On safari to Random Jungle: a fast implementation of Random Forests for high-dimensional data

Daniel F. Schwarz, Inke R. König* and Andreas Ziegler*

Institut für Medizinische Biometrie und Statistik, Universität zu Lübeck, Universitätsklinikum Schleswig-Holstein, Campus Lübeck, Maria-Goepert-Strasse 1, 23562 Lübeck, Germany

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: Genome-wide association (GWA) studies have proven to be a successful approach for helping unravel the genetic basis of complex genetic diseases. However, the identified associations are not well suited for disease prediction, and only a modest portion of the heritability can be explained for most diseases, such as Type 2 diabetes or Crohn's disease. This may partly be due to the low power of standard statistical approaches to detect gene–gene and gene–environment interactions when small marginal effects are present. A promising alternative is Random Forests, which have already been successfully applied in candidate gene analyses. Important single nucleotide polymorphisms are detected by permutation importance measures. To this day, the application to GWA data was highly cumbersome with existing implementations because of the high computational burden.

Results: Here, we present the new freely available software package Random Jungle (RJ), which facilitates the rapid analysis of GWA data. The program yields valid results and computes up to 159 times faster than the fastest alternative implementation, while still maintaining all options of other programs. Specifically, it offers the different permutation importance measures available. It includes new options such as the backward elimination method. We illustrate the application of RJ to a GWA of Crohn's disease. The most important single nucleotide polymorphisms (SNPs) validate recent findings in the literature and reveal potential interactions.

Availability: The RJ software package is freely available at http://www.randomjungle.org

Contact: inke.koenig@imbs.uni-luebeck.de; ziegler@imbs.uni-luebeck.de

Supplementary information: Supplementary data are available at Bioinformatics online.

Received on November 9, 2009; revised on April 16, 2010; accepted on May 16, 2010

1 INTRODUCTION

Genome-wide association (GWA) studies have become a standard approach for helping unravel the genetic basis of complex diseases. The recent successes are tremendous, and a series of new loci have been identified using single marker analyses. The recent successes are tremendous, and a series of new loci have been identified using single marker analyses (McCarthy et al., 2008; Samani et al., 2007; Wellcome Trust Case Control Consortium, 2007). Unfortunately, only a small portion of the heritability was explained by corresponding single nucleotide polymorphisms (SNPs) for most diseases such as Type 2 diabetes (6%) or Crohn’s disease (20%) (Manolio et al., 2009). Furthermore, SNPs identified by GWA studies for various diseases make poor classifiers (Jakobsdottir et al., 2009).

For overcoming such drawbacks and recognizing the complexity of the underlying biology, further mechanisms such as gene–gene interaction need to be taken into account (Moore et al., 2010). However, the discovery of gene–gene interactions using GWA studies remains challenging with traditional statistical approaches (Cordell, 2009; Moore et al., 2010). Given genotype data at different loci, an exhaustive search of interactions between all loci is the obvious way of testing interactions. Testing all two-locus interactions is computationally feasible although time demanding (Marchini et al., 2005). However, an exhaustive search of higher order interactions is computationally impractical because the number of tests increases exponentially with the order of interaction (Cordell, 2009).

One approach to deal with such large numbers of SNPs is to first perform univariate tests on each SNP, discard SNPs with high P-values and apply interaction methods, e.g. within logistic regressions, to SNP subsets afterwards (Hoh et al., 2000; Marchini et al., 2005). Unfortunately, such approaches may result in low power for SNP–SNP interactions with very small marginal effects. Another concern is genetic heterogeneity, i.e. different subsets of genes affect the same disease, and traditional statistical methods show limitations when genetic heterogeneity is present (Province et al., 2001).

A promising alternative is Random Forests (RFs; Breiman, 2001). RF was applied successfully to genetic data in various studies (Bureau et al., 2005; Chang et al., 2008; Hwang et al., 2009; McKinney et al., 2009; Sun et al., 2007), and it is anticipated that RF will help to detect gene–gene interactions in genome-wide data (Moore et al., 2010). It has been shown that RF can substantially be more efficient than standard statistical methods in ranking the true disease-associated SNPs in order to detect SNP–SNP interaction (Lunetta et al., 2004). The method is able to detect SNPs with small effects and to deal with genetic heterogeneity because separate models are automatically fit to subsets of data defined by early splits in the tree (Lunetta et al., 2004; Province et al., 2001). In addition, RF is able to handle SNPs that are associated in a non-linear fashion. The RF method is a specific data mining method. In data mining, in general, algorithms attempt to identify an unknown concept based on randomly chosen examples of the collected data. The aim is to find a prediction rule that correctly classifies new instances of the...
concept (Breiman, 2001). Thus, RF makes fewer assumptions about the functional form of the model, as required by statistical tests (McKinney et al., 2006).

A grown tree in a forest is often graphically represented by an upside down tree. Multiple paths lead through the tree from the root to different leaves via various nodes. Each node corresponds to a specific predictor variable. Thus, a path is a sequence of predictor variables (for details, see Section 2.1 and König et al., 2008). Such a predictor variable sequence includes potential interactions between them in terms of hierarchical dependencies (Cordell, 2009; Moore et al., 2010). Thus, the RF method allows for interactions between SNPs.

RF yields a classification result and a measure of the importance for each variable. Variable importance (VI) quantifies the impact of a SNP in predicting the response and may reflect a causal effect. In turn, it can be used to select the relevant SNPs from a GWA study (Ziegler et al., 2007).

Although appealing, RF has rarely been applied on the genome-wide level. In analogy to standard statistical approaches, this is due to the computational intensity and memory requirements (Zhang et al., 2009; Ziegler et al., 2007). The original RF implementation, termed RF in Fortran, by Breiman and Cutler (2004) designed to analyze low-dimensional data, i.e. a low number of SNPs, with a large number of observations, e.g. 100 SNPs and 10,000 observations. It has been successfully used, e.g. by Bureau et al. (2005) in a candidate gene case–control study involving 42 SNPs. Furthermore, the code is not user-friendly because the program has to be modified and compiled anew, whenever a new dataset is used.

An alternative implementation is the randomForest package for the programming language R (R Development Core Team, 2009) by Liaw and Wiener (2002). It is user-friendly, and it has been often used in applications (Ziegler et al., 2007). The source code of the package randomForest consists of R, C and Fortran source code. Elementary subroutines were left in Fortran code. However, the same computational and memory inefficient so that not more than 10,000 of SNPs can be analyzed on a standard machine within reasonable time and memory usage (Ziegler et al., 2007). Furthermore, the code is not user-friendly because the program has to be modified and compiled anew, whenever a new dataset is used.

An alternative implementation is the randomForest package for the programming language R (R Development Core Team, 2009) by Liaw and Wiener (2002). It is user-friendly, and it has been often used in applications (Ziegler et al., 2007). The source code of the package randomForest consists of R, C and Fortran source code. Elementary subroutines were left in Fortran code. However, the same computational and memory limitations apply as to RF in Fortran.

One approach to overcome the memory issue is to split up the GWA data into small chunks, which are subsequently analyzed separately (Jiang et al., 2009; Schwarz et al., 2007). The results of all processed chunks are finally combined. Through this, main effects are detected, but one may fail to discover some important interaction effects due to data separation. Thus, these approaches do not overcome the restrictions in detecting complex interactions.

An alternative has recently been presented by Zhang et al. (2009). In their package Willow, they compress the GWA data internally and subsequently apply RF. However, it is slow for large values of the mtry parameter (see Section 2) as recommended for datasets with many noise variables such as GWA study data (Breiman and Cutler, 2004; Liaw and Wiener, 2002). Tuning mtry for optimizing the performance of the forest is also strongly recommended (Breiman and Cutler, 2004), which can hardly be done with Willows. Using this program can be computationally intensive, thus time demanding.

Here, a novel software package called Random Jungle (RJ) is presented, which has been specifically tailored for the large-scale analysis of GWA studies. This computational and memory efficient implementation of RF is able to analyze hundreds and thousands of samples and SNPs.

In the following, we first briefly introduce the RF methodology, including the growing procedure and essential features. Next, we describe the estimation of various VI measures. Specifically, we show differences between importance scores of randomForest and RF in Fortran. After these theoretical considerations, we describe the RJ software and demonstrate its superior computational performance when compared with other implementations. Finally, we illustrate its use with data from a GWA study on Crohn’s disease.

2 METHODS

2.1 Random forests

RFs are an ensemble consisting in multiple classification and regression trees (CART) that are grown using a bootstrap sample of given data and without pruning. In general, an ensemble is a group of classifiers in which the classifier is only required to perform slightly better than random guessing or coin flipping. This property is fulfilled by many base classifiers, such as CART (Breiman, 1996; Schapire, 1990). With a CART as base classifier, a sample is classified by taking the majority vote over all tree classifiers in a forest (Breiman, 2001). RF has been shown to provide good accuracy, robustness to noise, internal estimation of error, stable classifiers and VI (Breiman, 2001; Breiman and Cutler, 2004; Meng et al., 2009). The RF procedure follows the following steps (Breiman, 2001):

1. Consider a dataset $X$, termed training data, consisting of one response variable and many predictor variables from $N$ samples. The total count of predictor variables is $M$, with $M$ being substantially larger than $N$.

2. A bootstrap sample $X^*$ consisting of $N$ samples is drawn with replacement from the original training data $X$. On average, one-third of all samples are left out due to the bootstrapping process. These samples are called ‘out-of-bag’ (OOB) data $X^*$.X.

3. A CART is grown using the bootstrap dataset $X^*$. The CART is constructed by recursively splitting data into distinct subsets, so that one parent node leads to two child nodes. For splitting data, an appropriate split rule has to be selected so that the subsets of each child node are purer than the subset of corresponding parent node. The goodness of the split is defined to be the decrease in impurity as follows: $\Delta i = \sum_{j=1}^{p} \left[ \frac{1}{2} \cdot \left( \frac{N_j}{N} \right) \cdot \left( 1 - \frac{N_{j}^*}{N} \right) \right]$, where $N$ is the proportion of samples that are labeled with class $j$ in that node. At each node, a random subset of all predictor variables is chosen without replacement to determine the best split. The size of the subset is given by the parameter mtry. Although different variables might be selected at each node to be tested, the number mtry is held constant during the procedure, and the default setting is $\text{mtry} = \sqrt{M}$, where $\lceil \cdot \rceil$ denotes the next larger integer.

4. The tree $t$ is grown to its largest extent, and no pruning proceeds. The final nodes are called terminal nodes.

5. Steps 1 to 4 are repeated to grow a specific number of trees, and, for classification, the majority vote over all trees in the resulting forest is used.

6. Finally, the OOB error fraction is calculated by classifying each sample of the OOB. Each observation is predicted by the trees for which it is an OOB observation. The prediction accuracy of the classifier is estimated by subtracting the OOB error fraction from its maximum, which is one. The prediction accuracy estimation method is a suitable surrogate for cross-validation (Breiman, 2001).

A special feature of RF is the calculation of proximities between samples.
As a result, both programs provide different scaled permutation importance scores. The estimator of randomForest is defined as

$$s^2 = \frac{1}{|T|} \sum_{t \in T} \frac{N_{OOB}(A_t - \bar{A}_t)^2}{s_t^2},$$

(3)

where \(N_{OOB}\) determines the number of samples in OOB of the current tree. The variance estimator of RF in Fortran is defined as

$$s^2 = \frac{1}{|T|} \sum_{t \in T} (A_t - \bar{A}_t)^2 - \bar{A}_t^2,$$

(4)

VI measures as described above can show a bias of correlated predictor variables such as SNPs in linkage disequilibrium (Meng et al., 2009; Nocedemus and Malley, 2009; Nocedemus et al., 2010, Strobl et al. 2008). Permuting a predictor variable using the usual permutation scheme disrupts a potential dependency structure between the permuted variable and the other predictor variables. The disruption entails an inflation of the importance value of the predictor variable when predictors were associated with the outcome (Nocedemus et al., 2010; Strobl et al., 2008).

The conditional VI (CVI) is an approach to solve this problem (Strobl et al., 2008). For preserving the dependency structure between a specific predictor variable and other predictor variables, the predictor variable in question is permuted only within groups of observations. Group assignment is determined by analyzing the corresponding dependency structure as described in detail by Strobl et al. (2008). It has been shown that the CVI reflects the importance of predictors of correlated predictors more reliably than usual importance measures (Strobl et al., 2008). Therefore, the CVI should be applied to data that contain correlated predictor variables such as SNPs in linkage disequilibrium.

### 3 IMPLEMENTATION

#### 3.1 Random jungle

The novel software package RJ implements all features of the reference implementation randomForest such as various tuning parameters, prediction of new datasets using previously grown forests, sample proximities and imputation. Commonly used VI measures are implemented, such as Gini importance, permutation importance and conditional importance measures. The features of RJ are shown in Supplementary Table 1. RJ additionally implements the variable backward elimination. When multiple CPU are available, RJ is able to perform RF on multiple CPUs simultaneously using multithreading and Message Passing Interface (MPI) parallelization.

RF in Fortran and randomForest grow ensembles of a CART, but the RF method is not restricted to a CART. Therefore, RJ serves a generalized framework for tree growing, which can be utilized to extend the set of tree types. RJ also implements CART, but several tree types such as conditional trees (Hothorn et al., 2006) are currently under construction and will be added to RJ in the future.

RJ is written in the C++ language, and the program structure fundamentally differs from the randomForest and RF in Fortran implementations. A comparison of importance scores, computing time and memory consumption across different implementations is given in Section 3.2.

The software is freely available on www.randomjungle.org, where a detailed documentation of RJ can be found.

#### 3.2 Comparison of importance values

A simulation study was set up for comparing importance score ranks of RJ with the reference implementation randomForest. To this end, we used the simulated data for rheumatoid arthritis (RA) that were provided for the Genetic Analysis Workshop (GAW) 15.
The real dataset was used to compare the different implementations on computers running the SUSE Linux operating system with a were compared using boxplots. All applications were performed ranks of predictor variables yielded by randomForest and RJ in the cases and controls combined or a \( P \)-value of 0.0001 for every CPU. RJ turned out to be the fastest program for the analysis. RJ required 1750 h for the analysis. RJ performed 159 times faster. With RJ running in a single CPU mode, a speed up of seven was still obtained. Willows required 1750h for the analysis. RJ turned out to be the fastest program for RJ and randomForest rank all variables in the same order. Results of Gini importance score and permutation importance scores investigation are shown in Supplementary Figure 1a, b and c. Both programs yield similar scores for all importance measures. All implementations are able to handle the real dataset, but computing time and memory consumption differed substantially (Fig. 1). Specifically, the randomForest and RF in Fortran analyzed the dataset in 88.8 and 84.1 h, respectively, whereas RJ performed the same analysis in only 0.53h using 40 CPUs in parallel. RF in Fortran is the fastest alternative tool. In comparison to RF in Fortran, RJ performed 159 times faster. With RJ running in a single CPU mode, a speed up of seven was still obtained. Willows required 1750h for the analysis. RJ turned out to be the fastest program for real data analysis. The computing time of all implementations is depicted in Figure 1a. The randomForest package and RF in Fortran consumed 9805 and 5421 MB memory, respectively (Fig. 1b). Considerably less memory was used by Willows, which consumed 136 MB memory. RJ spent 179 MB using one CPU. When using multiple CPUs, the program RJ distributed five processes among five computers using the MPI mode. Each process consumed 303 MB and utilized eight CPUs by using multithreading. RJ consumed more memory in the multi-processor mode because helping data structures have to be provided for every CPU. The importance scores of the SNPs and their chromosomal positions are depicted in Figure 2. The two highest peaks are located on chromosomes 1 and 16, which correspond to genes IL23R and NOD2, respectively. A comparison of positive CVI scores and two-sided \( P \)-values from the Cochran–Armitage trend test is shown in Supplementary Figure 2. The smallest \( P \)-value of all positive CVI scores is 2 \times 10^{-8}. The Pearson’s correlation coefficient between scores and \( P \)-values is 0.38, showing a moderate association between importance scores

4 RESULTS

Comparison of importance scores shows that RJ and randomForest yield similar scores for all importance measures. All implementations are able to handle the real dataset, but computing time and memory consumption differed substantially. RJ performed the same analysis in only 0.53h using 40 CPUs in parallel. RF in Fortran is the fastest alternative tool. In comparison to RF in Fortran, RJ performed 159 times faster. With RJ running in a single CPU mode, a speed up of seven was still obtained. Willows required 1750h for the analysis. RJ turned out to be the fastest program for real data analysis. The computing time of all implementations is depicted in Figure 1a. The randomForest package and RF in Fortran consumed 9805 and 5421 MB memory, respectively (Fig. 1b). Considerably less memory was used by Willows, which consumed 136 MB memory. RJ spent 179 MB using one CPU. When using multiple CPUs, the program RJ distributed five processes among five computers using the MPI mode. Each process consumed 303 MB and utilized eight CPUs by using multithreading. RJ consumed more memory in the multi-processor mode because helping data structures have to be provided for every CPU. The importance scores of the SNPs and their chromosomal positions are depicted in Figure 2. The two highest peaks are located on chromosomes 1 and 16, which correspond to genes IL23R and NOD2, respectively. A comparison of positive CVI scores and two-sided \( P \)-values from the Cochran–Armitage trend test is shown in Supplementary Figure 2. The smallest \( P \)-value of all positive CVI scores is 2 \times 10^{-8}. The Pearson’s correlation coefficient between scores and \( P \)-values is 0.38, showing a moderate association between importance scores
Crohn’s disease (Barrett SNPs are known to be strongly associated with susceptibility to were also identified by several GWA studies and corresponding be the most important genes by RJ analysis.

The four most important SNPs yielded the smallest P-values. Corresponding negative CVI scores are shown in Supplementary Figure 3.

<table>
<thead>
<tr>
<th>Gene</th>
<th>rs-Number</th>
<th>CVI score</th>
<th>Rank of CVI score</th>
<th>P-value (trend test)</th>
<th>Rank of P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEACAM4</td>
<td>rs5099916</td>
<td>5.98 × 10^-5</td>
<td>22</td>
<td>5.67 × 10^-3</td>
<td>2117</td>
</tr>
<tr>
<td>CDKAL1</td>
<td>rs9465994</td>
<td>1.30 × 10^-4</td>
<td>8</td>
<td>3.16 × 10^-4</td>
<td>158</td>
</tr>
<tr>
<td>CLSTN2</td>
<td>rs439924</td>
<td>1.05 × 10^-4</td>
<td>14</td>
<td>2.58 × 10^-5</td>
<td>18</td>
</tr>
<tr>
<td>FRN3</td>
<td>rs4527136</td>
<td>5.01 × 10^-5</td>
<td>26</td>
<td>4.73 × 10^-3</td>
<td>1807</td>
</tr>
<tr>
<td>IL23R</td>
<td>rs11209026</td>
<td>1.63 × 10^-3</td>
<td>1</td>
<td>2.00 × 10^-8</td>
<td>1</td>
</tr>
<tr>
<td>NOD2</td>
<td>rs2066843</td>
<td>5.48 × 10^-4</td>
<td>2</td>
<td>1.44 × 10^-7</td>
<td>3</td>
</tr>
<tr>
<td>PDE8B</td>
<td>rs2076756</td>
<td>3.65 × 10^-4</td>
<td>4</td>
<td>3.02 × 10^-8</td>
<td>2</td>
</tr>
<tr>
<td>PRKG1</td>
<td>rs766208</td>
<td>6.97 × 10^-5</td>
<td>20</td>
<td>4.38 × 10^-5</td>
<td>37</td>
</tr>
<tr>
<td>PRKRD</td>
<td>rs1889820</td>
<td>4.76 × 10^-5</td>
<td>27</td>
<td>1.12 × 10^-4</td>
<td>71</td>
</tr>
<tr>
<td>SNX8</td>
<td>rs11950641</td>
<td>1.05 × 10^-4</td>
<td>13</td>
<td>3.33 × 10^-5</td>
<td>31</td>
</tr>
<tr>
<td>TNFSF10</td>
<td>rs9859259</td>
<td>1.12 × 10^-4</td>
<td>12</td>
<td>1.57 × 10^-3</td>
<td>661</td>
</tr>
</tbody>
</table>

The table displays the most important genes, rs-numbers of corresponding SNPs, CVI scores, ranks of CVI scores, trend test P-values and the rank of the P-values.

and P-values. Corresponding negative CVI scores are shown in Supplementary Figure 3.

The 10 most important genes for the Crohn’s disease data are displayed in Table 1. The genes were derived by evaluating the most important SNPs that are located within genes as shown in Supplementary Table 3. The first 10 unique genes were selected for further investigations.

The four most important SNPs yielded the smallest P-values in the trend test. The interleukin 23 receptor (IL23R) and nucleotide-binding oligomerization domain containing 2 (NOD2) were found to be the most important genes by RJ analysis. IL23R and NOD2 genes were also identified by several GWA studies and corresponding SNPs are known to be strongly associated with susceptibility to Crohn’s disease (Barrett et al., 2008; Duerr et al., 2006; Rioux et al., 2007; Wellcome Trust Case Control Consortium, 2007).

Fig. 1. Comparison of computing time and memory usage of several RF implementations. Each program analyzed a simulated dataset comprising 1006 samples genotyped at 275 153 SNPs. A short bar indicates a fast implementation of RF in comparison to other programs. (a) Comparison of computing time of five implementations. The figure reads for example: For analyzing data, RJ calculations took 0.53 h. (b) Comparison of memory usage of five programs. Memory was sparsely used by Willows and RJ in comparison to randomForest and RF in Fortran. (Asterisk indicates memory usage of each computer node.)

Fig. 2. CVI scores of SNPs and their chromosomal position. The axis of CVI scores was log transformed. Small and negative CVI values were omitted.

Fig. 3. Six of the 10 most important genes can be combined to a potential pathway TNFSF10 potentially interacts with IL23R, NOD2 and PRKG1. The NOD2 conceivably interacts with CDKAL1.

A moderate association of calcyntenin 2 (CLSTN2) (P=6 × 10^-5) with Crohn’s disease was reported in literature (Rioux et al., 2007). The tumor necrosis factor superfamily member 10 (TNFSF10) was high ranked by RJ analysis. The traditional trend test of TNFSF10 yielded P=0.001576. TNFSF10 is involved in apoptosis and proliferation of human colon cancer cells (Baader et al., 2005; Saad et al., 2007; Tang et al., 2002; Tillman et al., 2003). Colorectal cancer and Crohn’s disease are related due to the fact that relative risk of colorectal cancers is significantly raised in Crohn’s disease (Canavan et al., 2006; Ekborn et al., 1990).

Five of the 10 most important genes can be linked to a potential pathway by consulting additional genes, proteins and transcripts. The potential pathway is shown in Figure 3. The IL23R and
TNFSF10 potentially interact via the signal transducer and activator of transcription 3 (STAT3; Niu et al., 2001; Parham et al., 2002). TNFSF10 possibly interacts with PKR1 via Sp1 transcription factor (SP1; Sellak et al., 2002; Xu et al., 2008). TNFSF10 contingently interacts with NOD2 via nuclear factor of kappa light polypeptide gene enhancer in B-cells 1 (NFKB1; Baetu et al., 2001; Gutierrez et al., 2002). Finally, NOD2 conceivably interacts with CDK5 regulatory subunit associated protein (CDKAL1) via hepatocyte nuclear factor 4 alpha (HNF4A; Odom et al., 2004; Rioux et al., 2007).

5 DISCUSSION

The RF method was applied to genome-wide data using RJ and assigned a score of importance to each SNP. The resulting list of SNPs was investigated for potential interactions.

The results of the real data analysis validated the findings of GWA studies such as NOD2 and IL23R. Results give also evidence of new potential interactions between genes that are associated with Crohn’s disease. Specifically, the TNFSF10 was not found to be strongly associated with Crohn’s disease by traditional statistical tests. In contrast, RJ analysis detected that TNFSF10 potentially interacts with NOD2, PKR1 and IL23R. STAT3 is considered to be a hub between TNFSF10 and IL23R, which was shown to be moderately associated with Crohn’s disease. The TNFSF10 is involved in apoptosis of human colon cancer cells. TNFSF10 might possibly explain a part of the high risk of colorectal cancers in Crohn’s disease patients.

However, interpreting results and assessing biological plausibility is challenging. Results may show false positives. Thus, further investigation is needed to validate this specific causal relationship. Furthermore, although RF has the ability to detect very small main effects, its power to identify interactions depends on the presence of main effects. Thus, gene–gene interaction with no marginal effects might be left unrevealed when RF is applied.

Nevertheless, an impressive computational efficiency and memory management of RJ allow for analyzing high-dimensional data in an acceptable amount of time. Analyzing GWA data comprising thousands of observations and a million SNPs seem to be feasible with respect to time and memory consumption. The software computes up to 159 times faster than the fastest alternative implementation, and the program shows the same importance software computes up to 159 times faster than the fastest alternative implementation, and the program shows the same importance.

In summary, RJ is a promising software package for applying RF method to high-dimensional data such as GWA data. The application of RF to GWA data may help to identify potential interacting SNPs that were not found by traditional statistical approaches.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful suggestions and comments. The NIDDK IBDGC Crohn’s Disease GWA Study was conducted by the NIDDK IBDGC Crohn’s Disease GWA Study Investigators and supported by the National Institute of Diabetes and Digestive and Kidney Diseases (NDDK). This manuscript was not prepared in collaboration with Investigators of the NIDDK IBDGC Crohn’s Disease GWA Study and does not necessarily reflect the opinions or views of the NIDDK IBDGC Crohn’s Disease GWA Study or the NIDDK.

Funding: DFG (KO 2250/3-1); intramural funding from Medical Faculty of the University at Lübeck (E32-2009, SPP2); NIH grants 5ROI-HL049609-14, 1RO1-AG021917-01A1; University of Minnesota; Minnesota Supercomputing Institute; GAW grant, R01-GM051575; ENGAGE (grant agreement number 201413); Atherosgenomics (grant agreement number 01GS0831).

Conflict of Interest: none declared.

REFERENCES


Jiang,R. et al. (2009) A random forest approach to the detection of epistatic interactions in case-control studies. BMC Bioinformatics, 10 (Suppl. 1), S65.


Meng Y. et al. (2009) Performance of random forest when SNPs are in linkage disequilibrium. BMC Bioinformatics, 10, 78.


