宏基因组中可移动序列的精确检测问题研究

彭 超[1,2] 王 普[1,2] 葛瑞泉[1,2] 周丰丰[1]

[1](中国科学院深圳先进技术研究院 深圳 518055)
[2](中国科学院大学深圳先进技术学院 深圳 518055)

摘 要 基因组组装是宏基因组分析的主要挑战之一。通常假设所有测序序列均来源于同一个基因组，微生物中非常活跃的可移动元件给这个前提假设提出了重大质疑。文章将该质疑抽象为可移动元件与宿主染色体之间的二分类问题，准确的二分类性能将进一步促进宏基因组学方面的研究。基于宏基因组测序数据的数值化特征，详细考察特征选择算法 ReliefF、卡方检验和 Fisher 判别 *t* 检验，并结合分类模型逻辑回归、极限学习机、支持向量机和随机森林，验证最优可移动元件检测模型的性能。实验结果表明，ReliefF 特征选择算法和随机森林分类算法的融合模型，使用 100 个特征即可正确分类 95% 以上的宏基因组测序数据，优于使用全部的 690 个特征。

关键词 基因分类；数据挖掘；特征选择；基因组条形码
中图分类号 TG 156 **文献标志码** A

Accurate Detection of Mobile Sequence in Metagenome

PENG Chao[1,2] WANG Pu[1,2] GE Ruiquan[1,2] ZHOU Fengfeng[1]

[1](*Shenzhen Institutes of Advanced Technology*, *Chinese Academy of Sciences*, *Shenzhen* 518055, *China*)
[2](*Shenzhen College of Advanced Technology*, *University of Chinese Academy of Sciences*, *Shenzhen* 518055, *China*)

Abstract Genome assembling is one of the challenges in metagenomic analysis. It is usually assumed that the sequencing reads are from the same genome. However, the mobile elements active in microbial genomes raise a critical question mark on this assumption. This work formulated this issue as a binary classification problem. The accurate discrimination of mobile elements from chromosomes could greatly facilitate the metagenomic analysis. After quantifying the sequencing reads in metagenome, the collaboration of binary classification algorithms with feature selection algorithms, including ReliefF, chi-squared test, and Fisher's *t*-test was investigated. All feature subsets were tested using the classification algorithms such as logisitic regression, extreme learning machine, support vector machine and random forest. Experimental results demonstrate that the model based on ReliefF algorithm and Random Forest algorithm achieves over 95% in accuracy with only 100 features, which outperforms the model utilizing all 690 features.

收稿日期：2015-11-07 **修回日期**：2015-12-25
作者简介：彭超，硕士研究生，研究方向为大数据并行挖掘算法；王普，博士研究生，研究方向为健康大数据融合及多标签分类算法研究；葛瑞泉，博士研究生，研究方向为健康大数据并行优化算法；周丰丰(通讯作者)，研究员 ，研究方向为异构健康大数据融合及多模整合性知识挖掘，E-mail：FengfengZhou@gmail.com。

Keywords gene classification; data mining; feature selection; genomic barcode

1 引 言

宏基因组(metagenome)是将环境中多个微生物细胞混合在一个样本中，采用当代基因组测序技术获取样本所有脱氧核糖核酸(DNA)数据的一种生物组学技术。宏基因组分析的主要挑战来自于宏基因组数据来源非常复杂，序列片段可能来自于不同的生物物种。宏基因组数据提供了所在环境中所有活跃的微生物种群信息，对研究人类疾病、生物质能源和自然界生命进化等重大问题具有关键作用。基因组序列组装(assembling)是宏基因组分析的基础，可以为进一步的基因预测、功能分析和差异性进化分析提供模版。普通微生物基因组组装问题的前提假设是——测序获得的所有脱氧核糖核酸数据均来源于同一个基因组，而宏基因组无法满足这个前提假设。微生物细胞中的质粒是可以在不同细胞之间转移的环形脱氧核糖核酸分子，其可移动特性给宏基因组组装问题提出了极大挑战。如果能够通过计算的方法将质粒序列数据与微生物染色体序列数据精准区分开，将显著降低宏基因组分析的难度[1]。

计算“k 字符短串”(kerner)出现的频率是基因分析工作最常见的研究方法，国内外发表了许多相关的研究成果。不同物种的基因序列会呈现不同的“k 字符短串”频率特征，为研究不同物种的基因特征提供了重要的依据，Robin 等[2]、Reinert 等[3]以及 Chor 等[4]均研究了“k 字符短串”频率在不同物种基因序列上的分布特征，并发表了一些研究成果。Kurtz 等[5]利用“k 字符短串”的频率分布特征，设计出了一种检测玉米、高粱和大米基因中的转子的方法。Macas 等[6]利用“k 字符短串”频率特征对卫星脱氧核苷酸的重复性进行了分析。Pevzner 等[7]就此做了基因组装分析的研究。

然而，由于质粒不仅可以在不同微生物细胞之间转移，还可以将部分序列插入到宿主染色体上。微生物宏基因组的分析工作中，目前精准区分来源于质粒或染色体的序列还有一定难度，相关方面的研究非常少，其中 cBar[1]是非常具有代表性的研究。该算法将该问题抽象为二分类数据挖掘模型，提取序列的“五字符短串”出现频率作为特征向量，进行二分类建模。通过十倍交叉验证详细测试 C4.5 决策树[8]、贝叶斯网络(Bayesian Network)、支持向量机-径向基内核函数(SVM-RBF kernel)[9]、序列最小优化(Sequential Minimal Optimization，SMO)[10]、神经网络(Neural Networks)[11]等分类算法的性能指标，取得了超过 80% 的接收器操作特性(Receiver Operating Characteristic，ROC)曲线下面积(Area Under roc Curve，AUC)性能。其中支持向量机(Support Vector Machine，SVM)算法的表现最优，AUC 可达到 0.915，表明可分类正确 91.5% 的数据。但是 cBar 算法未考虑不同长度的“k 字符短串”出现频率的组合效应及去除不相关特征的噪音影响。本文进行的实验综合考虑了 k 介于 2～5(包含)的所有“k 字符短串”组合，并且引入了特征选择思想，采用了逻辑斯蒂回归(Logistic Regression)[10]、SVM-RBF kernel、极限学习机(Extreme Learning Machine，ELM)[12]和随机森林(Random Forests)[13]等四种分类算法进行分类建模。其中，Random Forests 算法的表现最优，AUC 性能可达到 0.95，即可分类正确 95% 的数据，相对于 cBar 有较大的提升。

实验结果表明，文章提出的质粒特异性特征将有助于区分质粒与染色体序列的假设，是有效的。通过详细考察多个特征选择算法的性

能，比较分析不同分类建模算法的分类性能，给出了最优化质粒与染色体区分模型。最优模型的 AUC 性能远超过现有模型性能，相关成果预计将对宏基因组学数据的研究工作起到较好促进作用。

2　数据与方法

本实验于 2015 年 9 月 23 日从美国国家生物信息中心下载了完整微生物基因组 Bacteria 的所有序列文件，其中包括 3 198 条质粒序列和 2 044 条染色体序列。数据文件格式为 FASTA，包含了 A、C、G、T 四个碱基字符组成的序列数据，用文本的形式描述了基因序列的一维结构。

本工作将采用的数据挖掘算法，需要输入数据为连续数值型向量。首先，由于质粒分子可以将部分自身序列复制插入进宿主基因组中，所以通过 NCBI BLAST 软件包中的 BLASTN 功能检测出染色体序列中与质粒序列完全一样的部分、并去除后做进一步分析。

进而采用类似于文献[14]中的“反向互补序列对”出现频率作为给定一条脱氧核糖核酸的数值型特征数据，即计算该序列中所有 k 字符短串“反向互补序列对”的出现频率，其中 k 取值为 2～5。k 为奇数时，“反向互补序列对”数目 $N(k)=4k/2$；反之，k 为偶数时，$N(k)=(4k+4k/2)/2$。

因此，实验根据 k 的取值共提取出 690 个特征。提取特征之后，生成正负训练样本，选择数据挖掘算法进行模型训练，采用十倍交叉验证方法验证不同模型的表现。本实验采用的衡量指标有 AUC、敏感度(Sensitivity)、特异性(Specificity)。为优化模型的分类性能，实验对 690 个特征进行了特征选择，最优的特征组合能够很好地优化分类性能，并且对识别染色体序列和质粒序列具有重要意义。

3　实验设计

3.1　数据预处理

可移动的质粒分子以较高频率侵入宿主细胞染色体序列中，与染色体序列进行拼接，导致染色体序列含有大量的质粒序列片段。因此需要对 Bacteria 当中的染色体序列进行处理，找出每条染色体当中含有的质粒序列片段，并去除这些片段。实验先采用 NCBI BLAST 软件包的 BLASTN 功能对染色体序列和质粒序列进行同源性比对，找出每条染色体上质粒片段所处的位置；然后编写 JAVA 程序对染色体序列进行切割并去除质粒片段。在对 2 044 条染色体进行比对和切割之后，得到 955 102 个染色体序列片段，此数据量大大超过了 3 198 条的质粒序列，会导致正负样本训练集失去平衡，导致模型分类效果不佳，所以在其中随机抽取了 4 915 条长度大于 20 000 bp 的染色体序列，作为后续实验的正类训练样本，将质粒序列作为负类训练样本。

3.2　数值型特征提取

本实验提取的特征参考绘制基因条形码采用的特征，即 k-mer($1<k<6$)与其反向互补序列(反向互补序列对)在染色体序列或质粒序列中出现的频率，频率的计算公式如下所示。

$$fr=count/\ length \tag{1}$$

其中，fr 表示频率；$count$ 表示出现频数；$length$ 表示染色体序列或质粒序列的长度。

实验编写了 JAVA 程序用于提取特征，程序维护了两张表，一张表存储染色体序列的样本，另一张表存储质粒序列的样本。其中，每一张表有 690 列，每一列代表每一对反向互补序列对，表的行数等于样本数，因此染色体序列表的大小为 4 915 行 690 列，质粒序列表的大小为 3 198 行 690 列。生成这两张表，将作为模型训练用的正样本和负样本：实验设置染色体序列为正样本，类标号为 1；质粒序列为负样本，类标号

为 0。

3.3 模型训练

实验选取 Logistic Regression、SVM（RBF kernel）、极限学习机和 Random Forests 四种算法进行模型训练。

Logistic Regression 算法是二分类算法中较简单直接的一种算法，它训练的模型指标可作为后续三种进阶算法的参考基准。二项 Logistic Regression 模型的条件概率分布定义为：

$$P(Y=1|\boldsymbol{x})=\frac{e^{\boldsymbol{w}\cdot\boldsymbol{x}+b}}{1+e^{\boldsymbol{w}\cdot\boldsymbol{x}+b}},P(Y=0|\boldsymbol{x})=\frac{1}{1+e^{\boldsymbol{w}\cdot\boldsymbol{x}+b}} \quad (2)$$

核函数为 RBF 的支持向量机（SVM），基本模型是定义在特征空间上间隔最大的线性分类器[15]，它的基本思想是在特征空间找到能够正确划分训练数据集并且集合间隔最大的分离超平面。对于线性可分的数据集，可采用线性可分支持向量机算法；对于非线性可分的数据集，需要利用核函数通过一个非线性变换将输入空间（欧式空间或离散集合）转换到一个特征空间（希尔伯特空间），使得在输入空间中的超曲面模型对应于特征空间中的超平面模型（支持向量机）。常用的核函数有多项式核函数、高斯核函数、字符串核函数。本实验采用的是高斯核函数，公式定义如下：

$$K(\boldsymbol{x},\boldsymbol{z})=e^{-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}} \quad (3)$$

高斯核函数对应的支持向量机为高斯径向基分类器，分类决策函数的定义如下所示，其中 a_i^* 和 b^* 表示对偶问题的解。

$$f(x)=\mathrm{sgn}\left(\sum_{i=1}^{N_s}\boldsymbol{a}_i^*\boldsymbol{y}_i e^{-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}}+\boldsymbol{b}^*\right) \quad (4)$$

极限学习机算法是 Huang GB 等[16]提出的求解单隐含层神经网络的算法，相对于传统的神经网络，尤其是单隐含层前馈神经网络，极限学习机学习的速度有非常大的提高。假设单隐含层有 $\tilde{N}$ 个神经元（$\tilde{N}$ 可以随意取值，一般取大于等于 N 的值），N 表示训练样本数。对于 N 个训练样本 $(\boldsymbol{x}_i,\boldsymbol{t}_i)$，$\boldsymbol{x}_i=[x_{i1},x_{i2},\cdots,x_{in}]^{\mathrm{T}}\in\mathbb{R}^n$，$\boldsymbol{t}_i=[t_{i1},t_{i2},\cdots,t_{im}]^{\mathrm{T}}\in\mathbb{R}^m$，因此利用激励函数 $g(x)$ 建模如下所示：

$$\sum_{i=1}^{\tilde{N}}\boldsymbol{\beta}_i g(\boldsymbol{w}_i\cdot x_j+\boldsymbol{b}_i)=\boldsymbol{o}_j,j=1,\cdots,N \quad (5)$$

$\boldsymbol{w}_i=[\boldsymbol{w}_{i1},\boldsymbol{w}_{i2},\cdots,\boldsymbol{w}_{in}]^{\mathrm{T}}$ 是隐含层链接输入层的权重向量，$\boldsymbol{\beta}_i=[\boldsymbol{\beta}_{i1},\boldsymbol{\beta}_{i2},\cdots,\boldsymbol{\beta}_{im}]^{\mathrm{T}}$ 是隐含层链接输出层的权重向量，$\boldsymbol{b}_i$ 表示第 i 个神经元的偏移量。选取的激励函数 $g(x)$ 可以零误差的方式即 $\sum_{j=1}^{\tilde{N}}\|\boldsymbol{o}_j-\boldsymbol{t}_j\|=0$ 模拟目标输出[16]，公式如下所示。

$$\sum_{i=1}^{\tilde{N}}\boldsymbol{\beta}_i g(\boldsymbol{w}_i\cdot\boldsymbol{x}_j+\boldsymbol{b}_i)=\boldsymbol{t}_j,j=1,\cdots,N \quad (6)$$

以上公式可向量化表示为

$$\boldsymbol{H}\boldsymbol{\beta}=\boldsymbol{T} \quad (7)$$

其中，

$$\boldsymbol{H}(\boldsymbol{w}_1,\cdots,\boldsymbol{w}_{\tilde{N}},\boldsymbol{b}_1,\cdots,\boldsymbol{b}_{\tilde{N}},\boldsymbol{x}_1,\cdots,\boldsymbol{x}_N)=$$

$$\begin{bmatrix} g(\boldsymbol{w}_1\cdot\boldsymbol{x}_1+\boldsymbol{b}_1) & \cdots & g(\boldsymbol{w}_{\tilde{N}}\cdot\boldsymbol{x}_1+\boldsymbol{b}_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(\boldsymbol{w}_1\cdot\boldsymbol{x}_N+\boldsymbol{b}_1) & \cdots & g(\boldsymbol{w}_{\tilde{N}}\cdot\boldsymbol{x}_N+\boldsymbol{b}_{\tilde{N}}) \end{bmatrix}_{N\cdot\tilde{N}};$$

$$\boldsymbol{\beta}=\begin{bmatrix}\boldsymbol{\beta}_1^{\mathrm{T}}\\ \vdots \\ \boldsymbol{\beta}_{\tilde{N}}^{\mathrm{T}}\end{bmatrix}_{\tilde{N}\cdot m};\quad \boldsymbol{T}=\begin{bmatrix}\boldsymbol{t}_1^{\mathrm{T}}\\ \vdots \\ \boldsymbol{t}_N^{\mathrm{T}}\end{bmatrix}_{N\cdot m}。$$

为了训练单隐含层神经网络，我们期望得到 $\hat{\boldsymbol{w}}_i$、$\hat{\boldsymbol{b}}_i$、$\hat{\boldsymbol{\beta}}_i$，使得

$$\left\|\boldsymbol{H}(\hat{\boldsymbol{w}}_i,\hat{\boldsymbol{b}}_i)\hat{\boldsymbol{\beta}}_i-\boldsymbol{T}\right\|=\min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\beta}}\left\|\boldsymbol{H}(\boldsymbol{w}_i,\boldsymbol{b}_i)\boldsymbol{\beta}_i-\boldsymbol{T}\right\| \quad (8)$$

这等价于损失函数的最小化

$$E=\sum_{j=1}^{N}\left(\sum_{i=1}^{\tilde{N}}\boldsymbol{\beta}_i g(\boldsymbol{w}_i\cdot\boldsymbol{x}_j+\boldsymbol{b}_i)-\boldsymbol{t}_j\right)^2 \quad (9)$$

前馈神经网络通过梯度下降法迭代调整所有参数，收敛速度较慢，而且容易收敛到局部最小值，而全局最小值才是符合期望的。极限学习机算法在此基础上进行了改进，当输入权重 $\boldsymbol{w}_i$ 和

隐含层的偏移量 $\boldsymbol{b}_i$ 被随机确定，隐含层的输出矩阵 $\boldsymbol{H}$ 就被唯一确定，因此输出权重 $\boldsymbol{\beta}$ 的求解可转换为求解

$$\hat{\boldsymbol{\beta}}=\boldsymbol{H}^{\dagger}\boldsymbol{T} \tag{10}$$

其中，$\boldsymbol{H}^{\dagger}$ 是矩阵 $\boldsymbol{H}$ 的 Moore-Penrose 广义逆。极限学习机算法就是通过求解广义逆的过程来求解出最小且唯一的 $\hat{\boldsymbol{\beta}}$，这种方法的训练速度比前馈神经网络快上千倍，模型的学习效果也优于前馈神经网络[16]。

Random Forests 算法集成了多棵决策树，它通过有放回采样输入样本和特征来建立一棵决策树。每次输入样本进行预测时，每棵决策树对样本进行所属类投票，输出选择得票多的类。Random Forests 通过抽样的方法建立决策树，相当于进行了特征选择的过程，还可以很好地避免过拟合的问题。

3.4　特征选择

本文采用特征选择算法，来搜寻对分类染色体和质粒序列最有效的特征数据。特征选择方法可以划分为过滤式(Filter)和封装式(Wrapper)两类。过滤式方法是按照特定的度量值来评价特征与类之间的相关性，度量方法一般包括距离度量、信息度量、依赖性度量和一致性度量[17]。按照特定的度量值对特征进行排序，属于过滤式方法，因为只需计算度量值并对特征进行排序，这种方法对计算性能要求较低，另外，由于加入了偏移量能够很好地预防过拟合[18]，因此本文都采用过滤式方法，选取 ReliefF[19]、chi-square、Fisher、*t*-test 这四种方法进行特征选择。

ReliefF 基于 Relief[20]算法进行改进，对于数据缺失和噪音数据有更好的鲁棒性。基本原理是每次随机选取一个样本实例，根据欧氏距离找出 k 个同类的最近邻(Nearest Hits) $\boldsymbol{H}_j$ 和 k 个不同类的最近邻(Nearest Miss) $\boldsymbol{M}_j(C)$，C 代表不同的所属类。然后更新所有特征的权重 $W[\boldsymbol{A}]$[21]，更新方程如公式(11)所示，$\boldsymbol{A}$ 表示特征向量。其中，$P(C)$表示不同类的先验概率；$P(class(\boldsymbol{R}_i))$表示样本 $\boldsymbol{R}_i$ 所属类的先验概率。

$$W[\boldsymbol{A}]:=W[\boldsymbol{A}]-\sum_{j=1}^{k}\mathrm{diff}(\boldsymbol{A},\boldsymbol{R}_i,\boldsymbol{H}_j)/(m\cdot k)+\sum_{C\neq class(\boldsymbol{R}_i)}\left[\frac{P(C)}{1-P(class(\boldsymbol{R}_i))}\sum_{j=1}^{k}\mathrm{diff}(\boldsymbol{A},\boldsymbol{R}_i,\boldsymbol{M}_j(C))\right]/(m\cdot k) \tag{11}$$

函数 diff=$(\boldsymbol{A},\boldsymbol{I}_1,\boldsymbol{I}_2)$计算样本 $\boldsymbol{I}_1$、$\boldsymbol{I}_2$ 在属性 $\boldsymbol{A}$ 上的差异值；$value(\boldsymbol{A},\boldsymbol{I}_i)$ 表示属性 $\boldsymbol{A}$ 在第 i 个样本上的值；max($\boldsymbol{A}$)和 min($\boldsymbol{A}$)分别表示属性 $\boldsymbol{A}$ 在所有样本中的最大值和最小值。对于离散属性，diff 的计算公式如下所示。

$$\mathrm{diff}(\boldsymbol{A},\boldsymbol{I}_1,\boldsymbol{I}_2)=\begin{cases}0; & value(\boldsymbol{A},\boldsymbol{I}_1)=value(\boldsymbol{A},\boldsymbol{I}_2)\\ 1; & \text{其他}\end{cases} \tag{12}$$

对于数值属性，diff 的计算公式如下所示。

$$\mathrm{diff}(\boldsymbol{A},\boldsymbol{I}_1,\boldsymbol{I}_2)=\frac{\left|value(\boldsymbol{A},\boldsymbol{I}_1)-value(\boldsymbol{A},\boldsymbol{I}_2)\right|}{\max(\boldsymbol{A})-\min(\boldsymbol{A})} \tag{13}$$

Chi-square 算法基于卡方检验统计量，计算公式如下所示。其中，$\boldsymbol{A}_i$ 表示属性 $\boldsymbol{A}$ 在第 i 个样本上的值，表示对应属性的期望观测值，通过统计得到。

$$\chi^2=\sum_{i=1}^{k}\frac{(\boldsymbol{A}_i-E)^2}{E} \tag{14}$$

基本原理是对每个特征计算其与期望观测值之间的卡方值。卡方值越小，表明特征值与期望值越接近，也就表明该特征有更好的分类能力；反之，卡方值越大，表明特征值与期望值之间差异越大，表明特征的分类能力更弱。Chi-square 算法要求数据必须是离散型的，因此 chi-square 算法使用的数据，是计算反向互补序列对在染色体序列和质粒序列上出现的频次数，而不是频率，这是 chi-square 算法与其他三种有巨大区别的地方之一。

Fisher 算法基于 Fisher 判别，计算每一个特征的 Fisher score，按照 score 的大小降序排序，score 越大表示该特征的分类能力越强，Fisher score 的计算公式如下所示：

$$F(x^j)=\frac{\sum_{k=1}^{c} n_k(\mu_k^j-\mu^j)^2}{(\sigma^j)^2} \quad (15)$$

其中，$(\sigma^j)^2=\sum_{k=1}^{c} n_k(\sigma_k^j)^2$；$\mu^j$ 和 σ^j 分别表示第 j 个特征在全局数据集上的平均值和标准差；μ_k^j 表示在第 k 类数据上第 j 个特征的平均值；n_k 表示第 k 类样本的样本数。

t-test 算法进行特征选择基于 t 检验，它计算每个特征在两个不同类别之间的 t 统计量差异，差异越大，表明该特征的分类能力越强，t 统计量计算公式如下所示：

$$t_j=\frac{\bar{\mu}_1-\bar{\mu}_2}{\sqrt{\frac{\sum_{i=1}^{n_1} x_1^{\,2}+\sum_{i=1}^{n_2} x_2^{\,2}}{n_1+n_2-2}\times\frac{n_1+n_2}{n_1\times n_2}}} \quad (16)$$

其中，$\bar{\mu}_1$ 和 $\bar{\mu}_2$ 分别表示第 j 个特征在第一类和第二类上的取值均值；n_1 和 n_2 分别表示类样本数；x_1 和 x_2 分别表示第 j 个特征在两类上的取值。

4 结 果

4.1 全部特征数据的分类性能

不进行特征选择时，运用 Logistic Regression、SVM（RBF kernel）、ELM 和 Random Forests 训练出的模型表现如表 1 所示。其中 Random Forests 算法的表现最好，AUC 指标可达到 0.95，表示有 95% 的数据可被正确分类。可见，实验采用的特征对染色体和质粒二者的分类是非常有效果的。

4.2 特征选择对随机森林算法分类性能的影响

Random Forests 算法的决策树数目设置为 500 棵，AUC 指标的结果如表 2 和图 1 所示。

表 1 四种不同算法使用全特征训练的模型结果

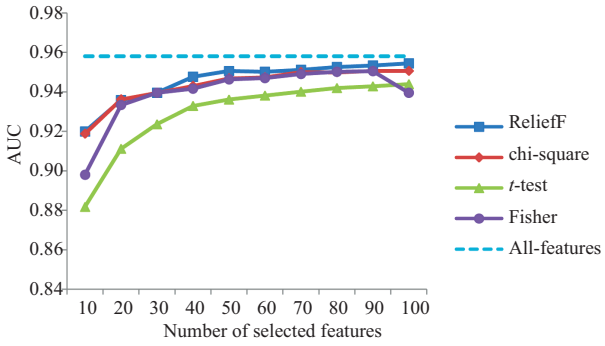
Table 1 Results of four classification models on all features

方法	算法	指标		
		Sensitivity	Specificity	AUC
十折交叉验证	ELM	0.778 61	0.784 56	0.851 25
	Logistic Regression	0.811 53	0.804 90	0.873 46
	SVM(RBF)	0.881 16	0.857 52	0.938 74
	Random Forest	0.926 62	0.861 77	0.958 09

表 2 Random Forests 算法验证四种特征选择算法的效果，以 AUC 指标为参考

Table 2 AUC results of Random Forests on the four feature selection algorithms

特征数量	特征选择算法			
	ReliefF	chi-square	*t*-test	Fisher
10	0.919 87	0.918 75	0.881 74	0.897 97
20	0.935 81	0.936 18	0.911 15	0.933 29
30	0.939 50	0.939 45	0.923 59	0.939 50
40	0.947 65	0.942 98	0.932 88	0.941 53
50	0.950 56	0.946 80	0.936 11	0.946 28
60	0.950 13	0.947 31	0.938 13	0.946 91
70	0.951 18	0.949 95	0.940 09	0.949 05
80	0.952 59	0.949 88	0.941 95	0.950 07
90	0.953 36	0.950 60	0.942 82	0.950 45
100	0.954 45	0.950 59	0.943 92	0.939 45



注：All-features 虚线表示在不进行特征选择时，Random Forests 算法的 AUC 表现

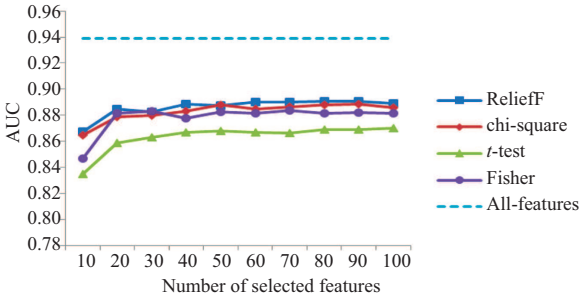
图 1　Random Forests 算法验证四种特征选择算法的效果，以 AUC 指标为参考

Fig. 1　AUC results of Random Forests on the four feature selection algorithms

ReliefF 算法的 AUC 表现最好，选取 ReliefF 排序的前 10 个特征即可正确分类 91% 以上的数据；chi-square 的 AUC 结果非常接近 ReliefF；*t*-test 和 Fisher 的结果略低于 ReliefF 和 chi-square，但都能正确分类 88% 以上的数据；Fisher 算法在 90 个特征时，AUC 达到 0.950 45，在 100 个特征时，AUC 出现下降。从表 2 中的数据可以看出，选取前 100 个特征，AUC 已经非常接近不做特征选择时的值，可见有大量的特征是冗余的，最优的特征组合是存在的。

4.3　特征选择对支持向量机算法分类性能的影响

SVM(RBF)算法的测试结果如表 3 和图 2 所示。从前 10 到前 100 个特征，SVM 算法的表现较 Random Forests 平缓，使用 ReliefF 算法选出的前 80 个特征时，AUC 达到最大值 0.890 68，前 90 个特征和前 100 个特征时，AUC 出现小幅下降。选取 chi-square 算法的前 50 个特征，AUC 值达到 0.887 80，之后 AUC 出现小幅下降，选取前 90 个特征时又达到最大值 0.888 43，前 100 个特征时出现下降。选取 Fisher 算法输出的前 30 个特征时，AUC 达到 0.883 11，之后 AUC 呈现小幅变化，在前 70 个特征时，达到峰值 0.883 64，与



注：All-features 虚线表示在不进行特征选择时，SVM 算法的 AUC 表现

图 2　SVM 算法验证四种特征选择算法的效果，以 AUC 指标为参考

Fig. 2　AUC results of SVM on the four feature selection algorithms

表 3　SVM 算法验证四种特征选择算法的效果，以 AUC 指标为参考

Table 3　AUC results of SVM on the four feature selection algorithms

特征数量	特征选择算法			
	ReliefF	chi-square	*t*-test	Fisher
10	0.867 11	0.864 42	0.834 57	0.846 63
20	0.884 43	0.878 73	0.858 73	0.881 30
30	0.882 43	0.879 60	0.862 77	0.883 11
40	0.888 77	0.883 31	0.866 67	0.877 92
50	0.887 55	0.887 80	0.868 13	0.882 58
60	0.890 05	0.884 97	0.867 02	0.881 69
70	0.889 94	0.886 23	0.866 40	0.883 64
80	0.890 68	0.887 73	0.869 05	0.881 49
90	0.890 42	0.888 43	0.868 84	0.881 81
100	0.889 22	0.885 79	0.869 87	0.881 59

前 30 个特征的效果相差不大。相比之下，*t*-test 的结果比其他三种算法都低 2% 左右。

可见分类算法本身对特征选择模型的影响也比较大，SVM 算法和 Random Forests 算法处理特征的方式非常不同，SVM 算法是根据核函数将特征映射到更高维的空间寻找分离超平面，而 Random Forests 则是通过生成众多决策树来决定模型的分类结果，因此二者的结果会出现比较大的差异。

4.4 特征选择对 ELM 算法分类性能的影响

经过效果对比，选取最佳的参数设置，ELM 算法的隐含层单元数设置为 500，激励函数设置为 sigmoid，测试结果如表 4 和图 3 所示。分别选取 ReliefF、chi-square 和 Fisher 的前 20 个特征，分类效果已经超过全部特征。选取 chi-square 算法的前 50 个特征，AUC 值达到最高，其他三种算法的表现略差于 chi-square，但差异都比较小。

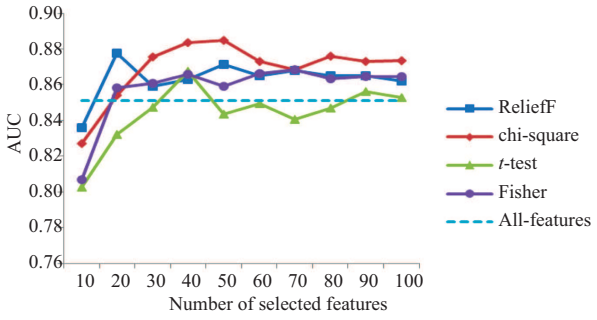
4.5 特征选择对逻辑回归算法分类性能的影响

Logistic Regression 算法的测试结果如图 4 和表 5 所示。相比以上三种算法，Logistic

表 4 ELM 算法验证四种特征选择算法的效果，以 AUC 指标为参考

Table 4 AUC results of ELM on the four feature selection algorithms

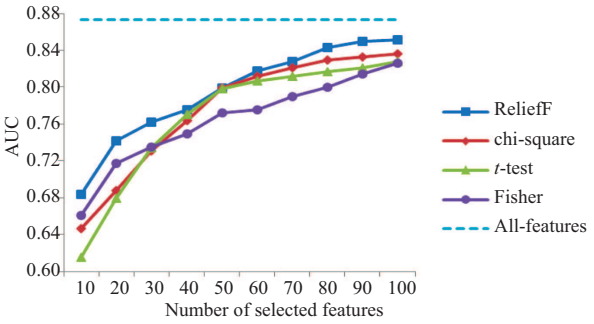
特征数量	特征选择算法			
	ReliefF	chi-square	*t*-test	Fisher
10	0.835 78	0.827 30	0.802 48	0.806 85
20	0.877 67	0.854 30	0.832 35	0.858 49
30	0.859 08	0.875 67	0.847 55	0.861 01
40	0.862 73	0.883 77	0.867 45	0.866 06
50	0.871 54	0.885 03	0.843 53	0.859 25
60	0.864 93	0.873 18	0.849 66	0.866 22
70	0.868 06	0.868 48	0.840 56	0.868 29
80	0.864 99	0.876 13	0.846 75	0.863 55
90	0.865 21	0.873 25	0.856 37	0.864 59
100	0.861 99	0.873 66	0.852 82	0.864 43



注：All-features 虚线表示在不进行特征选择时，ELM 算法的 AUC 表现

图 3 ELM 算法验证四种特征选择算法的效果，以 AUC 指标为参考

Fig. 3 AUC results of ELM on the four feature selection algorithms



注：All-features 虚线表示在不进行特征选择时，Logistic Regression 算法的 AUC 表现

图 4 Logistic Regression 算法验证四种特征选择算法的效果，以 AUC 指标为参考

Fig. 4 AUC results of Logistic Regression on the four feature selection algorithms

表 5　Logistic Regression 算法验证四种特征选择算法的效果，以 AUC 指标为参考

Table 5　AUC results of Logistic Regression on the four feature selection algorithms

特征数量	特征选择算法			
	ReliefF	chi-square	*t*-test	Fisher
10	0.683 34	0.646 81	0.614 94	0.660 44
20	0.742 14	0.687 69	0.679 76	0.717 56
30	0.762 25	0.730 62	0.734 50	0.735 04
40	0.775 47	0.763 96	0.770 87	0.749 51
50	0.799 40	0.799 22	0.798 40	0.772 56
60	0.817 75	0.811 83	0.806 92	0.775 30
70	0.828 12	0.820 81	0.811 70	0.789 92
80	0.842 77	0.829 91	0.816 79	0.799 93
90	0.849 82	0.833 11	0.821 13	0.814 45
100	0.851 24	0.836 21	0.827 46	0.826 36

Regression 在各个特征集合上的表现略差，选取 ReliefF 算法的前 100 个特征时，AUC 接近全特征时的 0.87。Logistic Regression 是运用多项式回归来拟合数据，理论上来说是特征越多越好，过少的特征(远小于样本数)会出现欠拟合，但特征数越多，学习的参数越多，容易出现过拟合。

4.6　综合性能比较

从实验结果来看，Random Forests 算法的表现最好，SVM、ELM 和 Logistic Regression 的结果稍次之。在特征选择方法上，ReliefF，chi-square 的表现最好，选取前 100 个特征训练的模型效果非常接近全特征模型的效果，Fisher 和 *t*-test 的结果稍次之。为观察 ReliefF、Fisher、chi-square、*t*-test 四种算法对特征进行的重要性排序，绘制图 5，图中从上至下的每一条横线代表一个特征，灰色的横线代表四种算法各自选出的前 100 个特征集合。例如，ReliefF 算法选出的前 100 个特征当中有 $f_2, f_3, f_5, \cdots, f_{127}, \cdots, f_{687}$，则第 2 条、第 3 条、第 5 条、…、第 687 条特征线分别标记成灰色，不可见的白色横线表示 100 名以后的特征，不在寻找最优特征组合的考虑范围内，因此标为白色。从图 5 可以看出，四种特征选择方法对特征的重要性排序方面有所不同，

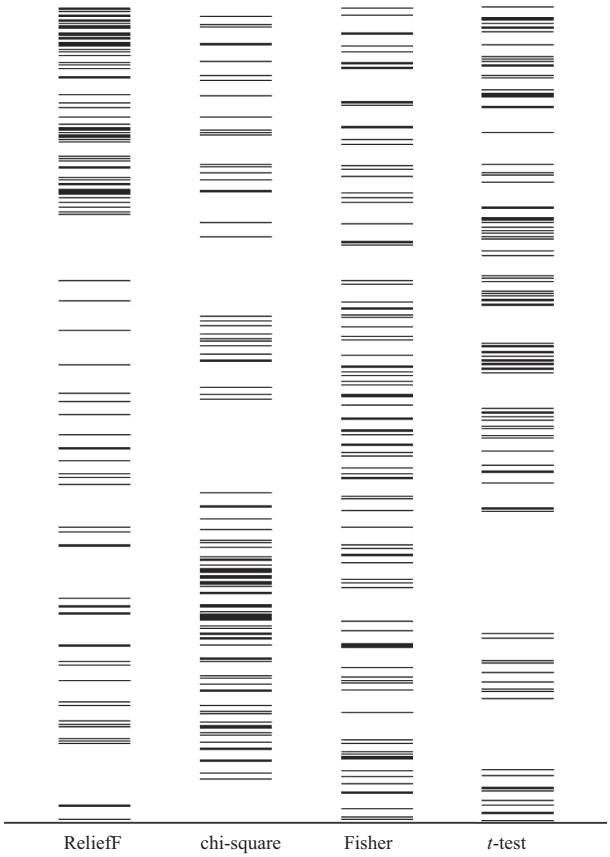


图 5　四种算法分别选出的 100 个特征组合(灰色横线表示被选中的特征)

Fig. 5　Top 100 features the four algorithms selected

ReliefF 认为重要的特征基本集中在靠前的特征，chi-square 认为重要的特征基本集中在靠后特征，Fisher 认为重要的特征分布比较分散，*t*-test 认为重要的特征也较为分散地集中在靠前的特征当中，但四种算法的特征选择组合在模型分类性能上相差不大。

为直观地理解所有反向互补序列对在染色体和质粒上的频率分布，做图如图 6 所示。为放大差异便于可视化，图中纵坐标表示频率值（×100），横坐标表示第 1 个到第 690 个特征，坐标中的点表示每个特征在染色体样本上或质粒样本上的频率平均值（×100）。从图中可以看出，所有的反向互补序列对在染色体和质粒上出现的频率并无显著差异，频率曲线非常相似。为进一步观察两者之间的特征差异，绘制两者平方之差的曲线，如图 7 所示。从图可以看出，两者的平方值在某些特征上呈现显著差异，集中在前 25 个。图 8 显示了平方差绝对值大于 1 的 26 个特征，我们选取这 26个特征进行模型训练，数据特征变换为原特征的平方，结果如表 6 所示。Random Forest 的表现最好，AUC 值超过 0.94，略低于 ReliefF 算法选出的前 100 个特征。

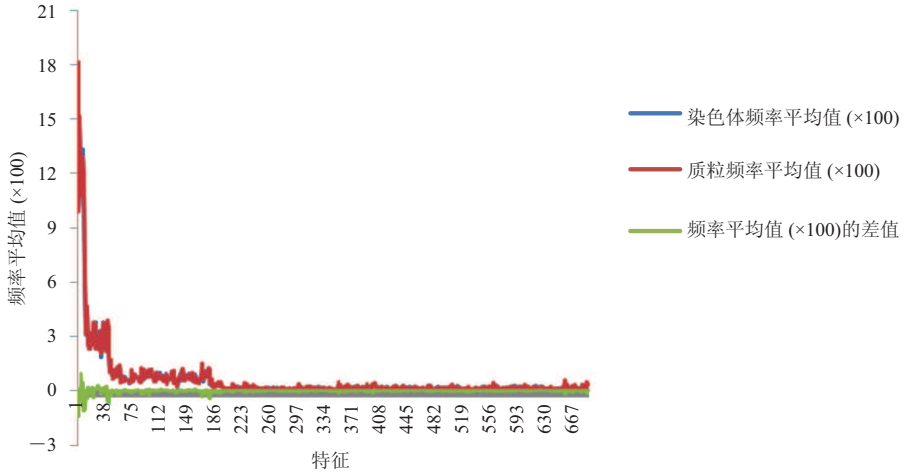


图 6 所有特征在染色体和质粒上的频率平均值（×100）差值分布

Fig. 6 Distribution of 100-fold frequency difference between chromosome and plasmid

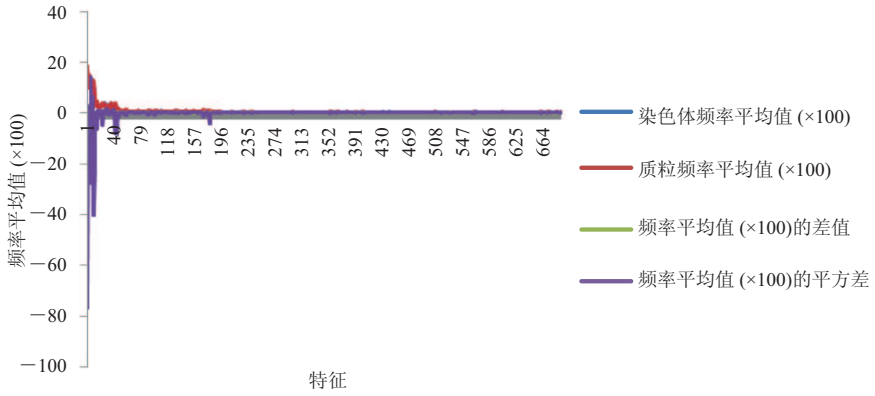


图 7 所有特征在染色体和质粒上的频率平均值（×100）差值和平方差值分布

Fig. 7 Distribution of squared 100-fold frequency difference between chromosome and plasmid

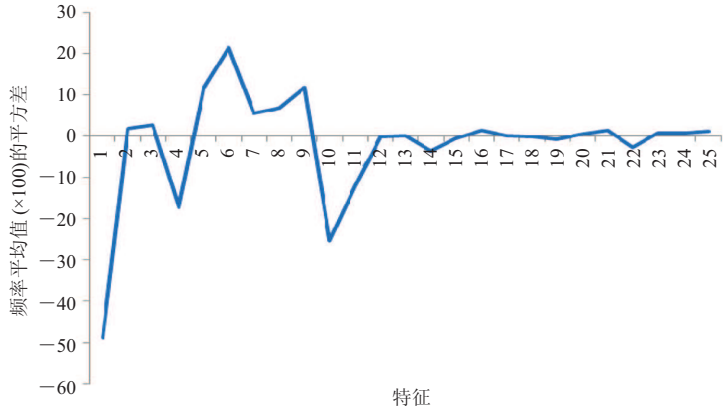


图 8　频率平均值(×100)平方差的绝对值大于 1 的前 26 个特征

Fig. 8　Top 26 features which absolute value of squared 100-fold frequency difference is above 1

表 6　频率平均值(×100)平方差的绝对值大于 1 的前 26 个特征用四种算法训练的模型效果

Table 6　Results of four classification models on the top 26 features which absolute value of squared frequency is above 1

方法	算法	指标		
		Sensitivity	Specificity	AUC
十折交叉验证	Logistic Regression	0.689 675	0.697 916	0.765 526
	ELM	0.714 204	0.752 665	0.804 623
	SVM(RBF)	0.822 806	0.795 442	0.890 024
	Random Forest	0.895 540	0.842 865	0.943 174

5　结　论

综合以上的讨论，选取 ReliefF 算法选出的前 100 个特征，运用 Random Forests 算法进行分类模型训练，可得到与选用全部特征时最接近的效果。本实验的实验结果与 cBar[1]的结果对比如表 7 所示。从表中可以看出，在 cBar 采用全部的 512 个“五字符短串”的频率特征得到的分类结果中，SVM 的表现最优，神经网络、贝叶斯和决策树(C4.5)的效果略差。本实验综合考虑 k 介于 2～5(包含)之间的所有“k 字符短串”的频率特征，并且引入特征选择算法。结果表明，Random Forests 算法采用 ReliefF 算法选择的前 100 个特征时，AUC 可达到所有模型中的最优值 0.954 5；SVM(RBF kernel)算法采用 ReliefF 算法选择的前 80 个特征时，AUC 最优可达到 0.890 7；ELM 算法采用 chi-square 算法选出的前 80 个特征，AUC 最优可达到 0.876 1；Logistic Regression 采用 ReliefF 算法选择的前 100 个特征，AUC 最优可达到 0.851 2。可见，实验采用的特征在远少于 cBar 的情况下，结果普遍优于 cBar。

表 7　本实验结果与 cBar 的结果对比

Table 7　Our experiment AUC result compared to cBar

模型	算法	AUC
cBar	cBar-SVM	0.915 0
	CBar-NN	0.864 0
	CBar-Bayes net	0.815 0
	cBar-C4.5	0.841 0
本实验	Random Forest-ReliefF(100 features)	0.954 5
	SVM-ReliefF(80 features)	0.890 7
	ELM-chi-square(80 feateres)	0.876 1
	Logistic Regression-ReliefF(100 features)	0.851 2

同时也可以看到，不同的特征选择算法选

出的特征组合，用不同的分类算法，结果会不一样，这跟分类算法处理特征的方法有关。Random Forests 本身具有特征选择的功能，特征选择之后再运用 Random Forests 算法，相当于进行了两次特征选择，因此分类性能比较突出。SVM（RBF kernel）算法、ELM 算法、Logistic Regression 算法本身没有再进行特征选择的处理，只是纯粹地进行分类，因此分类性能比 Random Forests 算法略差。

运用过滤式方法进行特征选择，不同算法对特征的重要性排序有较大的差异，运用不同的分类算法训练出的模型结果也有较大差异。从实验数据可以看出，在最优的 100 个特征以内，选取越多的特征，分类效果不一定越好，如 ELM 算法。排名靠后的特征不一定分类能力就比排名靠前的特征差，只是独立地考虑每个特征的分类能力，而忽略特征之间的一些关系，是过滤式方法最大的缺点之一，这是本文后续研究中需要不断探索的方向。

致谢

感谢中科院战略先导项目（XDB13040400）、深圳市孔雀计划项目（KQCX20130628112914301, KQCX20130628112914291）、国家“863”计划（SS2015AA020109-4）以及深圳市研究资助项目（JCYJ20130401114111457, JCYJ20130401170306884）的支持。感谢中国科学院深圳先进技术研究院生物医学信息技术重点实验室对本工作的支持，本研究的部分计算资源由中国科学院曙光超级计算集群支持。

参 考 文 献

[1] Zhou FF, Xu Y. cBar: a computer program to distinguish plasmid-derived from chromosome-derived sequence fragments in metagenomics data [J]. Bioinformatics, 2010, 26(16): 2051-2052.
[2] Robin S, Schbath S. Numerical comparison of several approximations of the word count distribution in random sequences [J]. Journal of Computatioal Biology, 2001, 8(4): 349-359.
[3] Reinert G, Schbath S, Waterman MS. Probabilistic and statistical properties of words: an overview [J]. Journal of Computational Biology, 2000, 7(1-2): 1-46.
[4] Chor B, Hom D, Goldman, et al. Genomic DNA k-mer spectra: models and modalities [J]. Genome Biology, 2009, 10(10): 571.
[5] Kurtz S, Narechania A, Stein JC, et al. A new method to compute k-mer frequencies and its application to annotate large repetitive plant genomes [J]. Bmc Genomics, 2008, 9(1): 517.
[6] Macas J, Neumann P, Novak P, et al. Global sequence characterization of rice centromeric satellite based on oligomer frequency analysis in large-scale sequencing data [J]. Bioinformatics, 2010, 26(17): 2101-2108.
[7] Pevzner PA, Tang H, Waterman MS. An eulerian path approach to DNA fragment assembly [J]. Proceedings of the National Academy of Sciences of the United States of America, 2001, 98(17): 9748-9753.
[8] Quinlan JR. Bagging, boosting, and C4.5 [C] // Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference, 1996: 725-730.
[9] Cortes C, Vapnik V. Support-vector networks [J]. Machine Learning, 1995, 20(3): 273-297.
[10] Cessie SL, Houwelingen JCV. Ridge estimators in logistic-regression [J]. Applied Statistics, 1992, 41(1): 191-201.
[11] Werbos PJ. Generalization of backpropagation with application to a recurrent gas market model [J]. Neural Networks, 1988, 1(88): 339-356.
[12] Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and applications [J]. Neurocomputing, 2006, 70(1-3): 489-501.
[13] Breiman L. Random forests [J]. Machine Learning, 2001, 45(1): 5-32.
[14] Zhou F, Olman V, Xu Y. Barcodes for genomes and applications [J]. Bmc Bioinformatics, 2008, 9(29): 1-11.
[15] 李航. 统计学方法 [M]. 清华大学出社, 2012.
[16] Huang GB, Zhu QY, Siew CK. Extreme learning machine: a new learning scheme of feedforward neural networks [C] // Proceedings of IEEE International Joint Conference on Neural Networks, 2004: 985-990.
[17] 姚旭, 王晓丹, 张玉玺, 等. 特征选择方法综述 [J]. 控制与决策, 2012, 27(2): 161-166.
[18] Guyon I, Elisseeff A. An introduction to variable and feature selection [J]. Journal of Machine Learning Research, 2003, 3: 1157-1182.
[19] Kononenko I, RobnikSikonja M, Pompe SU. ReliefF for estimation and discretization of attributes in classification, regression, and ILP problems [J]. Artificial Intelligence: Methodology, Systems, Applications, 1996, 35: 31-40.
[20] Kira K, Rendell LA. The feature selection problem: traditional methods and a new algorithm [C] // Tenth National Conference on Artificial Intelligence, 1992: 129-134.
[21] Robnik-Sikonja M, Kononenko I. Theoretical and empirical analysis of ReliefF and RReliefF [J]. Machine Learning, 2003, 53(1-2): 23-69.