A Novel Approach to detect Copy Number Variation using Segmentation and Genetic Algorithm

Chihyun Park[1], Youngmi Yoon[1,2], Jaegyoon Ahn[1], Myungjin Moon[1] and Sanghyun Park[1]

1. Department of Computer Science, Yonsei University, South Korea
2. Department of Information Technology, Gachon University of Medicine and Science, South Korea

{tianell, amyyoon, ajk, psiwind, sanghyun}@cs.yonsei.ac.kr

ABSTRACT

Among many forms of genomic variations, copy-number variations (CNVs) can be defined as gains or losses of several kilobases to hundreds of kilobases of genomic DNA. Since many CNVs include genes that result in differential levels of gene expression, CNVs may account for a significant proportion of normal phenotypic variation. Some scientists demonstrated that a large portion of overlapping, currently known common human CNVs, were smaller in his dataset. However, previous experimental studies, performed primarily by a-CGH techniques, are limited to detection of CNVs of large-sized CNVs. Efficient algorithms for finding small-sized CNVs are essential. In our paper, we propose a novel approach to find small-sized CNVs on a-CGH data which is a sequential 2-dimensional clustering method. The algorithm we propose is robust to some level of noise. And regardless of the size of probes, our algorithm can find CNVs consisting of small number of probes.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications – *Data Mining*; J.3 [Life and Medical Sciences] : *Biology and genetics*;

Keywords

Copy number variation, a-CGH, WGTP, Segmentation, Genetic Algorithm, Parameter estimation

1. INTRODUCTION

Variation in the human genome is present in many forms, including Single-Nucleotide Polymorphisms (SNP), small insertion-deletion polymorphisms, variable numbers of repetitive sequences, and genomic structural alterations [16]. Among these genomic variations, Copy-Number Variations (CNVs) can be defined as gains or losses of several kilobases to hundreds of kilobases of genomic DNA among phenotypically normal individuals [6]. Since many CNVs include genes that result in differential levels of gene expression, CNVs may account for a significant proportion of normal phenotypic variation [4].

Recently, the researches relating to the genomic variation of human genome are being actively carried out. Two representative platforms to assess CNVs are as follows: (1) comparative analysis of hybridization intensities on SNP genotyping array (2) comparative genomic hybridization with a Whole Genome TilePath (WGTP) array. WGTP platform comprises more than 90% of the euchromatic portion of the human genome and microarray Comparative Genomic Hybridization (a-CGH) data of human genome are being generated. In this paper, we analyzed the data from WGTP platform because this platform is prevalent in current CNV assay.

Perry [9] repeated the comparison with CNVs called by the Redon [12], revealed that 213 of 264 overlapping CNVs (80%) were smaller in his dataset, with 154 of the 264 CNVs (58%) smaller by more than 50%, and concluded that the total genomic content of currently known common human CNVs is likely to be smaller than previously thought. However, previous experimental studies, performed primarily by a-CGH techniques, are limited to detection of CNVs of large-sized CNVs, tens or hundreds of kilobases [14]. Efficient algorithms for finding small-sized CNVs are essential, and we focused on this problem.

In our paper, we propose a sequential 2-dimensional clustering method to find small-sized CNVs. Our algorithm uses log ratio value and position information from WGTP sample and finds segments which are used for scoring phase with six parameters. We assign scores to the probes based on the average log ratio value of the segments, and find CNVs by selecting top scoring probes. Genetic Algorithm (GA) helps to estimate six optimal parameters because their search spaces are wide. GA has excellent exploration power that provides the capability of escaping from local optima and working well when solutions to a problem contain complex interacting part [2].

There are two types of CNVs. One is called gain, which means relatively duplicated part compared with a reference sample. The other is called loss, which means relatively deleted part compared with a reference sample. There could be a possibility that the segments having high intensity ratio which is supposed to be a gain could not be a genomic duplication since a-CGH data have some level of noise in acquired hybridization intensity ratios. The proposed algorithm is robust to some level of noise. Regardless of the size of probe, our algorithm can find CNVs consisting of small number of probes. In other words, if the a-CGH dataset is from higher-resolution tiling arrays, our algorithm can detect small-sized CNVs.

2. RELATED WORKS

2.1 SW-ARRAY

SW-ARRAY [11] is a popular method to find CNVs on a-CGH

SAC'09, March 8-12, 2009, Honolulu, Hawaii, U.S.A.

platform. SW-ARRAY adapted the Smith-Waterman dynamic programming algorithm to provide a sensitive and robust analytic approach for CNV detection. This method scans through a-CGH data using one threshold and finds a segment while giving penalty, if some unexpected data is inserted into the segment. Finally this method applies robustness value to the segments found to select only sensitive segments. SW-ARRAY method takes its basic idea from Smith-Waterman algorithm. The major feature of Smith-Waterman algorithm is to find a segment which is the longest and continuous as far as possible. SW-ARRAY also can find long consecutive gain or loss from a-CGH data. However this method is not suitable for finding short segments. Still SW-ARRAY is popular in many researchers due to its simple principle.

2.2 CNV-FINDER

The purpose of CNV-finder [3] is to find proper threshold to minimize false rate. False positive and false negative are important problem in CNV detection. CNV-finder finds optimal value to minimize false rate by additional experiments and maximize the number of CNV found by statistical approach. They hypothesize that the majority of observations are normally distributed around a $\log_2$ ratio of zero. They used multiples of the SDe (N*SDe(Standard Deviation)) to define positive and negative thresholds beyond which ratios are unlikely to occur by chance in the absence of CNV. CNV-finder is a representative detection-method like SW-ARRAY and has some advantages compared with SW-ARRAY. This method can minimize false rate. Since it finds CNVs by only SD values without considering the length of the CNVs, it is not robust to the noise of a-CGH data.

2.3 OTHER METHODS

There are many algorithms other than two methods previously mentioned: HMM [5], CGHseg [10], CLAC [15]. Lai [7] compared 11 different algorithms for analyzing array CGH data and detailed the relative merits of these methods. Lai [7] computed the Receiver Operating Characteristic (ROC) curves to quantify sensitivity and specificity for various levels of signal-to-noise ratio and different sizes of abnormalities. Most algorithms did work well in detecting aberrations with large width and high SNR (Signal-to-Noise Ratio). The experimental results of Lai [7] revealed that most algorithms did not reliably detect the aberrations with small width and low SNR because the signal is too weak to be differentiated from the noise.

2.4 GENETIC ALGORITHM

Evolutionary Computation (EC) is a population-based stochastic iterative optimization technique based on the Darwinian concepts of evolution. Inspired by these principles, like survival of the fittest and selective pressure, EC tackles difficult problems by evolving approximate solutions of an optimization problem inside a computer. An algorithm based on EC is called an Evolutionary Algorithm (EA) [1]. GA is the most prominent example of EA. GA is widely used to solve a problem which has to find an optimal solution [8]. GA begins with a "population" which is a set of encoded chromosomes, and evolves population similar with real world evolution. During evolution, a fitness function measures the degree of optimum in a set of chromosomes and the set of chromosomes are processed under selection and genetic operators.

3. ALGORITHM

3.1 SYSTEM OVERVIEW

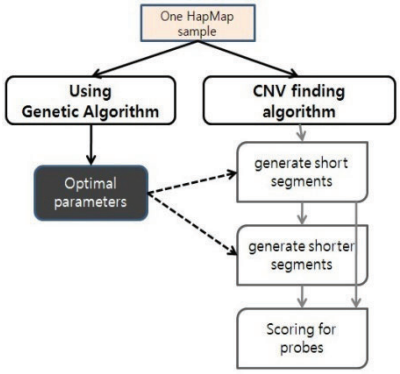

Figure 1: Overall Process of our algorithm

In the first stage, the optimal parameters are estimated by GA. Parameters are estimated using one HapMap [13] sample which is the target for CNV detection. The obtained parameter from first stage is used in the second stage. CNV finding algorithm which is second stage consists of three phases as shown in Fig. 1. The segment is defined as a small subset of probes. A short segment consists of probes whose log ratio values are similar and whose positions are consecutive, with some gapped probes inside. The log ratio is the quantified expression value of the Cyanine 3 to Cyanine 5 intensities for each probe. The first phase is to find initial segments from a-CGH data. The second phase is to find shorter segments from segments generated by first phase. However, only some of segments that can be divided into shorter ones are chosen. Then we assign scores to each probe according to the average log ratio value of all the probes in the short segments to which the probe belongs, and determine whether the probe is gain or loss.

3.2 PARAMETER ESTIMATION USING GA

There are six parameters in our algorithm, which are shown in Fig. 2. Six parameters divide the data into segments through sequential 2-dimensional clustering procedure. Log ratio parameter and weight of log ratio parameter decide the size of segment. And position parameter and weight of position parameter decide the continuity of segment. Different segments can be generated according to these parameters, and finally these segments influence CNV labeling process. Thus we have to find the best set of parameter combination whose range is wide as shown in Fig. 2.

3.2.1 Encoding scheme for parameters

In this paper, we encode a solution like as Fig. 2. One chromosome is composed of six parameters which can be presented by real numbers. The reason for not using binary encoding schema is that we want to maintain the building block which represents the solution fitted within our encoding boundary.

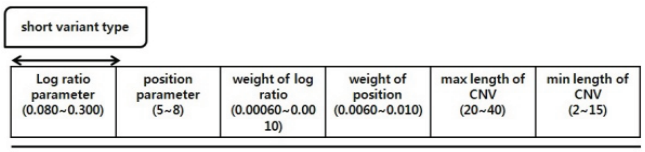

Figure 2: Encoded individual or chromosome

3.2.2 Selection mechanism and genetic operators

In our system, selection mechanism is a tournament scheme and genetic operators are one point crossover and mutation. In our system, crossover rate is 0.9 and mutation rate is 0.03. Two rates 0.9 and 0.03 are obtained by set of experiments.

3.2.3 Fitness evaluation

The fitness function measures the quality of the chromosomes. Therefore, the better fitness of chromosome is measured, the more possibilities that the chromosome is selected for reproduction exist, and the more parts of its genetic material will be passed on to the next generations [1]. In this paper, fitness function is the number of the segments called CNV candidates which consist of low or high log ratio value of probes. CNV candidate is defined as a segment of consecutive probes that have log ratio values lower or higher than average of log ratio value of all the segments. The reason why we use this fitness function is that the more segments which consist of low or high log ratio values are generated, the more correct scoring value of the probes can be calculated.

3.3 CNV finding algorithm

Detecting CNVs from a-CGH data is same as finding a continuous segment having similar log ratio values. In this paper, we propose a novel algorithm to segment log ratio values of a-CGH data efficiently. Our algorithm consists of three main phases mentioned at the beginning of the section 3.

The first phase of our algorithm is suggested in order to generate temporary segments while considering log ratio values and position values of probes from given sample. Our algorithm uses three parameters which are log ratio parameter, position parameter and minimum length parameter chosen from GA. From now on, we call the probe an element in a segment.

The difference between any two consecutive log ratio values in the same segment should not exceed the log ratio parameter which is a result of the GA. Because input data are sorted by log ratio value, we can easily get elements which have similar log ratio values. Note that the positions within one segment are not ordered since segments are built using log ratio value. Because position values should be considered for continuity of the segment, our algorithm sorts elements in each segment by their position. The difference of positions of any two adjacent elements in a segment should not exceed the position parameter which is also a result of the GA. While considering position parameter, the minimum length of segment should be satisfied. If a segment does not satisfy the minimum length, the consecutive elements can be included into that segment. As a result, segments are composed of the elements which have continuous position values and similar log ratio values. Therefore elements in a same segment can have same label.

In the first phase, outlier probes can be included in the segments according to those three parameters mentioned above. However, outliers are not consecutive and usually their log ratio values are abnormally high or low. In this case, the log ratio parameter and position parameter filter these probes out, and made them be exceptionally short segments. The parameter, minimum length of segments, prunes these segments. As a result, outlier probes are not included in any segments. Outlier probes have no chance to be labeled as gain or loss since their score is not calculated at all.

After phase 1, our algorithm checks whether the segments which are identified in the first phase can be divided into shorter segments or not. The possibility of each segment being divided further is determined by the $F_{select\_measeure}()$ function as follows:

$$F_{select\_measure}(\text{segment}) = |N_E - (P_L - P_F)| + (P_L - P_F)$$

N_E = Element Number of Segment
P_L = Position of Last element
P_F = Position of First element

This function measures how the elements of a segment are distributed and how many there are missing elements in this segment. Well divided segment has the property that it has a few number of missing elements, and the position distance between the first element and the last element is short. If a candidate segment has many missing elements or the position distance between the first element and the last element is long, this segment should be divided into shorter segments further. To choose relatively long segments, ones whose $F_{select\_measeure}()$ is greater than the average of $F_{select\_measeure}()$ of all segments are selected as long segments.

The inputs of the second phase are segments selected by the first phase. Through this phase, we can get shorter and more continuous segments that have much more similar log ratio value than the input segment. In this phase, our algorithm uses three parameters which are two weights and maximum length of segment. The two weights are related with new log ratio parameter and new position parameter.

Input segments are sorted by log ratio value and shorter segments are generated using new ratio parameter. The elements of newly created segments are sorted by their position and the shorter segments are generated using new position parameter. While generating new segment, the maximum length of segment should be considered. The formulas for estimating the new log ratio and position parameters are following. GA finds optimal weight value, *Wr* and *Wp* for new log ratio parameter and position parameter, respectively, which reflect the length of long segments. These two new thresholds have a primary role to build shorter segments.

$\text{Threshold}_{postion} = P_p - (P_p * F_{s\_m} * W_p)$ $\quad$ $\text{Threshold}_{ratio} = R_p - (R_p * F_{s\_m} * W_r)$

P_p = *Position Parameter* $\quad$ R_p = *Ratio Parameter*

W_p = *adjust weight* $\quad$ W_r = *adjust weight*

$F_{s\_m} = F_{select\_measure}(\text{segment})$ $\quad$ $F_{s\_m} = F_{select\_measure}(\text{segment})$

Then, our algorithm assigns scores to each probe according to the short segments to which that probe belongs.

Let the short segment be s_i and let the average of element's log ratio values of s_i be avg_i. After segmentation process, each probe may belong to one or more short segments or may not belong to any segments. Therefore, the probe may have a set of *avg*s, $\{avg_i \mid$ the probe belongs to $s_i\}$.

If every *avg*s of the probe are commonly low, then the probe can be considered as loss. And if every *avg*s of the probe are commonly high, then the probe can be considered as gain. However, if the variance of *avg*s is high, then this probe can be neither gain nor loss. For example, suppose that the probe belongs to two segments s_1 and s_2, where avg_1 = -0.5 and avg_2 = 0.5 (-0.5

and 0.5 are relatively high values). Then there is high possibility that the log ratio value of the probe is noisy, and we can ignore the probe.

By the observations above, we can build a score measure as follows:

$$score = \text{variance of S} - |\text{average of S}|$$

, where S is the set of *avgs* of the probe. We can easily know that the probe with a set of *avgs* which have lower variance and higher absolute the value of average can get a lower score. The probes with lower scores can be considered as gain or loss more certainly. Therefore, we can check top n probes, while increasing n. For example, the first 10 probes are considered as definitely gains or losses, and next 10 probes can also be gains or losses, but the reliability of the next 10 probes as gains or losses are not high as those of previous 10 probes.

4. EXPERIMENTAL RESULTS

In this section, we describe the experimental results of parameter estimation process using GA and CNV finding algorithms. We used HapMap samples using array-based comparative genome hybridization with a genome-wide WGTP array consisting of ~27000 large-insertion clones. The data is available on the web site [17]. The data were using WGTP array with dye-swap for 269 HapMap individuals and using a single male reference, HapMap individual NA10851 [17]. We used HapMap sample, NA07055 for experiment. Among 23 chromosomes, we used chromosome 19. Our approach can be applied to any chromosome and any HapMap sample.

4.1 Parameter estimation results

GA found the optimal result within 200 generation. The value of 190th generation was the optimal solution. The following Table 1 shows optimal solution, which means the best combination of parameters.

Table 1: Found optimal parameters by GA

HapMap sample, NA07055 parameters by GA		
Set of optimal parameter	Log ratio parameter	0.086
	Position parameter	6
	Weight of log ratio	0.008
	Weight of position	0.01
	Minimum length of Segment	4
	Maximum length of Segment	26

Table 2: Top 10~90 probes from out result

The number of Top scoring probes(case)	The range of scores	Gain probes	Loss probes
10	-0.12689 ~ -0.08610	6	4
30	-0.12689 ~ -0.06770	11	19
50	-0.012689 ~ -0.05080	15	35
70	-0.012689 ~ -0.04442	28	42
90	-0.012689 ~ -0.04131	37	53

4.2 Analysis of algorithm result

The experiments of CNV finding algorithm consist of the two parts. The first part describes the relation between the number of top scoring probes and our result. The second part describes the comparison of our result with other methods. We made comparison with other algorithms for all the cases in Table 2. Here we exhibited the comparison result for top 70-case.

4.2.1 The first part of algorithm experiment

Table 2 shows the relation between the number of top scoring probes and the number of probes that can be gain or loss. As we have already mentioned, the probes with lower scores are considered as gain or loss more certainly. Therefore, while we choose top n scoring probes, we increase n so that chosen probes can be considered to be gain or loss, and also can compose long CNVs.

4.2.2 The second part of algorithm experiment

Fig. 3 shows the raw a-CGH data and CNVs that is composed of 70 probes found by our algorithm. The coordinate x indicates the position of each probe in a-CGH data. In Fig. 4 and 5 the coordinate y indicates the position of each probe in a-CGH data. Fig. 4 shows the comparison of our algorithm with SW-ARRAY for the same HapMap sample NA07055. SW-ARRAY found two CNVs, one is gain and the other is loss, as shown in the left and right graph of Fig. 4, respectively. Our algorithm found 23 CNVs, while SW-ARRAY found by only 2 CNVs, and the length of CNVs was shorter. There are overlapped areas found by two methods, but these areas are not wide because of the difference in the basic idea of two algorithms. SW-ARRAY is based on the Smith-Waterman algorithm which finds the longest segments if possible while giving penalty when unexpected value is encountered. However, our algorithm divides one sample into short segments as much as possible not accepting large unexpected value. Fig. 5 shows the comparison of our algorithm with Redon's. We can see that CNVs found by our algorithm and Redon's CNV event loci were not exactly matched, because our experiment was carried out only using NA07055, Redon's result was CNV regions which are the union of many CNVs from all of 270 HapMap samples.

5. CONCLUSION

The main contribution of this paper is to suggest a novel approach which can find small-sized CNVs. In order to build short segments, several parameters are used, and these parameters are obtained through genetic algorithm. The main algorithm to find short segment consists of three phases. Former two phases find short segments. In the third phase, our algorithm assigns scores to each probe, and determines whether the probe is gain or loss. The experiments were carried out with chromosome 19 of HapMap sample, NA07055. The result of experiments demonstrates that our algorithm is able to find small-sized CNVs which could not be found using other algorithms. We cannot be sure that all the CNVs found by our algorithms are real ones until biological validation like RT-PCR is performed. However, algorithms which can find small-sized CNVs are essential since currently known common human CNVs are likely smaller than previously thought. Our experimental results show that CNVs which were not detected by Redon's and/or SW-ARRAY method were detected by our algorithm. We are currently investigating lowering the false rate for future works.

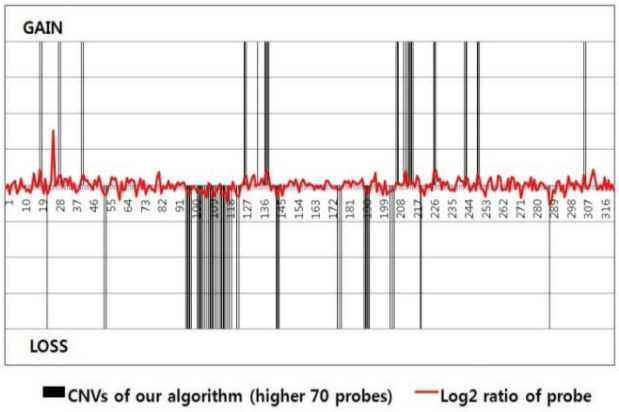

Figure 3: Raw aCGH data and result of our algorithm

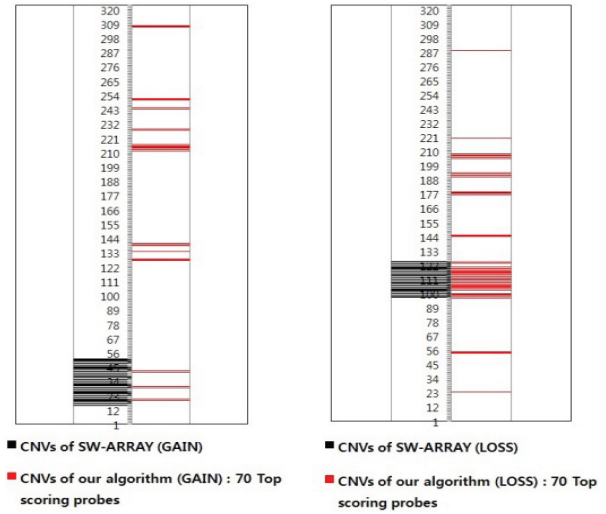

Figure 4: Comparison of our algorithm with SW-ARRAY

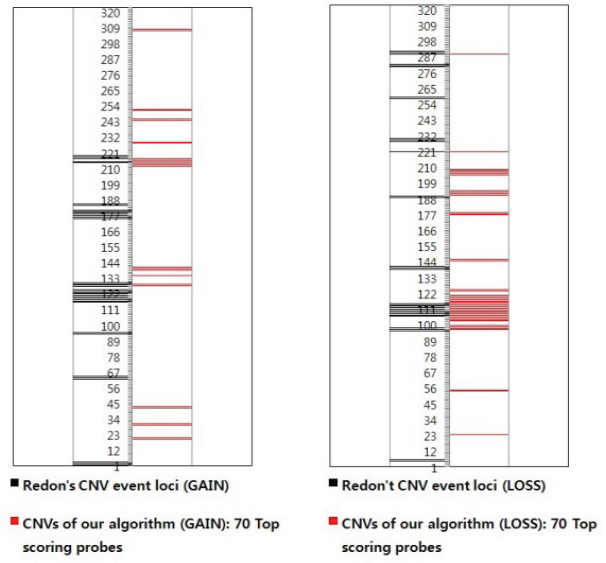

Figure 5: Comparison of our algorithm with Redon's CNV event loci

6. ACKNOWLEDGMENTS

"This work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MOST) (No. R01-2006-000-11106-0)."

7. REFERENCES

[1] F. Divina, J. S. Aguilar-Ruiz, "Biclustering of Expression Data with Evolutionary Computation," IEEE Trans. On Knowledge and Data Engineering, KVol.18, 2006, PP.590-602.

[2] A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing, Springer-Verlag, 2003.

[3] H. Fiegler, R. Redon, D. Andrews, et al., "Accurate and reliable high-throughput detection of copy number variation in the human genome," Genome Research, Vol. 16, 2006, pp.1566-1574.

[4] J. L. Freeman, G.H. Perry, et al. ,"Copy number variation: New insights in genome diversity," Genome Research, Vol.16, 2006, pp.949-961.

[5] J. Fridlyand, et al., "Hidden Markov Models approach to the analysis of array CGH data," J. Multivariate Anal., Vol. 90, 2004, pp.132-153.

[6] A.J. Iafrate, L. Feuk, M.N. Rivera1, M.L. Listewnik, et al., "Detection of large-scale variation in the human genome", Nature Genetics, Vol. 36, 2004, pp.949-851.

[7] Weil R. Lai, et al, "Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data," Vol.21, 2005, pp.3763-3770.

[8] M. Mitchell, "An introduction to genetic algorithms", A Bradford Book The MIT Press, 1999.

[9] G. H. Perry, A. Ben-Dor, A. Tsalenko, et al., "The Find-Scale and Complex Architecture of Human Copy-Number Variation," The American Journal of Human Genetics, Vol. 82, 2008, pp.685-695.

[10] F. Picard, et al., "A statistical approach for array CGH data analysis," BMC bioinformatics, Vol. 6, 2005

[11] T.S. Price, R. Regan, R. Mott, et al. ,"SW-ARRAY: a dynamic programming solution for the identification of copy-number changes in genomic DNA using array comparative genome hybridization data," Nucleic Acids Research, Vol. 33, No. 11, 2005, pp. 3455-3464.

[12] R. Redon, S. Ishikawa, KR. Fitch, et al. ,"Global variation in copy number in the human genome," Nature, Vol.444, 2006, pp. 444-454.

[13] The International HapMap Consortium, "A Haplotype map of the human genome," Nature, Vol. 437, 2005, pp.1299-1320.

[14] K Wang, M Li, D Hadley, et al., "PennCNV: An integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data," Genome Research, Vol.17, 2007, pp.1665-1674.

[15] P. Wang, et al., "A method for calling gains and losses in array CGH data," Biostatistics, Vol. 6, 2005, pp.45-58.

[16] A.F. Wright, Nature Encyclopedia of the Human Genome vol. 2, 959–968 (Nature Publishing Group, London, 2003).

[17] http://www.sanger.ac.uk/humgen/cnv/data/