BIOINFORMATICS APPLICATIONS NOTE

Vol. 24 no. 23 2008, pages 2773–2775
doi:10.1093/bioinformatics/btn517

Genome analysis

CNVDetector: locating copy number variations using array CGH data

Peng-An Chen[1], Hsiao-Fei Liu[1] and Kun-Mao Chao[1,2,3,*]

[1]Department of Computer Science and Information Engineering, [2]Graduate Institute of Biomedical Electronics and Bioinformatics and [3]Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan

Received on March 12, 2008; revised on September 4, 2008; accepted on October 4, 2008

Advance Access publication October 7, 2008

Associate Editor: Alex Bateman

ABSTRACT

Summary: CNVDetector is a program for locating copy number variations (CNVs) in a single genome. CNVDetector has several merits: (i) it can deal with the array comparative genomic hybridization data even if the noise is not normally distributed; (ii) it has a linear time kernel; (iii) its parameters can be easily selected; (iv) it evaluates the statistical significance for each CNV calling.

Availability: CNVDetector (for Windows platform) can be downloaded from http://www.csie.ntu.edu.tw/~kmchao/tools/CNVDetector/. The manual of CNVDetector is also available.

Contact: kmchao@csie.ntu.edu.tw

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Copy number variations (CNVs) of a target genome are DNA segments which are longer than 1 kb and present at different copy number in comparison with the reference genome (Redon *et al*., 2006). CNVs can be located by the high-resolution array comparative genomic hybridization (array CGH) approaches (Pinkel *et al*., 1998). Array CGH approaches are microarray-based techniques which provide fluorescence intensity ratios at arrayed small DNA samples. The higher fluorescence intensity ratio indicates that the target genome contains more copies of the corresponding DNA sample than the reference genome does. Given the array CGH data, many methods and programs for finding CNVs in a single target genome have been proposed (Hupé *et al*., 2004; Olshen and Venkatraman, 2004; Picard *et al*., 2005). Comparison studies of these CNV detection programs can be found in Lai *et al*. (2005) and Willenbrock and Fridlyand (2005). As indicated in Lai *et al*. (2005), however, these programs more or less suffer from (i) high-time complexity of computing the results, (ii) sensitive outputs to parameters; and (iii) high time complexity of selecting the parameters.

Recently, Lipson *et al*. (2006) proposed a new framework for identifying CNVs in a single target genome by assuming that the measurement noise along the chromosome is independent for distinct probes and normally distributed. One advantage of this framework is that it provides the statistical significance for each CNV calling and these values are useful for further analysis. Also, this framework requires less computation time. Lipson *et al*. (2006) proposed an $O(n^2 \cdot k)$-time program for this framework and showed the program runs in $O(n^{1.5} \cdot k)$ time in practice, where n is the input size and k is the number of detected CNVs. However, there is no evidence supporting that the noise of the array CGH data is always normally distributed.

In this work, we enhance Lipson *et al*.'s framework so that it can work well even if the noise is not normally distributed. Besides, we implement an efficient $O(n \cdot k)$-time program, CNVDector, for this enhanced framework based on algorithms of Bernholt *et al*. (2007). Since the resolution of array CGH approaches is increasing, the efficiency of CNVDetector is desirable. CNVDetector is able to find CNVs in a single genome and these CNV callings can be further analyzed by the common CNV detecting frameworks which deal with CNV calling data from multiple samples. We demonstrate that our program can be used to find CNVs in acute myeloid leukemia (AML) samples.

2 PRELIMINARIES

CNVs are caused by either amplification events (duplications of DNA segments) or deletion events (deletions of DNA segments). The amplification event of one DNA segment suggests that the target DNA genome contains more copies of the corresponding DNA segment than the reference genome does. The deletion event of one DNA segment suggests that the target DNA genome contains less copies of the corresponding DNA segment than the reference genome does. Our goal is to locate the aberrant regions which present adjacent gains (amplification events) or losses (deletion events) in the target genome.

Array CGH approaches are based on microarrays, which consist of a number of probes and each probe contains a small DNA fragment. By measuring the fluorescence intensity at each probe, the array CGH approaches can provide a vector $V = (v_1, v_2, ..., v_n)$, where v_i is the log ratio of the fluorescence intensity in the target genome to the fluorescence intensity in the reference genome for the i-th probe.

Given the vector V, Lipson *et al*. (2006) proposed a statistical model for asserting aberrant regions. Assume that the noise in the array CGH data is independent for distinct probes. Note, here we do not assume that the noise in array CGH data is normally distributed. Let μ and σ be the mean and SD of the normal genomic data. Let, the null hypothesis be that there are no events present in the target

[*]To whom correspondence should be addressed.

DNA sequence. Given a region I, define $\varphi^{sig}(I)$ by:

$$\varphi^{sig}(I)=\sum_{i\in I}\frac{v_i-\mu}{\sigma\sqrt{|I|}},$$

where $|I|$ is the number of probes in I. In general, if $|I|$ is larger than 25 (Hogg and Tanis, 2006), the distribution of $\varphi^{sig}(I)$ will be approximately a normal distribution, where we have

$$Prob(|\varphi^{sig}(I)|>z)\approx\frac{1}{\sqrt{2\pi}}\cdot\frac{1}{z}e^{-\frac{1}{2}z^2}.$$

Therefore, $\varphi^{sig}(I)$ is used to assess the statistical significance of region I provided that $|I|\geq 25$.

3 ALGORITHM

First, we normalize the vector V such that the mean of V is equal to zero after the normalization. By the statistical model, the score function of the region I is defined by $f(I)=|\sum_{i\in I}v_i|/\sigma\sqrt{|I|}$. In addition, the number of probes in the located regions must be no less than a length lower bound L. Furthermore, if we want to locate CNVs with length upper limit, the number of probes in the located regions is set to be no greater than a length upper bound U. The aberrant region is located by finding the region I_{max} which maximizes the function f subject to $L\leq|I_{max}|\leq U$.

As the function f is quasiconvex, we can use the $O(n)$-time algorithm presented by Bernholt *et al.* (2007) to find I_{max}, where n is the number of probes in V. After the max-score region, I_{max} is reported; we remove I_{max} and apply the algorithm to the remaining parts of the vector V recursively until there are no regions with score larger than a given threshold value ϕ_{th}. Since, each detection of one CNV costs the algorithm $O(n)$ time, the total runtime is $O(n\cdot k)$, where k is the number of detected CNVs.

4 PERFORMANCE EVALUATION

We benchmark the performance of CNVDetector by finding the max-score regions on five datasets. Datasets 1, 2, 3 and 4 are synthetic and are generated as follows. First, the vector V is generated by drawing n numbers from a uniform distribution with the range $[-2,0]$. Second, insert an amplification interval into the vector V. The length of the amplification interval is equal to L_I+10, where L_I is randomly drawn from a geometric distribution with success probability 0.01. The value of each element in the amplification interval is randomly drawn from a uniform distribution with the range $[10,50]$. Dataset 5 is comprised of the analysis result of AML cell lines taken from patent no. 43 (Yamashita *et al.*, 2007). It contains 38 246 probes from chromosome 1 to chromosome 22, providing an average spacing of 35 kb between each consecutive pair of probes. The benchmark is performed on Intel Xeon 3.2G with Linux 2.6.22 and the benchmark results of these five datasets are listed in Table 1. We also compare the performance of CNVDetector with that of four well-known CNV detection tools. See, the Supplementary Material for details.

Table 1. Benchmark results

Dataset No.	Number of probes in the vector	Time (s)
1	5000	0.116
2	10 000	0.236
3	50 000	1.184
4	1 00 000	2.448
5	38 246	0.944

5 RESULTS

In Dataset 5, we set ϕ_{th} to 2.0 and the length lower bound L to 25. The length upper bound U is not imposed here. (A plot analysis of Dataset 5 can be found in the Supplementary Material.) CNVDetector detects amplification regions in chromosomes 8, 17 and 19. It also finds deletion regions in chromosomes 5, 7, 8, 9, 16, 17 and 19. These regions are also found in the results from Yamashita *et al.* (2007). However, some amplification regions detected in chromosomes 1, 4, 6 and 21 by CNVDetector are not found in the previous results. The amplification region in chromosome 21 is located on 21q22.11–21q22.3 and its average (0.366) is much higher than 0. Moreover, Yamashita *et al.* (2007) detect the boundaries by a moving window of 1-Mb width, which may not detect the probes on the boundaries of CNVs. CNVDetector does not have this side effect since it is not a window-based method.

We also apply Lipson *et al.*'s original framework (StepGram) to Dataset 5. We use the default setting (8.0) for the threshold and set minDiff to 0. StepGram found 136 regions in Dataset 5 and 80 regions of them contain less than five probes. These short regions are not statistically meaningful without the normality assumption. If the genome is divided by one short region, it may lose some probes on the boundaries of CNVs. For example, our method finds an amplification region in 1p34.1–1p13.2 (from probe 980 to probe 3888); StepGram loses some probes on the boundary (from probe 980 to probe 1087) since it divides the genome by one probe in 1p32.3 (probe 1070). These probes should be detected since the score of region $[980,3888]$ is greater than that of region $[1088,3888]$ in both two frameworks.

ACKNOWLEDGEMENTS

We thank Dione K. Bailey, Steve Laderman and Yi-Shing Lin for the instructions of the Agilent Human Genome CGH Oligo Microarray Kit. We thank Ming-Chiang Chen, Chun-Ming Ho, Ping-Hui Hsu, Shu-Yu Huang, Cheng-Wei Luo, Hung-Lung Wang and Roger Yang for helpful discussion.

Funding: National Science Council, Taiwan (95-2221-E-002-126-MY3 and 96-2221-E-002-034 to P.-A.C., H.-F.L. and K.-M.C.).

Conflict of Interest: none declared.

REFERENCES

Bernholt,T. *et al.* (2007) A geometric framework for solving subsequence problems in computational biology efficiently. In *Proceedings of the 23rd Annual Symposium on Computational Geometry*, pp. 310–318.

Hogg,R.V. and Tanis,E.A. (2006) *Probability and Statistical Inference*. 7th edn. Pearson Prentice Hall.

Hupé,P. *et al.* (2004) Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics*, **20**, 3413–3422.

Lai,W.R. *et al.* (2005) Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics*, **21**, 3763–3770.

Lipson,D. *et al.* (2006) Efficient calculation of interval scores for DNA copy number data analysis. *J. Comput. Biol.*, **13**, 215–228.

Olshen,A.B. and Venkatraman,E.S. (2004) Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, **5**, 557–572.

Picard,F. *et al*. (2005) A statistical approach for array CGH data analysis. *BMC Bioinformatics*, **6**, 27.

Pinkel,D. *et al*. (1998) High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nat. Genet.*, **20**, 207–211.

Redon,R. *et al*. (2006) Global variation in copy number in the human genome. *Nature*, **444**, 444–454.

Willenbrock,H. and Fridlyand,J. (2005) A comparison study: applying segmentation to array CGH data for downstream analyses. *Bioinformatics*, **21**, 4084–4091.

Yamashita,Y. *et al*. (2007) Analysis of chromosome copy number in leukemic cells by different microarray platforms. *Leukemia*, **21**, 1333–1337.