MLFSdel: An accurate approach to discover genome deletions

Yao Zhang[a], Jingyang Gao[b*]

College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, 100029, China

[a]email: 872270928@qq.com, [b*]email:gaojy@mail.buct.edu.cn

Keywords: Sequence Analysis; Feature; Model;

Abstract. Genome deletions are one of the common types of structural variations. The discovery of deletions has become an important research field in SVs detection of genome sequences. At present, the existing methods have their own limitations, and these methods are also insufficient in precision and sensitivity. Hence, improving the detecting efficiency has become a critical target in subsequent research. In this paper, we developed a method, namely MLFSdel. Essentially, MLFSdel employs four machine learning models and implements a novel feature selection strategy. By eliminating the features having the negative effect on the overall classification results, the proposed method improves the precision and sensitivity in comparison to four previous methods for detecting deletions. In addition, it further proves that the feature-based machine learning methods are applicable to detect genome deletions.

Introduction

Genomic structural variants (SVs) are a major contributor to common diseases, which are the cause of many rare genomic disorders. SVs of DNA segments include gains, losses and rearrangements in comparison to the reference genome. Most methods detecting SVs from sequence reads harness one or several of the following theories: (i) paired-end reads [1]: the paired-end reads with known insert size are mapped to reference genome, then determine whether there are in consistencies of insert sizes. (ii) split-mapped reads [2-3]: split-mapped reads span the breakpoints of the deletions and their segments are mapped to different positions of the reference, thus split-mapped reads may indicate the existence of deletions. (iii) read-depth [4]: normal read depth-of-coverage tends to imply deletion or duplication. (iv) Local assembly [5]: mapping an assembled contig around potential variation may detect SVs.

Each existing method has its strength and weakness. For example, DELLY [6] works better on small indels, while Clever-SV [7] performs better on middle size deletions. Hence, the results from these methods are insufficient in detecting deletions of all sizes. To conquer, a recent research proposes a feature-based machine learning method, namely Concod [8]. Concod is the recent research result by our group. When a deletion occurs in a region, various signatures of deletions may be found from the reads. Concod extract features from the read-pair, split-read and read-depth signatures as introduced above. The features are refined according to with concordant and discordant pair-end reads, fully and clipped mapped reads, read depth and Sequence reads mapping respectively. And the last feature is deletion length. Therefore, Concod exact 49 features in total. Concod employs SVM and gives better result in detecting deletions. However, Concod still needs to run multiple methods, and its feature process needs further study.

Machine learning methods can be faster and automatically generating models to analyze larger complex data, and give out more accurate results. A number of machine learning theory have been proposed, for example, k-nearest neighbor classification, Classification and Regression Trees, Gradient Boosting Decision Tree, and Random Forest.

In this paper, we develop a deletion detection method, called MLFSdel, which generates models by training data for detecting deletions. Moreover, MLFSdel implements feature selection, which can be used to exclude redundant and irrelevant features and achieve higher accuracy. In addition, the prior Concod method fixes the input feature number, i.e. 49， the input of MLFSdel is any

Copyright © 2017, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

numbers of features. Hence, our method can deal with a wider range of feature input. MLFSdel performs better in detecting deletions of all sizes and all frequencies, and improves the precision and sensitivity in comparison to another four famous methods for detecting the deletions. MLFSdel has some improvement than Concod in feature research.

METHOD

The implementation architecture of the proposed method MLFSdel has four parts. At first, MLFSdel inputs feature datasets of candidate deletions. Secondly, MLFSdel chooses whether to conduct feature selection on the original feature datasets. Our method uses selections: custom select the features by a remaining feature number provided by users or automatic select the best features by the feature selection process. Thirdly, a classification model is established by training these selected features. Finally, MLFSdel uses the model to classify all the candidate deletions.

The input of MLFSdel

The input of MLFSdel is features which need to be standardized. The process of extracting features is based on the three theories, which are paired-end reads, split-mapped reads and read-depth. The detailed process of feature exaction is described in Concod. Feature datasets are consist of classification labels and all values of features. Left breakpoint and right breakpoint are treated separately in Concod. Several experiments show the breakpoint separation has no influence to filter out false positive deletions. The separation is not considered in our research. Therefore, we exact 38 features in total for the next experiments.

Feature selection

To express characteristic of deletions explicitly, we intend to extract all the features from the signatures. However, more features may cause other problems. The existence of irrelevant and redundant features may reduce the classification performance. Hence, we need to use the features, which are beneficial for the classification, and prevent over-fitting occurrence.

To eliminate appearances of irrelevant and redundant features, some measures are used to determine the importance of each feature. In the paper, we employ information gain. After comparing values of the measure for all features, we choose different combinations of features to compose the optimal or customized features subset. Information gain also determines whether a feature is useful for the classification.

Information gain

Information gain can estimate the amounts of information, which are brought from introduction of new feature. Information gain is an important indicator in the feature selection. It can be defined as a feature how much information brought for classification system. The more information bring, the more important the feature will be. For a feature T, which value range is ($t_1, t_2, \ldots, t_n$), its information gain is IG(T)=H(C)-H(C|T). C is a classification result, the value of which is either c_0 or c_1. H(C |T) has two situations: one is the appearance of feature T, signed as t; another is no appearance of feature T, signed as t'. Formulas of H(C) is as follows. Figure 1 shows one example of the information gains for calling deletions on real sequence data.

$$H(C) = P(c_0)\log_2(P(c_0)) + P(c_1)\log_2(P(c_1)) \quad (1)$$

The basis for feature selection is associating information gain. According to the values of metric, the priority values are generated for all features.

Obtaining the optimal subset

We classify different features to form subsets of different numbers of features by using the priority values of features. Then these feature subsets are used to generate the optimal feature subset. We take an example for calling deletions on real sequence data. We show an example for calling deletions on real sequence data in Figure 2.

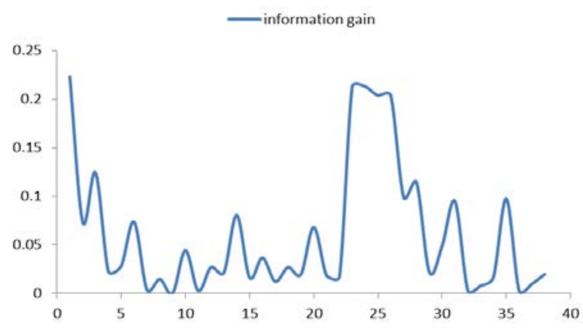


Fig.1.Information gain of all features

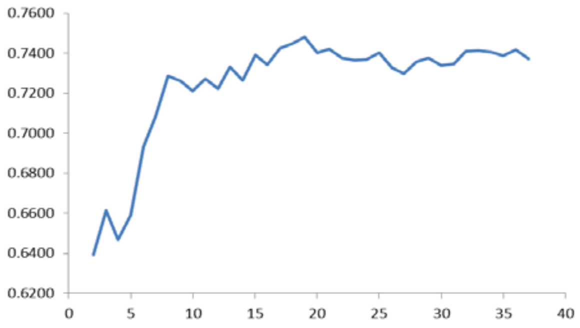


Fig.2. Performance of different feature combinations

RESULT

Data preparation

A total number of 47 individuals from the third phase of the 1000 Genomes Project are adopted. The latest structural variations [9] released by the 1000 Genomes Project are treated asthe benchmark data. These structural variations have been validated by the 1000 Genomes Project. 34706 candidate deletions are obtained after using Pindel, SVseq2, DELLY and BreakDancer to detect these individuals. For the candidate deletion set, 39 individuals are selected as training data, the remaining individuals (8) are chosen as the testing data. 38 features are exacted from the candidate deletions. These feature datasets are labeled to obtain experimental features according to the benchmark data. We use feature selection on the experimental feature datasets to obtain optimal features (19 features). The optimal feature datasets are used as the input of MLFSdel.

Table I. Performance comparisons of the tools.

Tool		Call set	True Positive	Precision (%)	Sensitivity (%)
Pindel		1564	705	44.87	52.79
SVseq2		1736	764	46.58	57.51
BreakDancer		1910	665	35.90	50.71
DELLY		1736	764	46.58	57.51
MLFSdel without selection	CART	2573	1824	70.89	60.20
	GBDT	2721	2176	79.97	64.65
	RF	2606	2080	79.82	59.97
	KNN	2769	1969	71.11	59.21
MLFSdel with selection	CART	2601	1902	73.51	60.55
	GBDT	2666	2174	81.55	65.33
	RF	2572	2101	81.69	60.88
	KNN	2494	1903	76.30	59.67

Comparisons between MLFSdel and these tools

First, the candidate deletions of the real datasets (47 individuals) are detected by employing BreakDancer, Pindel, SVseq2 and DELLY. Table I presents the detection sensitivity and precision of these four tools, which are roughly between 0.4 and 0.5. In the table, *Call set* is the number of deletions detected by each tool, true positive means the number of deletions that are truly detected.

Our approach depends on four machine learning methods, which are CART, KNN, RF, and GBDT. We input the optimal feature datasets to MLFSdel. MLFSdel employs these four machine learning methods to establish four different learning models, respectively. These learning models detect the testing datasets and produce the classification results. As shown in table I, the results of four machine learning models outperform SVseq2, BreakDancer, DELLY and Pindel. The experimental results suggest that the feature-based machine learning methods are applicable to detect deletions.

To validate the efficiency of the proposed feature selection strategy, we compare two results of

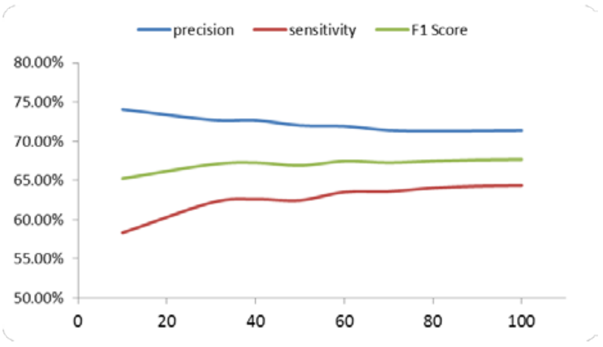
MLFSdel by carrying out feature selection or not. Table I further lists the performance of four machine learning models without feature selection and the performance of four machine learning models with feature selection. All the results of MLFSdel on four models are accurate with feature selection.

1) Adjusting parameters of models

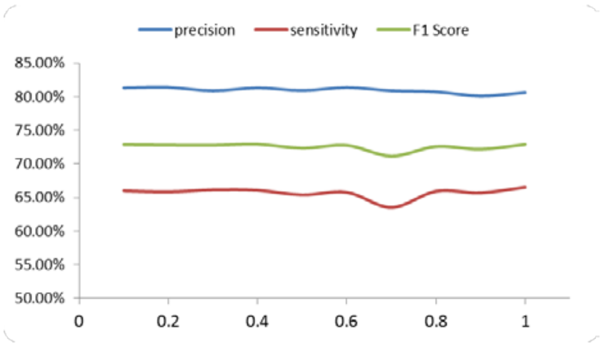
Due to four learning models having many different parameters, these parameters are adjusted based on the models. According to each parameter, we summary and analyze detection results of KNN, GBDT, CART and RF.

KNN needs a parameter n_neighbors for unclassified sample in KNN model. GBDT is an iterative decision tree. The algorithm consists of multiple trees, which adds up results of all the trees to the final one. Learning rate is the major parameter in the algorithm. CART is a binary tree. The tree requires an important parameter, namely *max_features*. For RF, the tree numbers are the essential parameter as the algorithm employs multiple trees to train samples and predict results.

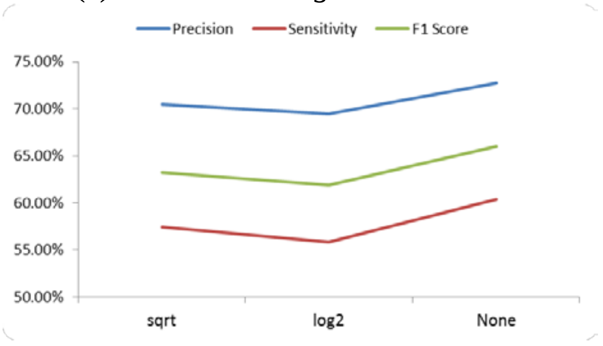
Figure 2 shows that as the parameters changing, the precision and sensitivity of the four models are relatively stable. Hence, the outperformance achieved in the proposed method is not randomly occurred.



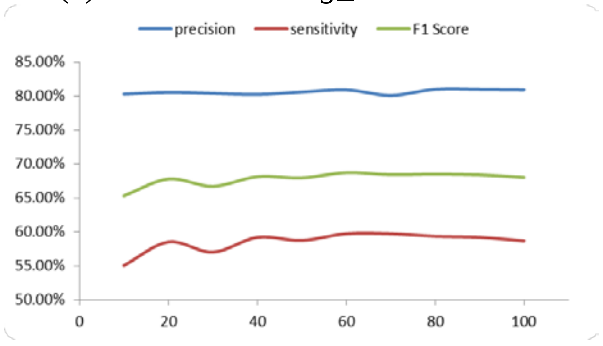
(a) Different *n_neighbors* for KNN



(b) Different *learning_rate* for GBDT



(c) Different *max_features* for CART



(d) Different *n_estimators* for RF

Fig.3. The result of adjusting parameters of models

Conclusion

In this paper, we presented our deletions discovery approach MLFSdel that aims to improve both the precision and sensitivity of deletions detection. We test MLFSdel on genome sequence data from 1000 Genome Project. The four machine learning models are employed by MLFSdel to detect deletions, the result of each model is better than the four methods. It also proves that machine learning approaches are effective ways to distinguish true and false deletions. Besides, MLFSdel implements feature selection. For the original feature datasets, excluding redundant and irrelevant features gives out better feature datasets, which can further improve the precision and sensitivity in the original basis. Overall, MLFSdel outperforms Pindel, SVseq2, BreakDancer and DELLY.

In addition, MLFSdel not only performs better on the feature datasets (38 features) which are adapted by our experiments, but also easily adapts other feature datasets (arbitrary number of feature) in applications. MLFSdel can be extended to call other types of SVs.

Acknowledgement

In this paper, the research was sponsored by the National Natural Science Foundation of China (Project No. 61472026).

References

[1] Chen K, Wallis J W, McLellan M D, and et al, "BreakDancer: an algorithm for high-resolution mapjping of genomic structural variation," Nature methods, 2009, vol. 6(9),pp. 677-681.

[2] Ye K, Schulz M H, Long Q, and et al, "Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads,"Bioinformatics, 2009, vol. 25(21), pp. 2865-2871.

[3] Zhang J, Wang J, and Wu Y, "An improved approach for accurate and efficient calling of structural variations with low-coverage sequence data," BMC bioinformatics, 2012, vol. 13(6),pp. 1.

[4] Abyzov A, Urban A E, Snyder M, and et al, "CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing," Genome research, 2011, vol. 21(6), pp. 974-984.

[5] Narzisi G, O'Rawe J A, Iossifov I, and et al, "Accurate de novo and transmitted indel detection in exome-capture data using microassembly," Nature methods, 2014, vol. 11(10),pp. 1033-1036.

[6] Rausch T, Zichner T, Schlattl A, and et al, "DELLY: structural variant discovery by integrated paired-end and split-read analysis," Bioinformatics, 2012, vol. 28(18), pp. i333-i339.

[7] Marschall T, Costa I G, Canzar S, and et al, "CLEVER: clique-enumerating variant finder," Bioinformatics, 2012, vol. 28(22), pp. 2875-2882.

[8] Zhang X, Chu C, Zhang Y, and et al, "Concod: Accurate Consensus-based Approach of Calling Deletions from High-throughput Sequencing Data," BIBM 2016: 72-77

[9] ALL.wgs.integrated_sv_map_v2.20130502.svs.genotypes.vcf