ABSTRACT

Motivation: Feature selection is one of the main challenges in analyzing high-throughput genomic data. Minimum redundancy maximum relevance (mRMR) is a particularly fast feature selection method for finding a set of both relevant and complementary features. Here we describe the mRMRe R package, in which the mRMR technique is extended by using an ensemble approach to better explore the feature space and build more robust predictors. To deal with the computational complexity of the ensemble approach, the main functions of the package are implemented and parallelized in C using the openMP Application Programming Interface.

Results: Our ensemble mRMR implementation outperforms the classical mRMR approach in terms of prediction accuracy. They identify genes more relevant to the biological context and may lead to richer biological interpretations. Let an ensemble composed of multiple mRMR classifiers be the primary motivator for the development of a new R package, mRMRe, which implements an ensemble variant of mRMR, in which multiple feature sets, rather than a single list of features, is built. Also included in the package is a function for computing a mutual information matrix (MIM) based on the appropriate estimators for each variable type (i.e. continuous, discrete and survival data). Both these package features have been adapted to fully use multicore platforms.

2 METHODS

The \( \mu \)nu function computes a MIM using a linear approximation based on correlation such that mutual information (MI) is estimated as \( I(x, y) = \frac{1}{2} \ln(1 - \rho(x, y)^2) \), where \( I \) and \( \rho \), respectively, represent the MI and correlation coefficient between variables \( x \) and \( y \). Correlation between continuous variables can be computed using either Pearson’s or Spearman’s estimators, whereas Cramer’s \( V \) is used for correlation between discrete variables and Somers’ Dxy index is used for correlation between continuous variables and survival data.

The mRMRe technique, as implemented in the mRMR.classic function, allows an efficient selection of relevant and non-redundant features (Ding and Peng, 2005). Let \( y \) be the output variable and \( X = \{x_1, \ldots, x_n\} \) be the set of \( n \) input features. The method ranks \( X \) by maximizing the MI with \( y \) (maximum relevance) and minimizing the average MI with all the previously selected variables (minimum redundancy). Let \( x_i \) be the feature with highest MI with the output variable and thus selected first

\[
   x_i = \arg \max_{x \in X} I(x, y)
\]

1 INTRODUCTION

In genomic data analysis, phenotype-associated feature selection is of utmost importance in understanding the biological processes underlying the relevant phenotype and in building accurate predictive models. This is not a trivial task for high-throughput genomic data (microarray, next-generation sequencing, etc.), as they are high dimensional, noisy and have complex intercorrelational structures. Because feature selection is an nondeterministic polynomial time (NP)-hard problem, one must resort to the use of heuristics, which find suitable and sub-optimal sets of relevant features in high-dimensional datasets. Among these heuristics, the minimum redundancy maximum relevance (mRMR) feature selection technique is particularly appealing because of the relatively low computational complexity of its algorithm for finding a set of relevant and complementary features (Ding and Peng, 2005), from which accurate predictive models are developed. The challenge lies in that mRMR, like all feature selection algorithms in a low sample-to-dimensionality ratio setting, produces highly variable results, and small changes in sample data often lead to dramatically different sets of selected features.

In the past decade, a new trend has emerged whereby highly accurate ‘ensemble’ classifiers are produced by combining less accurate ones, on the grounds that model variance is reduced without affecting bias (Kittler et al., 1998). However, the computational cost of an ensemble variant of the mRMR method is high, as multiple mRMR feature selections must be done. This was the primary motivator for the development of a new R package, mRMRe, which implements an ensemble variant of mRMR, in which multiple feature sets, rather than a single list of features, is built. Also included in the package is a function for computing a mutual information matrix (MIM) based on the appropriate estimators for each variable type (i.e. continuous, discrete and survival data). Both these package features have been adapted to fully use multicore platforms.
The set of selected features, denoted by $S$, is then initialized with $x_i$. Next, another feature is added to $S$ by choosing the feature having the highest relevance with the output variable and the lowest redundancy with the previously selected features, thus maximizing the score $q_j$ at step $j$

$$q_j = I(x_j, y) - \frac{1}{|S|} \sum_{k \in S} I(x_j, x_k)$$  \hspace{1cm} (2)

This step is repeated until the desired solution length has been attained. We have implemented this approach for continuous/survival and discrete variables, also referred to as F-test Correlation Difference (FCD) and Mutual Information Difference (MID) schemes in Table 1 of (Ding and Peng, 2005), respectively.

Although mRMR is a fast and greedy heuristic, it is not guaranteed to find a global optimal solution should one exist. Alternative feature subsets of equivalent or better quality than the one identified may exist. Moreover, the features selected by a single mRMR run are unlikely to adequately account for the diversity of the biological processes associated with the phenotype under study.

To alleviate these problems we implemented two ensemble approaches to generate multiple mRMR solutions in parallel; these two techniques are referred to as exhaustive and bootstrap ensemble mRMR. The exhaustive variant extends the classical mRMR heuristic by initializing multiple feature selection procedures with the first selected feature is guaranteed to be different. The pseudo-code describing the algorithm implementing the exhaustive ensemble mRMR feature selection is provided in Algorithm 1 in Supplementary Material.

The bootstrap variant resamples (with replacement) the original dataset to generate $k$ bootstraps, and classical mRMR feature selection is performed in parallel for each of the bootstrapped datasets, thus generating $k$ mRMR solutions. The pseudocode describing the algorithm implementing the bootstrap ensemble mRMR feature selection is provided in Algorithm 2 in Supplementary Material.

A considerable share of the computational complexity of existing mRMR packages is due to inefficient computation of the MIM. These, although mRMR is a fast and greedy heuristic, it is not guaranteed to find a global optimal solution should one exist. Alternative feature subsets of equivalent or better quality than the one identified may exist. Moreover, the features selected by a single mRMR run are unlikely to adequately account for the diversity of the biological processes associated with the phenotype under study.

To alleviate these problems we implemented two ensemble approaches to generate multiple mRMR solutions in parallel; these two techniques are referred to as exhaustive and bootstrap ensemble mRMR. The exhaustive variant extends the classical mRMR heuristic by initializing multiple feature selection procedures with the $k>1$ most relevant features. Subsequently, $k$ mRMR solutions are produced in parallel, in which the first selected feature is guaranteed to be different. The pseudocode describing the algorithm implementing the exhaustive ensemble mRMR feature selection is provided in Algorithm 1 in Supplementary Material.

The bootstrap variant resamples (with replacement) the original dataset to generate $k$ bootstraps, and classical mRMR feature selection is performed in parallel for each of the bootstrapped datasets, thus generating $k$ mRMR solutions. The pseudocode describing the algorithm implementing the bootstrap ensemble mRMR feature selection is provided in Algorithm 2 in Supplementary Material.

A considerable share of the computational complexity of existing mRMR packages is due to inefficient computation of the MIM. These, such as minet (Meyer et al., 2008) and sideChannelAttack (Lerman et al., 2011), compute the MIM completely before performing the mRMR feature selection. However, only a small portion of the MIM is generally required to compute mRMR scores (Equations 1 and 2) during the feature selection process. We accelerated our mRMR implementation by computing the MI score between features in a lazy-evaluation manner, computing them only when needed. This significantly reduces the run-time complexity of mRMR and is a critical point for controlling the computational time of the ensemble variants.

3 CASE STUDY

In this case study, we assess the benefits of using the exhaustive and bootstrap ensemble mRMR feature selection methods (referred to as mRMR.e and mRMR.b, respectively) by analyzing two recently published pharmacogenomic datasets generated within the Cancer Genome Project [CGP; (Garnett et al., 2012)] and the Cancer Cell Lines Encyclopedia [CCLE; (Barrett et al., 2012)]. In these large datasets of cancer cell lines, the authors measured sensitivity (IC_{50}) to Irinotecan (Camptothecin), a therapeutic drug mainly used in colon cancer. This metric was used to discriminate between resistant and sensitive cell lines. Similar to Papillon-Cavanagh et al. (2013), we used CGP as a training set, whereas CCLE was split in two validation sets: CCLE COMMON contains cell lines common to both CGP and CCLE (471), whereas CCLE NEW contains cell lines absent in CGP (565).

We then implemented five feature selection methods: (i) SINGLEGENE and (ii) RANK consist in ranking the features by correlation with drug sensitivity so as to, respectively, select the first and the top $n$ features; (iii) mRMR is used to select the most relevant, less redundant set of $n$ features associated to drug sensitivity; (iv) mRMR.e and (v) mRMR.b perform multiple mRMR feature selections in parallel to identify 200 mRMR solutions using the exhaustive and bootstrap approach, respectively. Feature selection was followed by linear regression model fitting using the selected features to predict drug sensitivity. For mRMR methods, drug sensitivity is predicted by averaging predictions of the 200 multivariate models corresponding to each mRMR solution.

To strike a balance between model complexity, considered here as the number of selected features (solution length), and prediction performance, we performed 100 resamplings of the training set and estimated the concordance index (Harrell et al., 1996) of the resulting predictive model with respect to feature selection method and solution length (Fig. 1A). As the concordance index is a generalization of the area under the receiver-operating characteristic curve, high-performing models are associated with index values close to 1, and random models are expected to yield index values close to 0.5. We observed that mRMR methods yielded higher performance (Wilcoxon signed rank test $P < 0.001$, see Supplementary Table S1), especially for small ($\leq 5$) solution lengths. As expected, the gain in performance rapidly diminished with increasing solution length (Supplementary Fig. S1). We consequently selected 15 as the solution length, exhibiting the most balance between model complexity and performance (Fig. 1B). In addition, we computed the variance of the concordance index over multiple resamplings (Fig. 1B and Supplementary Fig. S2) and observed that the classical and ensemble mRMR variants produced lower variance when compared with RANK or SINGLEGENE. Variance was much lower for ensemble mRMR techniques when compared with classical mRMR techniques for small ($\leq 5$) solution lengths. However, no difference was observed for larger solution lengths (Supplementary Fig. S2).

In addition to performance assessment diversity (Tsybulev et al., 2005) and stability (Guzmán-Martínez and Aláz-Rodríguez, 2011; Kuncheva, 2007) of the mRMR.e feature selection techniques implemented were investigated (Supplementary Fig. S3). Although RANK and mRMR select different genes during each resampling of the training set, the mRMR.e techniques identify multiple, possibly diverse, mRMR solutions in parallel. In fact, we observed that, on average, mRMR.e selects 210 distinct features shared between the 200 distinct mRMR solutions (Supplementary Fig. S3A), whereas
mRMRe.b is much more diverse with an average of 988 distinct features (Supplementary Fig. S3B). The greater diversity of mRMRe.b can partly explain the better performance observed for ensemble mRMR approaches, especially mRMRe.b because it identifies a more diverse panel of mRMR solutions (Supplementary Fig. S4).

We further investigated the advantage of using mRMRe for biological interpretation by performing a Gene Ontology analysis using the GOSim package. Because ensemble mRMR techniques, by their very nature, select more genes than the other techniques, we expect more biological terms to be significantly enriched with the former. This is indeed the case, as illustrated in Supplementary Figure S5: 100, 21, 167 and 400 Gene Ontology terms have been identified as significantly enriched (Fisher’s exact test $P < 0.05$) for RANK, mRMR, mRMRe.e and mRMRe.b, respectively (Supplementary File 3).

Finally, we compared the run-time performance of our functions with those of the sideChannelAttack package (Lerman et al., 2011); the latter implements the classical mRMR and relies on the miner package (Meyer et al., 2008) to build the MIM. As reported in Table 1, our package, on a 8-core workstation, is twice as fast for full MIM construction and 9.2 times faster for classical mRMR feature selection; this is mainly because of the lazy MIM implementation. It is worth noting that the search for 200 mRMR solutions using mRMRe.e is still 5.9 times faster than the classical mRMR with sideChannelAttack. Moreover, performing ensemble mRMR feature selection using the bootstrap method is as computationally demanding, as a new (lazy) MIM must be computed for each bootstrap.

### 4 CONCLUSION

The R package mRMRe provides functions to efficiently perform ensemble mRMR feature selection by taking full advantage of parallel computing. Ensemble mRMR can be beneficial from both a predictive (lower bias and lower variance) and biological (more thorough feature space exploration) point of view, which makes it particularly attractive for high-throughput genomic data analysis.

**Funding:** B.H.K’s start-up funds (to N.D.J. and S.P.C.). Belgian French Community ARC funding (C.O. and G.B.).

**Conflict of Interest:** none declared.

### REFERENCES


