

Technical Note

A Performance Comparison of Two CGH Segmentation Analysis Algorithms: DNACopy and segMNT

Segmentation analysis of array Comparative Genomic Hybridization (array CGH) data identifies regions of copy number change in the genome. Two segmentation algorithms are available for the analysis of NimbleGen array CGH: DNACopy and segMNT. Following extensive testing of these algorithms, Roche NimbleGen concluded the segMNT algorithm is a superior analytical tool because it more accurately detects copy number changes and their associated breakpoints, can more effectively detect copy number changes in raw (unaveraged) data, and has a predictable processing time that is often faster than the DNACopy algorithm.

Here we provide a brief description of the DNACopy and segMNT algorithms and compare their performance with datasets acquired using NimbleGen high-density array CGH.

Introduction to the DNACopy and segMNT Algorithms

The DNACopy and segMNT algorithms are available in Roche NimbleGen NimbleScan software (1, 2). Each algorithm attempts to separate array CGH data into regions of homogenous copy number, or segments, and map breakpoints between these segments. How the algorithms segment and map breakpoints for array CGH data is different and described below.

DNACopy Algorithm

The DNACopy algorithm identifies copy number changes using circular binary segmentation.

1. Determine breakpoints: This step recursively divides contiguous segments of a data track each into two or three segments. A t-like statistic and permutation-based critical value are used to justify the optimal and significant segmentation. Starting from the entire data track, the procedure is iterated until further segmentation is no longer statistically significant. This step is time consuming.

2. Remove unmeaningful breakpoints: Local trends in probe values can lead to determination of biologically unmeaningful breakpoints. The DNACopy algorithm applies a pruning procedure to remove breakpoints that do not significantly improve the segmentation result.

The segMNT Algorithm

The segMNT algorithm identifies copy number changes using a dynamic programming process that minimizes the squared error relative to the segment means.

1. Generate a list of candidate breakpoints: A sliding window t-like statistic test is the key component of this step. Through this test, each probe in the data track is given an empirical p-value quantifying the probability that the flanking regions are of unequal copy number. A pre-determined number of probes of the smallest p-value are selected as candidate breakpoints. The number of candidate breakpoints can be adjusted and impacts the algorithm's sensitivity: too few and not all of the breakpoints will be found, too many and the runtime increases greatly.

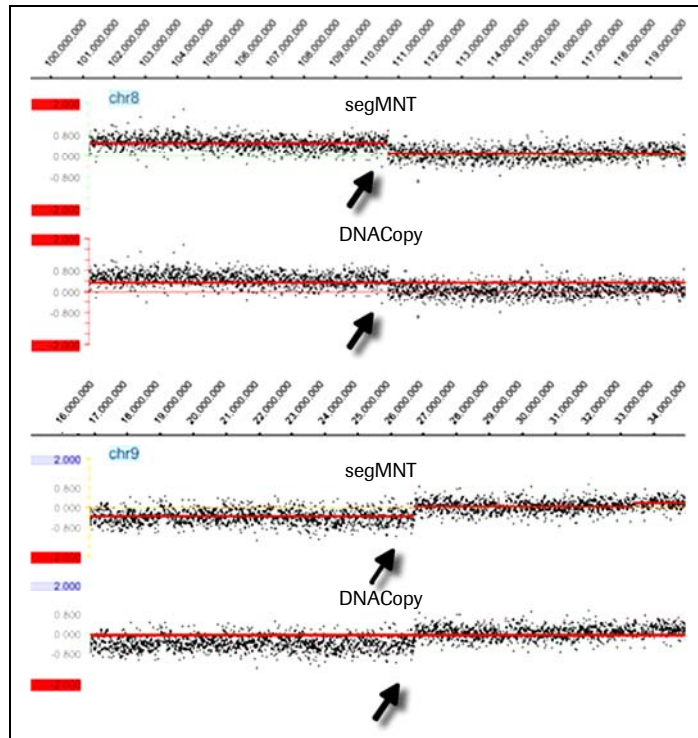


Figure 1. The segMNT Algorithm Detects Breakpoints Not Detected by the DNACopy Algorithm. Array CGH data analyzed by the DNACopy and segMNT algorithms reveal the difference in their performance. The segMNT algorithm successfully detected breakpoints (indicated by a break in the red line) produced by large amplifications (chromosome 8, top panel) and deletions (chromosome 9, bottom panel) that the DNACopy algorithm missed.

2. Identify the best segmentation for each given number of breakpoints: Optimization by dynamic programming is the key component of this step. The best segmentation of n breakpoints is determined for each n ranging from 1 to N , the maximum number of breakpoints to be considered per chromosome. “Best,” for this purpose, means minimizing the sum of squares error around the segment mean (noted as $\text{minvar}(n)$). An important difference here from the DNACopy algorithm is that the segMNT algorithm uses dynamic programming in a confined search space to find the globally optimal solution. The DNACopy algorithm locates the best segment(s) but is not guaranteed to find anything but the locally optimal solution.

3. Determine the number of segments to output: The stopping criterion is the key component of this step. The segMNT algorithm uses permutation to justify the number of segments to output as the result. Steps 1 and 2 are applied to a number of scrambled data tracks (e.g. 20), and the segment mean from each of these scrambled data tracks are recorded. The final segmentation produced by the algorithm is the one of the largest n , where the difference in the minimal sum of squares error between n and $n-1$ segments for the original data ($|\text{minvar}(n-1)-\text{minvar}(n)|$) is significantly different from the cumulative maximum minvar-difference for the permuted data ($\max\{|\text{minvar}(m-1)-\text{minvar}(m)|; m = 1, \dots, n\}$).

Performance Comparison between the DNACopy and segMNT Algorithms

The segMNT algorithm consistently shows better performance than the DNACopy algorithm in several aspects:

- Using datasets with known copy number changes, the segMNT algorithm finds more true positive and fewer false positive breakpoints than the DNACopy algorithm. Figures 1, 2, and 3 show the superior detection capabilities of the segMNT algorithm when compared to the DNACopy algorithm.
- The DNACopy algorithm often requires window averaging of \log_2 -ratio data to detect copy number changes. When data are window-averaged, resolution is decreased. The segMNT algorithm works well with unaveraged (raw) data, which enables detection of copy number changes at maximum resolution.
- Through the use of constrained dynamic programming, the segMNT algorithm allows researchers the flexibility to set a minimum required difference between neighboring segments. This setting can eliminate over-segmentation in noisy datasets and improve performance.
- Given a defined number of candidate breakpoints, the segMNT algorithm finishes in a more predictable amount of time and in some instances more than 20-fold faster than the DNACopy algorithm. The processing time for the DNACopy algorithm varies greatly, depending on the number of data points in each region and the amount of noise in the data. Refer to Figure 4 for a comparison of the processing times of the segMNT and DNACopy algorithms.

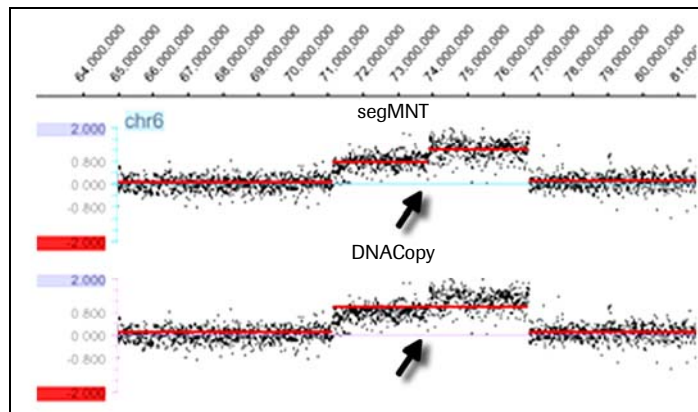


Figure 2. The segMNT Algorithm Distinguishes Multiple Copy Number Differences Not Detected by the DNACopy Algorithm. In human chromosome 6, two contiguous but discrete amplification events were correctly identified by the segMNT algorithm. The DNACopy algorithm correctly recognized the amplification event but was unable to distinguish the complex structure of the region.

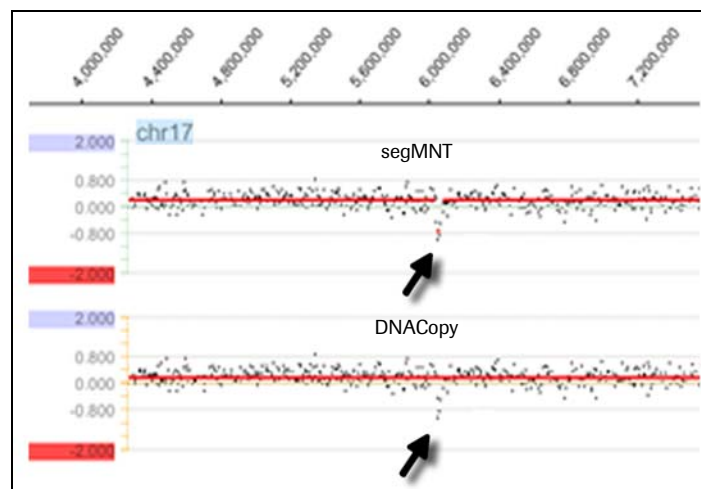


Figure 3. The segMNT Algorithm Identifies a Microdeletion That the DNACopy Algorithm Fails to Detect. Data from human chromosome 17 show a 6kb deletion successfully detected by the segMNT algorithm but not identified by the DNACopy algorithm.

Array Platform	Array Description	Processing Time	
		DNACopy	segMNT
385K (372,497 data points)	Single Region - Array 1	0m 31s	2m 15s
	Single Region - Array 2	37m 39s	2m 13s
	Single Region - Array 3	10m 45s	2m 38s
385K (385,807 data points)	Whole Genome - Array 1	1m 13s	13m 6s
	Whole Genome - Array 2	1m 57s	12m 59s
	Whole Genome - Array 3	0m 30s	12m 48s
HD2 (2,146,888 data points)	Single Region - Array 1	663m 48s	31m 12s
	Single Region - Array 2	382m 58s	31m 2s
	Single Region - Array 3	298m 10s	31m 1s
HD2 (2,175,073 data points)	Whole Genome - Array 1	27m 29s	24m 59s
	Whole Genome - Array 2	28m 47s	24m 55s
	Whole Genome - Array 3	26m 28s	24m 48s

Figure 4. Processing Times for the segMNT Algorithm Are More Predictable and Usually Faster Than for the DNACopy Algorithm.

A comparison of the processing times for segmentation analysis reveals that the DNACopy algorithm can be highly variable, based on data quality and number of copy number events. Processing times for the segMNT algorithm are more predictable and in some instances more than 20-fold faster than the DNACopy algorithm. The segMNT algorithm is particularly well-suited to the 2.1 million data points of the NimbleGen HD2 platform.

Conclusion

Roche NimbleGen concluded that the segMNT algorithm provides improved performance to the DNACopy algorithm in the analysis of array CGH data. The segMNT algorithm more accurately detects copy number changes and their associated breakpoints, can use unaveraged (raw) data, and provides a predictable, often faster processing time.

Notes

1. Roche NimbleGen NimbleScan v2.4 software processes CGH data using the DNACopy v1.4 algorithm (Olshen AB, et al. 2004) or segMNT v1.1 algorithm, based on the user's selection.
2. In January 2008, the Roche NimbleGen service lab switched CGH data analysis from the DNACopy v1.6 algorithm to segMNT v1.1 algorithm.

Reference

Olshen AB, et al. Circular binary segmentation for the analysis of array based DNA copy number data. *Biostatistics* 2004; 5(4):557-72.

For More Information

Toll-free in US: (877) NimbleGen / (877) 646-2534
 (608) 218-7600
 ngsales@nimblegen.com
 www.nimblegen.com



HIGH - DEFINITION GENOMICS™

© June 2008 Roche NimbleGen, Inc. All Rights Reserved.
 05338310001 • Reprinted 06/08 • Original Publication 01/08



Roche NimbleGen, Inc.
 500 S. Rosa Road
 Madison, WI 53719 USA