BIOINFORMATICS APPLICATIONS NOTE Vol. 26 no. 10 2010, pages 1366–1367
doi:10.1093/bioinformatics/btq155

Genome analysis

CGHnormaliter: a Bioconductor package for normalization of array CGH data with many CNAs

Bart P.P. van Houte, Thomas W. Binsl, Hannes Hettling and Jaap Heringa*

Centre for Integrative Bioinformatics VU (IBIVU), VU University Amsterdam, De Boelelaan 1081A, 1081 HV Amsterdam, The Netherlands

Associate Editor: Alfonso Valencia

ABSTRACT

Summary: CGHnormaliter is a package for normalization of array comparative genomic hybridization (aCGH) data. It uses an iterative procedure that effectively eliminates the influence of imbalanced copy numbers. This leads to a more reliable assessment of copy number alterations (CNAs). CGHnormaliter is integrated in the Bioconductor environment allowing a smooth link to visualization tools and further data analysis.

Availability and Implementation: The CGHnormaliter package is implemented in R and under GPL 3.0 license available at Bioconductor: http://www.bioconductor.org

Contact: heringa@few.vu.nl

Received on November 20, 2009; revised on March 12, 2010; accepted on April 7, 2010

1 INTRODUCTION

Array comparative genomic hybridization (aCGH) is a popular experimental technique for detection of copy number alterations (CNAs) at high resolution. Its principal application areas are cancer research and clinical genetics. Recently, in Britain the first baby was born with the help of aCGH to screen eggs for a normal number of chromosomes, after the mother had undergone 13 failed attempts at IVF and three miscarriages (BBC News, 2009, http://news.bbc.co.uk/2/hi/health/8232146.stm).

Since resolutions of (oligonucleotide) aCGH chips increase constantly and already reach up to 1 M probes, experiments yield massive amounts of data. Normalization is an important first step in the analysis of these data and aims at minimizing the effect of the experimental bias (e.g. dye bias) in the measurements. Usually standard methods originating from the gene expression data area, such as global-median and LOWESS (Hwa Yang *et al.*, 2002) normalization, are employed for this purpose. However, as Staaf *et al.* (2007) have convincingly shown, application of these standard techniques to aCGH data with many CNAs lead to an improper centralization and hence to inaccurate downstream analyses.

In the recent past, a few aCGH normalization methods for two-dye aCGH data have been designed that deal with this issue. Their common approach is filtering of CNAs from the data to calculate a more appropriate median value or LOWESS regression curve for the whole dataset. As a result, after normalization, the $\log_2$ intensity ratios of the non-CNAs (normals) will generally be closer to zero and hence better reflect the biological reality. S-Lowess (Supervised Lowess) is the most low-tech method to do this, since CNAs are to be selected by hand (Van Hijum *et al.*, 2008). The recently published popLowess algorithm automatically separates the CNAs from the normals through k-means ($k=3$) clustering (Staaf *et al.*, 2007). However, 'calling' of aberrations through a clustering method is rather coarse-grained. Another recent normalization and centralization method was proposed by Chen *et al.* (2008), in which normalization is also based on the normal probes, albeit in a different fashion. In their algorithm normalization is performed by regressing the highest ridgeline of a 2D intensity distribution that is assumed to correspond to normal probes. Subsequently, the most abundant probe intensity (i.e. the highest peak in the intensity distribution) is used for centralization.

We have recently published a novel normalization method, called CGHnormaliter, which offers a more sophisticated normalization of aCGH data (Van Houte *et al.*, 2009). Initially, the $\log_2$ intensity ratios are segmented using DNAcopy (Venkatraman and Olshen, 2007). The segmented data are then given as input to a calling tool named CGHcall (Van de Wiel *et al.*, 2007) to discriminate the normals from the CNAs. These normals are subsequently used for normalization based on LOWESS. These steps are then iterated to refine the normalization.

In a comparative analysis, CGHnormaliter has been shown to outcompete the contenders mentioned above (Van Houte *et al.*, 2009). We completely reimplemented the CGHnormaliter algorithm and it has now become available as a package in the Bioconductor project. The new package makes use of Bioconductor data structures, so that it can easily be combined with other Bioconductor tools to create a workflow for advanced analysis and visualization of aCGH data. A re-evaluation of the new implementation yielded identical results as the implementation used in the original paper. Regarding the time complexity of the new implementation, the running time appears to scale linearly with the array resolution. For example, eight samples at a resolution of 32 K, 100 K and 500 K run for 0.98, 2.1 and 8.1 h, respectively.

2 DESCRIPTION

The main function of the package is `CGHnormaliter`, which performs the actual normalization of an aCGH dataset. Below, we briefly describe its use and that of additional functions. Comprehensive instructions are available in the vignette and manual pages accompanying the package.

Input: the required input is either a `data.frame` or the file name of a tab-separated text file. The first four columns should

*To whom correspondence should be addressed.

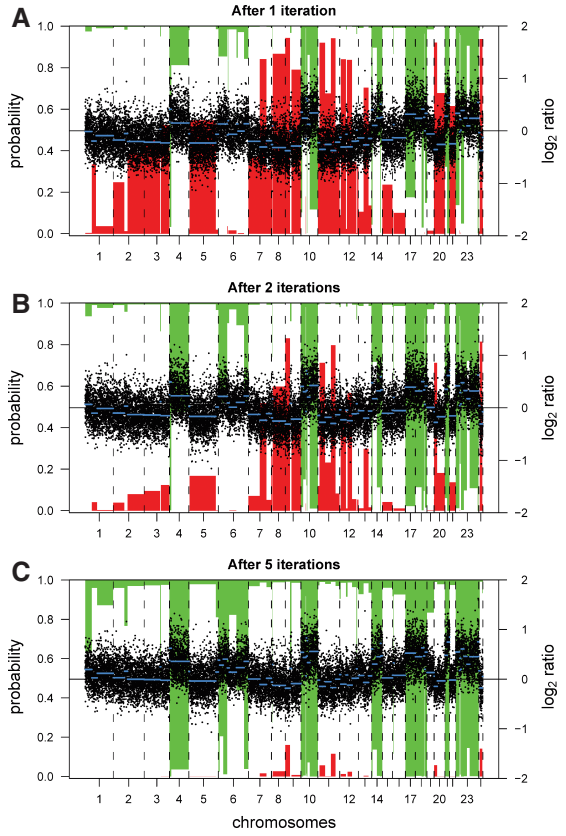

Fig. 1. Example of an iterative refinement of aCGH data normalization performed by CGHnormaliter. Normalization, segmentation and calling results on an ALL tumor sample after 1 (**A**), 2 (**B**) and 5 (**C**) iterations are shown. Normalized $\log_2$ intensity ratios and segments are represented by dots and blue horizontal lines, respectively. Aberration probabilities are indicated by the length of the green downward (gain) and red upward (loss) bars. Segments are designated gain or loss if their probabilities exceed 0.5. G-banding and FISH analyses revealed gains in eight chromosomes [4, 6, 10, 14, 17, 18, 21, 23(X)] which are largely confirmed using CGHnormaliter.

describe the clone and its position on the genome: (i) clone ID, (ii) chromosome number, (iii) start position and (iv) end position. The next columns hold the actual data. For each sample in the experiment, there must be two adjacent columns with the *reference* and *test* intensities, respectively. Three additional parameters are available to control the normalization: `nchrom` (default: 24) sets the number of chromosomes to be normalized, while `max_iterations` (default: 5) sets the maximum number of iterations and `stop_threshold` (default: 0.01) sets the threshold value for the mean difference between the LOWESS regression curves from two consecutive iterations. The iteration is terminated if this difference is below this value for all samples.

Output: the output of `CGHnormaliter` is a `cghCall` container. Its fields can be accessed via several functions. The normalized, segmented and called data can be retrieved using the respective functions `copynumber`, `segmented` and `calls`. The normalized data can be plotted using the `plot` function (Section 3). Finally, the package provides the function `CGHnormaliter.write.table` to save the normalized data into a tab-delimited plain text file.

3 EXAMPLE

To demonstrate the improvement of aCGH data normalization during the iterative process, CGHnormaliter was run on 32K bacterial artificial chromosome (BAC) aCGH data of an acute lymphoblastic leukemia (ALL) tissue sample (Paulsson *et al.*, 2006). Figure 1 depicts the results after 1, 2 and 5 iterations (by setting the input parameter `max_iterations` at the proper value). It becomes clear that the normals gradually approach the baseline as the iterative procedure progresses. In addition, the good agreement between the calling results and the fluorescent *in situ* hybridization (FISH) analysis confirms the high accuracy of the normalization results after five iterations.

4 DISCUSSION

We have developed a Bioconductor package CGHnormaliter to accurately normalize aCGH data. The algorithm effectively eliminates the 'overnormalizing' effect of CNAs as well as large-scale copy number variations (LCVs) by temporarily excluding them. Additionally, results are enhanced by iterative refinement and by employing a state-of-the-art calling method to identify these CNAs. The calling results are provided to the user as a bonus. CGHnormaliter can easily be combined with other Bioconductor or R packages to form a workflow. Moreover, the standard format of the results allows for straightforward downstream analysis of the normalized data.

Funding: Netherlands Genomics Initiative (HH: NGI/Centre for Medical Systems Biology, BvH: NGI/Ecogenomics); Netherlands Bioinformatics Centre (TB: NBIC/BioRange).

Conflict of Interest: none declared.

REFERENCES

Chen,H. *et al.* (2008) A probe-density based analysis method for array CGH data: simulation, normalization and centralization. *Bioinformatics*, **24**, 1749–1756.

Hwa Yang,Y. *et al.* (2002) Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res.*, **30**, e15.

Paulsson,K. *et al.* (2006) Identification of cryptic aberrations and characterization of translocation breakpoints using array CGH in high hyperdiploid childhood acute lymphoblastic leukemia. *Leukemia*, **20**, 2002–2007.

Staaf,J. *et al.* (2007) Normalization of array-CGH data: influence of copy number imbalances. *BMC Genomics*, **8**, 382.

Van de Wiel,M.A. *et al.* (2007) CGHcall: calling aberrations for array CGH tumor profiles. *Bioinformatics*, **23**, 892–894.

Van Hijum,S.A. *et al.* (2008) Supervised lowess normalization of comparative genome hybridization data-application to lactococcal strain comparisons. *BMC Bioinformatics*, **9** [Epub ahead of print, doi:10.1186/1471-2164-10-401].

Van Houte,B.P.P. *et al.* (2009) CGH-normaliter: an iterative strategy to enhance normalization of array CGH data with imbalanced aberrations. *BMC Genomics*, **10** [Epub ahead of print, doi:10.1186/1471-2105-9-93].

Venkatraman,E.S. and Olshen,A.B. (2007) A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics*, **23**, 657–663.