGURE 1 | Random forest algorithm.

'important' predictor is expected to increase the OOB error, leading to a high permutation VIM.

Although 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, which does not use the decrease of Gini impurity as a splitting criterion, computes only the permutation VIM. If all predictors are noninformative 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 noninformative 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

Perhaps the 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, whereas 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.[8] This so-called 'selection bias' transfers directly into a 'Gini VIM bias' because 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 as 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 noninformative 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.[9,10] In genetic epidemiology, noninformative single nucleotide polymorphisms (SNPs) with large minor allele frequency (MAF) are systematically favored by the Gini VIM over noninformative SNPs with small MAF, potentially yielding misleading rankings of the candidate SNPs. The use of the permutation VIM, which 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.[11] If all predictors are noninformative, 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.[11]

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 Ref 3, and references therein). This may be related to the higher instability of the permutation VIM.[12]

PRACTICAL GUIDANCE FOR THE CHOICE OF PARAMETERS

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

Number of Trees

The number of trees in the forest should 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 the number 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.[3] 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[13] 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 noninformative, yielding inaccurate trees. More generally, small number of candidate predictors possibly lead to the selection of suboptimal predictors and thus to information loss. 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. Thus a small number of candidate predictors may also lead to a better utilization of available information.

Size of 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 candidate predictors 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, e.g., the minimal size that a node should have to be split, the maximal number of layers, or a threshold value for the splitting criterion.

Size of Leaves

Although they are also related to tree size, the parameters controlling the minimal size of the leaves 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 leaf. Even if it is selected as the best predictor according to the splitting criterion, such a predictor would be excluded because it yields a leaf smaller than the prespecified size. For example, suppose that we set the minimal size of leaves to 10 (a quite large value). If a given binary predictor has two classes of size 91 and 9, respectively, it cannot be selected for splitting even if it predicts the response class perfectly. Our advice is thus 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.

Resampling Scheme

An 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.[8] 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. We recommend to systematically use sampling without replacement to avoid the bias outlined by Strobl et al.,[8] i.e., to avoid that noninformative predictors with many categories are selected more often than informative predictors with few categories. The size of the subsamples is then an additional parameter, which can, e.g., be set to 0.632 in analogy with the average proportion of observations included in a bootstrap sample drawn with replacement.[8]

Handling Unbalanced Data

Like many other machine learning algorithms the standard RF classifier may perform poorly in case of extremely unbalanced classes. Oversampling the minority class and/or downsampling the majority class, respectively, have been suggested to balance the class distribution. These sampling strategies can be specified by the user through wrapper functions if not supported by the software. The R package *randomForest*, e.g., provides the option `sampsize` that enables the user to specify how many observations to draw from each class. An alternative solution consists in changing the cutoff value for the majority vote. In RF, an observation is allocated to a class if its average vote in favor of this class exceeds a predefined threshold. By default, for each class, the corresponding threshold takes the value $\frac{1}{k}$ with k denoting the number of classes. Assigning a lower threshold to a class with less observations puts more weight on this class and allows the allocation to that class even if only few trees vote for it. Furthermore, we recommend to grow the trees with maximal depth to give the minority class a better chance to dominate the majority class in a leaf. For more information on the use of RF for unbalanced data, we refer to Refs 14, 15, and 16.

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.[3]. 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.[17]

IMPLEMENTATIONS AND EXAMPLE CODE

Implementations

A brief overview of available RF implementations is given in Table 1, whereas 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*.[18] 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 of important arguments for the R functions `randomForest` and `cforest` is shown in Tables 3 and 4.

Recommendations

The R packages *randomForest* and *party* present a number of advantages. First, the code is freely available. It is thus easier to understand exactly what has been implemented and to adapt the code to the specific needs of the considered application or to develop further variants of RFs. Note that R is a free open-source software that can be easily installed on any

TABLE 1 Overview of Random Forest Implementations

Name	RF Only[a]	MT[b]	System	Code
ALGLIB[20]	No	No	Win/Unix	C++
cforest Function in R package party[21]	No	Yes	All	C++/S
FastRandomForest[22]	Yes	Yes	All[c]	Java
Orange[23]	No	No	Win/Unix/Mac	C++/Python
PARF—parallel RF algorithm[25]	Yes	Yes	All[c]	F90
Random forest[25]	Yes	No	All[c]	F77
Randomforest–matlab[26]	Yes	No	All[c]	C/C++/
randomForest–R package[27]	No	Yes	All	C++/S
Random Jungle[28]	Yes	Yes[d]	Win/Unix	C++
RT-Rank[29]	Yes	Yes	Unix[c]	C++/Python
Waffles[30]	No	No	Win/Unix/Mac	C++
Weka 3[31]	No	No	All[c]	Java

[a] RF only indicates whether this is a program only for RF analysis (yes) or part of a broader software package (no).
[b] Multithreading ability.
[c] Provided that a compiler is available.
[d] Available only for UNIX machines.

PC. Second, R implements an extremely wide range of statistical methods. It is thus very convenient to run a complete analysis involving RFs among other methods. Third, the documentation of the R implementations of RF are on average of higher quality than in other software packages. Fourth, the R package *randomForest* has powerful and easy to handle graphical facilities to illustrate results, including, e.g., plots of the margins of the observations, multidimensional scaling plots of proximity matrices, variable importance plots, and partial dependence plots depicting the marginal effect of a variable on the response. Last but not least, R is convenient to produce 'reproducible analyses' (i.e., analyses that can be rerun by an independent user just by copy-pasting the code) or to automatically integrate results into a Latex document through the tool Sweave.[19]

The Random Jungle (RJ) implementation of RFs has been especially designed for genome-wide association studies and can be easily accessed through R (see *Example Code 2: Random Jungle*). It can deal with large data sets that are intractable in R and implements both the RF variant and the CIF variant of RFs among many others. Through its multithreading capability and a special sparse memory use RJ addresses the computational limitations of R implementations of RF. For more technical details and, in particular, the comparison of performance of different implementations (RJ, RF, and the *randomForest* R package) on a real genome-wide association study, we refer to Ref 28.

Altogether, we believe that the R packages *randomForest* and *party* (for data sets of moderate size, especially as far as *party* is concerned) and the Random Jungle software cover the needs of bioinformatics applications reasonably well and we consequently recommend their use. Exemplary R codes are displayed below for both the R package *randomForest* and the Random Jungle software.

Example Code

The following RF example consists of two parts: an example code using the R package *randomForest* and an example code using the RJ implementation.[28] 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'[32] from the UCI repository (`http://archive.ics.uci.edu/ml/`) is used for illustrative purposes. It includes $n = 699$ observations and only nine predictors. The response variable (`class`) is binary (benign versus malignant).

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.

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	Reimplementation 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.
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 offers 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 such as 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).
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, and so on.	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.

TABLE 3 | Important Arguments to the Function `randomForest` from the R Package *randomForest*

Parameter	Acronym	Default (Classification resp. Regression)
Number of trees	`ntree`	500
Number of candidate predictors	`mtry`	$\sqrt{p}$ resp. $p/3$
Maximum number of terminal nodes	`maxnodes`	Not restricted
Minimum size of terminal nodes	`nodesize`	1 resp. 5
Resampling scheme	`replace`	`TRUE`

```
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˜., importance = TRUE,
                              data=cancerDf, mtry=3, ntree=500)
## get importance measurements
impClass <- as.data.frame(classRFCancer$importance)
```

Example Code 2: Random Jungle

The RJ example is called from the R environment to provide better data handling. A compiled version of RJ can be downloaded from the project page `http://randomjungle.de` for several operating systems. The help pages of RJ 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
                    "--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
```

CHALLENGES IN BIOINFORMATICS APPLICATIONS

The Use of RF in Bioinformatics: a Brief Overview

RF methodology has been widely used in computational biology and medicine. In most of these applications, the true relationship between response and predictors is complex and the predictors are strongly

TABLE 4 | Important Arguments to the Function `cforest` from the R Package *party*

Parameter	Acronym	Default
Number of trees	`ntree`	500
Number of candidate predictors	`mtry`	5
P value threshold	`mincriterion`	0.95
Minimum size of node to be split	`minsplit`	20
Maximal number of layers	`maxdepth`	Not restricted
Minimum size of leaves	`minbucket`	7
Resampling scheme	`replace`	`TRUE`

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.[33]

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[34–38] and biomedical applications[39–46] including genome-wide studies.[28,47–50] 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,[49] others focused on the detection of gene–gene interactions.[40] 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.[51] Although all of these approaches are very promising, validation of the results is still mostly lacking.[52] As a consequence, if regions were identified that had not been detected using standard approaches, this is yet more difficult to interpret than in the case of statistical tests yielding *P* values. Validation of the results using independent data may thus be even more crucial when RF is used than if the results are obtained based on standard approaches.

Other applications include prediction of patient outcome from high-dimensional gene expression data[53–55] or proteomic mass spectra classification,[56,57] 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,[58] prediction of C-to-U edited sites in plant mitochondrial DNA based on surrounding nucleotides,[59] or the assessment of the relation between rifampin resistance and amino acid sequence.[60] 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 quantitative structure–activity relationship modeling is given in Ref 61. A further field where RFs have been successfully applied is ecology. Garzón et al., Evans et al., Cutler et al. and Hernandez et al. predict the presence of a species from climatic and topographic variables and Peters et al. show that RF performs well in the prediction of vegetation types from environmental variables. Perdiguero-Alonso et al.[67] used RF to classify fish populations based on parasites as a marker for population assignment.

Strong correlations between the predictors, the need to assess the predictors' importances in form of *P* values (as usual in biomedicine), the presence of interactions between predictors, and the importance of variable selection in high-dimensional settings characterize applications of RFs in the context of bioinformatics and computational biology/medicine, although these problems are also relevant to many other fields of applications. They are discussed in the following section, including practical advice regarding the choice of the method and software.

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. For example, SNP data may be correlated due to linkage disequilibrium (LD), and transcriptomic or metabolomic data may show high correlation within functional groups. Although 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[11] point out that the Gini VIM

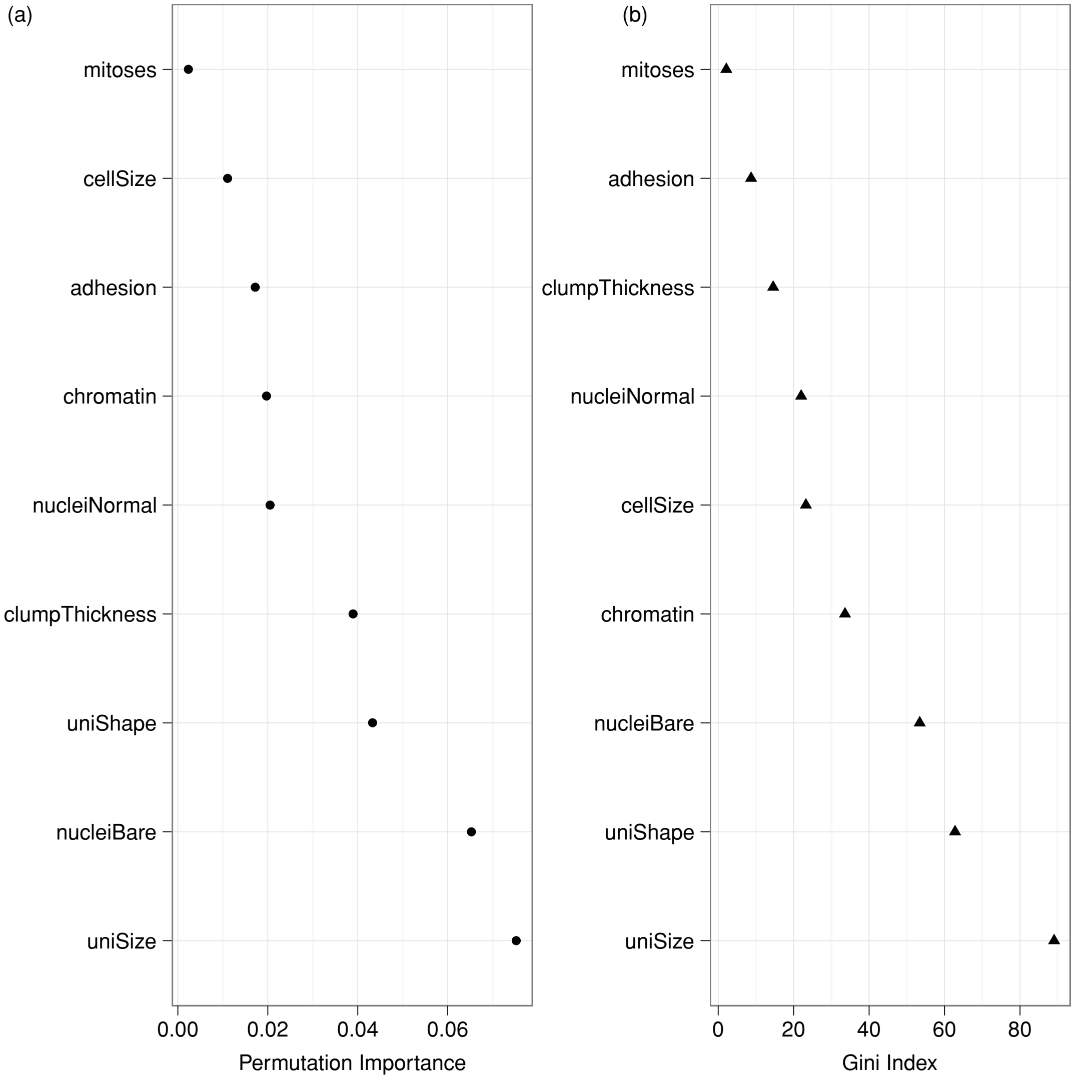

FIGURE 2 | Two different types of variable importance of the example data set. (a) permutation VIM. (b) Gini VIM. The two VIMs result in a different ordering.

systematically favors uncorrelated SNPs over strongly correlated SNPs even if all SNPs are noninformative. They consequently recommend the use of the permutation VIM. Nicodemus et al.[68] 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 where only some of the correlated predictors have an effect on the response. With such simulated data, we know which predictors are causal. Nicodemus et al.[68] conclude that predictors highly correlated with causal predictors but not having their own direct effect on the response are ranked higher than uncorrelated predictors and thus may be difficult to distinguish from causal 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.[69] 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, whereas Meng et al.[50] 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. Both procedures are implemented in the Random Jungle software. Finally, note that, like most data analysis tools, RF is based on the notion of correlation and cannot prove causality relationships.

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. This is often considered as an important pitfall in the context of clinical bioinformatics because in medical literature *P* values and confidence intervals are usually offered much attention. 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[70] 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[71] 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. In our opinion the definition of a statistically correct testing framework is an important challenge that could be addressed in future research.

Complex permutation-based testing approaches are discussed by Wang et al.[72] and Altmann et al.[73] 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.[72] 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.

Predictors Involved in Interactions

Another important methodological challenge that is particularly relevant to the field of computational biology/medicine is the identification of predictors involved in interactions and their efficient use in the prediction rule. 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.[74–77] 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.[77]

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., Rodenburg et al.). 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.[79] 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 (1) based on the original data set and (2) 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.[80] 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. Sakoparnig et al.[81] define importance measures for pairs of predictors based on the frequency of common occurrences in branches, whereas Yoshida and Koike[82] propose an alternative tree algorithm that jointly selects several predictors to better take interactions into account. Finally, some authors apply a two-stage approach.[83,84] 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[83] or Bayesian network analyses.[84]

Random Forests and Variable Selection

In bioinformatics applications, the predictor space is often highly dimensional, gene expression data analysis being a prominent example. In this case, it might be useful to incorporate a variable selection procedure into the RF algorithm to better separate noise from informative predictors.

When used as a prediction method, the RF algorithm is thus sometimes embedded into complex model selection approaches. Recursive variable selection methods constructing a RF at each iteration have been proposed by Svetnik et al. in the context of quantitative structure–activity relationship modeling and by Díaz-Uriarte and De Andrès 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[53] or the smallest area under the curve.[85] An alternative variable selection approach based on a nested collection of RFs is described in Ref 86. Again, it needs to be emphasized that the resulting model with selected variables needs to be externally validated.

CONCLUSION

RF has become a popular analysis tool in many application fields including bioinformatics and will most probably remain relevant in the future due to its high flexibility, its in-built VIMs, and its attractive and understandable principle. However, RF approaches still have to face a number of challenges. They produce ‘unexpected results’ in some specific cases, e.g., a bias depending on the distribution of the predictor. For the analysis of data including some categorical predictors, we recommend the use of CIF (as implemented, e.g., in the package *party*) instead of the standard RF algorithm. If RF is used, the permutation VIM should be preferred to the Gini importance measure. As far as software is concerned, sensible choices are the R implementations of RF and CIF (available in the packages *randomForest* and *party*, respectively) as well as the Random Jungle software developed for genome-wide SNP studies.

It is likely that further biases and problems will be discovered in the next years, especially in the context of bioinformatics applications, where the data often have particular complex structures. The model-free character of RF does not imply that the methodology can always be applied blindly to any type of data without caution.

An additional aspect that is currently gaining importance in the bioinformatics community and could be addressed in future research is the challenge of ‘reproducibility’ in a broad sense[87] and stability of research results. 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 and are particularly relevant to the field of high-dimensional omics analysis that is characterized by high instability.

REFERENCES

1. König IR, Malley JD, Pajevic S, Weimar C, Diener HC, Ziegler A. Patient-centered yes/no prognosis using learning machines. *Int J Data Min Bioinf* 2008, 2:289–341.
2. Malley JD, Malley KG, Pajevic S. *Statistical Learning for Biomedical Data* 2011. New York, USA: Cambridge University Press.
3. Goldstein BA, Polley EC, Briggs FBS. Random forests for genetic association studies. *Stat Appl Genet Mol Biol* 2011, 10:32.
4. Chen X, Wang M, Zhang H. (2011b). The use of classification trees for bioinformatics. *Data Min Knowl Discov*, 1:55–63.

5. Verikas A, Gelzinis A, Bacauskiene M. Mining data with random forests: a survey and results of new tests. *Pattern Recognit* 2011, 44:330–349.
6. Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and Regression Trees*. Wadsworth, Belmont, USA: Chapman & Hall; 1984.
7. Hothorn T, Hornik K, Zeileis A. (2006b). Unbiased recursive partitioning: a conditional inference framework. *J Comput Graph Stat*, 15:651–674.
8. 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.
9. Nicodemus KK. Letter to the editor: on the stability and ranking of predictors from random forest variable importance measures. *Brief Bioinf* 2011, 12:369–373.
10. Boulesteix AL, Bender A, Bermejo JL, Strobl C. Random forest Gini importance favours SNPs with large minor allele frequency: assessment, sources and recommendations. *Brief Bioinf* 2012, 13:292–304.
11. Nicodemus KK, Malley JD. Predictor correlation impacts machine learning algorithms: implications for genomic studies. *Bioinformatics* 2009, 25:1884–1890.
12. Calle ML, Urrea V. Letter to the editor: stability of random forest importance measures. *Brief Bioinf* 2011, 12:86–89.
13. Breiman L. Random forests. *Mach Learn* 2001, 45:5–32.
14. Blagus R, Lusa L. Class prediction for high-dimensional class-imbalanced data. *BMC Bioinformatics* 2010 2010, 11:523.
15. Lin WJ, Chen JJ. Class-imbalanced classifiers for high-dimensional data. *Briefings in Bioinformatics* 2012. [Epub ahead of print].
16. Chen C, Liaw A, Breiman L. Using random forest to learn imbalanced data. Technical report, # 666, Department of Statistics, University of California, Berkeley, CA; 2004. Available at: http://www.stat.berkeley.edu/tech-reports/666.pdf. (Accessed 15 August 2011).
17. König IR, Malley JD, Weimar C, Diener HC, Ziegler A. Practical experiences on the necessity of external validation. *Stat Med* 2007, 26:5499–5511.
18. Ishwaran H, Kogalur UB. Random survival forests for R. *R News* 2007, 7:25–31.
19. Leisch F. Sweave: dynamic generation of statistical reports using literate data analysis. In: Härdle W, Rönz B, eds, *Compstat 2002-Proceedings in Computational Statistics* 2002. Heidelberg, Germany: Physica Verlag; 2002, 575–580.
20. Bochkanov S, Bystritsky V. ALGLIB - a cross-platform numerical analysis and data processing library. ALGLIB Project. Novgorod, Russia; 2011.
21. Hothorn T, Bühlmann P, Dudoit S, Molinaro A, Van Der Laan, M. (2006a). Survival ensembles. *Biostatistics*, 7:355–373.
22. Supek F. FastRandomForest—an efficient, multi-threaded implementation of the Random Forest classifier for Java integrates into Weka. Centre for Informatics and Computing of Ruder Boskovic Institute, Zagreb, Croatia; 2009. Available at: http://code.google.com/p/fast-random-forest/. (Accessed 15 August 2011).
23. Curk T, Demsar J, Xu Q, Leban G, Petrovic U, Bratko I, Shaulsky G, Zupan B. Microarray data mining with visual programming. *Bioinformatics* 2005, 21:396–398.
24. Topić, G, Šmuc T. PARF–Parallel Random Forest Algorithm. Centre for Informatics and Computing of Ruder Boskovic Institute, Zagreb, Croatia; 2011. Available at: http://www.irb.hr/en/research/projects/it/2004/2004-111/. (Accessed 15 August 2011).
25. Breiman L, Cutler A. Random forests—original implementation, Berkeley, CA; 2004. Available at: http://www.stat.berkeley.edu/ breiman/RandomForests/. (Accessed 15 August 2011).
26. Jaiantilal A. randomforest-matlab: Random Forest (regression, classification and clustering) implementation for MATLAB (and Standalone), Boulder; 2010. Available at: http://code.google.com/p/randomforest-matlab/. (Accessed 15 August 2011).
27. Liaw A, Wiener M. Classification and regression by randomForest. *R News* 2002, 2:18–22.
28. 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.
29. Mohan A, Chen Z, Weinberger KQ. Web-search ranking with initialized gradient boosted regression trees. *J Mach Learn Res Workshop Conf Proc* 2011, 14:77–89.
30. Gashler M. Waffles—a collection of command-line tools for researchers in machine learning, data mining, and related fields. Brigham Young University.
31. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH.2009 The WEKA Data Mining Software: an update. *SIGKDD Explor* 2011, 11:10–18.
32. Mangasarian OL, Wolberg WH. Cancer diagnosis via linear programming. *SIAM News* 1990, 23:1–18.
33. Heidema AG, Boer JMA, Nagelkerke N, Mariman ECM, Van der A DL, Feskens EJ. M. The challenge for genetic epidemiologists: how to analyze large numbers of SNPs in relation to complex diseases. *BMC Genet* 2006, 7:23.
34. Zhang W, Xiong Y, Zhao M, Zou H, Ye X, Liu J. Prediction of conformational B-cell epitopes from 3D structures by random forest with a distance-based feature. *BMC Bioinformatics* 2011, 12:341.

35. 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.
36. Yang WW, Gu CC. Selection of important variables by statistical learning in genome-wide association analysis. *BMC Proc* 2009, 3(suppl 7):S70.
37. Amaratunga D, Cabrera J, Lee YS. Enriched random forests. *Bioinformatics* 2008, 24:2010–2014.
38. Lee SSF, Sun L, Kustra R, Bull SB. EM-random forest and new measures of variable importance for multilocus quantitative trait linkage analysis. *Bioinformatics* 2008, 24:1603–1610.
39. Cabras S, Castellanos ME, Biino G, Persico I, Sassu A, Casula L, Del Giacco, S, Bertolino F, Pirastu M, Pirastu N. A strategy analysis for genetic association studies with known inbreeding. *BMC Genet* 2011, 12:63.
40. Liu C, Ackerman HH, Carulli JP. A genome-wide screen of gene–gene interactions for rheumatoid arthritis susceptibility. *Hum Genet* 2011, 129:473–485.
41. Briggs FB, Goldstein BA, McCauley JL, Zuvich RL, De Jager PL, Rioux JD, Ivinson AJ, Compston A, Hafler DA, Hauser SL, et al. Variation within DNA repair pathway genes and risk of multiple sclerosis. *Am J Epidemiol* 2010, 172:217–224.
42. Nicodemus KK, Callicott JH, Higier RG, Luna A, Nixon DC, Lipska BK, Vakkalanka R, Giegling I, Rujescu D, Clair DS, Muhlia P, Shugart YY, Weinberger DR. (2010a). Evidence of statistical epistasis between DISC1, CIT and NDEL1 impacting risk for schizophrenia: biological validation with functional neuroimaging. *Hum Genet*, 127:441–452.
43. Wang M, Chen X, Zhang M, Zhu W, Cho K, Zhang H. Detecting significant single-nucleotide polymorphisms in a rheumatoid arthritis study using random forests. *BMC Proc* 2009, 3(suppl 7):S69.
44. Chang JS, Yeh RF, Wiencke JK, Wiemels JL, Smirnov I, Pico AR, Tihan T, Patoka J, Miike R, Sison JD, Rice T, Wrensch MR. Pathway analysis of single-nucleotide polymorphisms potentially associated with glioblastoma multiforme susceptibility using random forests. *Cancer Epidemiol Biomarkers Prev* 2008, 17:1368–1373.
45. Sun YV, Cai Z, Desai K, Lawrance R, Leff R, Jawaid A, Kardia SLR, Yang H. Classification of rheumatoid arthritis status with candidate gene and genome-wide single-nucleotide polymorphisms using random forests. *BMC Proc* 2007, 1(suppl 1):S62.
46. Lunetta KL, Hayward LB, Segal J, Van Eerdewegh, P. Screening large-scale association study data: exploiting interactions using random forests. *BMC Genet* 2004, 5:32.
47. Roshan U, Chikkagoudar S, Wei Z, Wang K, Hakonarson H. Ranking causal variants and associated regions in genome-wide association studies by the support vector machine and random forest. *Nucleic Acids Res* 2011, 39:e62.
48. Xu M, Tantisira KG, Wu A, Litonjua AA, Chu J, Himes BE, Damask A, Weiss ST. Genome wide association study to predict severe asthma exacerbations in children using random forests classifiers. *BMC Med Genet* 2011, 12:90.
49. Goldstein BA, Hubbard AE, Cutler A, Barcellos LF. An application of random forests to a genome-wide association dataset: methodological considerations and new findings. *BMC Genet* 2010, 11:49.
50. 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.
51. Cosgun E, Limdi NA, Duarte CW. High-dimensional pharmacogenetic prediction of a continuous trait using machine learning techniques with application to warfarin dose prediction in african americans. *Bioinformatics* 2011, 27:1384–1389.
52. König IR. Validation in genetic association studies. *Brief Bioinf* 2011, 12:253–258.
53. Díaz-Uriarte R, De Andrès SA. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics* 2006, 7:3.
54. Statnikov A, Wang L, Aliferis CF. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics* 2008, 9:319.
55. Boulesteix AL, Porzelius C, Daumer M. Microarray-based classification and clinical predictors: on combined classifiers and additional predictive value. *Bioinformatics* 2008, 24:1698–1706.
56. Geurts P, Fillet M, De Seny D, Meuwis MA, Malaise M, Merville MP, Wehenkel L. Proteomic mass spectra classification using decision tree based ensemble methods. *Bioinformatics* 2005, 21:3138–3145.
57. Menze BH, Kelm BM, Masuch R, Himmelreich U, Bachert P, Petrich W, Hamprecht FA. A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC Bioinformatics* 2009, 10:213.
58. Segal MR, Barbour JD, Grant RM. Relating HIV-1 sequence variation to replication capacity via trees and forests. *Stat Appl Genet Mol Biol* 2004, 3:2.
59. Cummings MP, Myers DS. Simple statistical models predict C-to-U edited sites in plant mitochondrial RNA. *BMC Bioinformatics* 2004, 5:132.
60. Cummings, MP, Segal MR. Few amino acid positions in rpoB are associated with most of the rifampin resistance in Mycobacterium tuberculosis. *BMC Bioinformatics* 2004, 5:137.
61. Svetnik V, Liaw A, Tong C, Culberson JC, Sheridan RP, Feuston BP. Random forest: a classification and regression tool for compound classification and QSAR

modeling. *J Chem Inf Comput Sci* 2003, 43:1947–1958.

62. Garzón MB, Blazek R, Neteler M, De Dios RS, Ollero HS, Furlanello C. Predicting habitat suitability with machine learning models: the potential area of Pinus sylvestris L. in the Iberian Peninsula. *Ecol Model* 2006, 197:383–393.
63. Evans JS, Cushman SA. Gradient modeling of conifer species using random forests. *Land Ecol* 2009, 24:673–683.
64. Cutler DR, Edwards TC, Beard KH, Cutler A, Hess KT, Gibson J, Lawler JJ. Random forests for classification in ecology. *Ecology* 2007, 88:2783–2792.
65. Hernandez PA, Franke I, Herzog SK, Pacheco V, Paniagua L, Quintana HL, Soto A, Swenson JJ, Tovar C, Valqui TH, Vargas J, Young BE. Predicting species distributions in poorly-studied landscapes. *Biodivers Conserv* 2008, 17:1353–1366.
66. Peters J, De Baets, B, Verhoest NEC, Samson R, Degroeve S, De Becker, P, Huybrechts W. Random forests as a tool for ecohydrological distribution modelling. *Ecol Model* 2007, 207:304–318.
67. Perdiguero-Alonso D, Montero FE, Kostadinova A, Raga JA, Barrett J. Random forests, a novel approach for discrimination of fish populations using parasites as biological tags. *Int J Parasitol* 2008, 38:1425–1434.
68. Nicodemus KK, Malley JD, Strobl C, Ziegler A. (2010b). The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC Bioinformatics*, 11:110.
69. Strobl C, Boulesteix AL, Kneib T, Augustin T, Zeileis A. Conditional variable importance for random forests. *BMC Bioinformatics* 2008, 9:307.
70. Breiman L, Cutler A. Random forests—classification manual, Berkeley, CA; 2008. Available at: http://www .math .usu .edu/ ~adele/ forests/ cc_graphics.htm. (Accessed 15 August 2011).
71. Strobl C, Zeileis A. Danger: high power!—exploring the statistical properties of a test for random forest variable importance. In *COMPSTAT 2008 - Proceedings in Computational Statistics*, Vol. II. Heidelberg, Germany: Physica-Verlag; 2008, 59–66.
72. Wang M, Chen X, Zhang H. Maximal conditional chi-square importance in random forests. *Bioinformatics* 2010, 26:831–837.
73. Altmann A, Toloşi L, Sander O, Lengauer T. Permutation importance: a corrected feature importance measure. *Bioinformatics* 2010, 26:1340–1347.
74. García-Magariños M, López-de Ullibarri I, Cao R, Salas A. Evaluating the ability of tree-based methods and logistic regression for the detection of SNP–SNP interaction. *Ann Hum Genet* 2009, 73:360–369.
75. Molinaro AM, Carriero NJ, Bjornson R, Hartge P, Rothman N, Chatterjee N. Power of data mining methods to detect genetic associations and interactions. *Hum Heredity* 2011, 72:85–97.
76. Szymczak S, Biernacka JM, Cordell HJ.and González-Recio O, König IR, Zhang H, Sun YV. Machine learning in genome-wide association studies. *Genet Epidemiol* 2009, 33:S51–S57.
77. Chen CC, Schwender H, Keith J, Nunkesser R, Mengersen K, P M. (2011a). Methods for identifying SNP interactions: a review on variations of logic regression, random forest and Bayesian logistic regression. *IEEE/ACM Trans Computat Biol Bioinf*, 8:1580–1591.
78. Rodenburg W, Heidema AG, Boer JMA, Bovee-Oudenhoven IMJ, Feskens EJM, Mariman ECM, Keijer J. A framework to identify physiological responses in microarray-based gene expression studies: selection and interpretation of biologically relevant genes. *Physiol Genom* 2008, 33:78–90.
79. Tang R, Sinnwell JP, Li J, Rider DN, De Andrade, M, Biernacka JM. Identification of genes and haplotypes that predict rheumatoid arthritis using random forests. *BMC Proc* 2009, 3(suppl 7):S68.
80. Bureau A, Dupuis J, Falls K, Lunetta KL, Hayward B, Keith TP, Van Eerdewegh P. Identifying SNPs predictive of phenotype using random forests. *Genet Epidemiol* 2005, 28:171–182.
81. Sakoparnig T, Kockmann T, Paro R, Beisel C, Beerenwinkel N. Binding profiles of chromatin-modifying proteins are predictive for transcriptional activity and promoter-proximal pausing. *J Comput Biol* 2012, 19:126–138.
82. Yoshida M, Koike A. Snpinterforest: a new method for detecting epistatic interactions. *BMC Bioinformatics* 2011, 12:469.
83. Jiang R, Tang W, Wu X, Fu W. A random forest approach to the detection of epistatic interactions in case-control studies. *BMC Bioinformatics* 2009, 10(suppl 1):S65.
84. Meng Y, Yang Q, Cuenco KT, Cupples LA, DeStefano AL, Lunetta KL. Two-stage approach for identifying single-nucleotide polymorphisms associated with rheumatoid arthritis using random forests and Bayesian networks. *BMC Proc* 2007, 1(suppl 1):S56.
85. Calle ML, Urrea V, Boulesteix AL, Malats N. AUC-RF: A new strategy for genomic profiling with random forest. *Hum Heredity* 2011, 72:121–132.
86. Genuer R, Poggi JM, Tuleau-Malot C. Variable selection using random forests. *Pattern Recognit Lett* 2010, 31:2225–2236.
87. Hothorn T, Leisch F. Case studies in reproducibility. *Brief Bioinf* 2011, 12:288–300.