Bioinformatics Advance Access published June 22, 2007

BIOINFORMATICS

Vol. 00 no. 00 2007
Pages 1–2

beadarray: R classes and methods for Illumina bead-based data

Mark J. Dunning [a]*, Mike L. Smith [a], Matthew E. Ritchie [a] and Simon Tavaré [a]

[a]Department of Oncology, University of Cambridge, CRUK Cambridge Research Institute, Li Ka Shing Centre, Robinson Way, Cambridge, CB2 0RE, United Kingdom

Associate Editor: Prof. Alfonso Valencia

ABSTRACT

Summary

The R/Bioconductor package *beadarray* allows raw data from Illumina experiments to be read and stored in convenient R classes. Users are free to choose between various methods of image processing, background correction and normalisation in their analysis rather than using the defaults in Illumina's proprietary software. The package also allows quality assessment to be carried out on the raw data. The data can then be summarised and stored in a format which can be used by other R/Bioconductor packages to perform downstream analyses. Summarised data processed by Illumina's BeadStudio software can also be read and analysed in the same manner.

Availability: The *beadarray* package is available from the Bioconductor web page at www.bioconductor.org. A user's guide and example data sets are provided with the package.

Contact: md392@cam.ac.uk

1 INTRODUCTION

Illumina have created an alternative microarray technology (BeadArray) based on randomly arranged beads. A specific oligonucleotide sequence is assigned to each *bead type*, which is replicated about 30 times on an array. A series of decoding hybridisations is used to identify every bead (Gunderson et al., 2004). The high degree of replication makes robust measurements for each bead type possible. We have previously used the *beadarray* package to demonstrate some of the statistical properties of BeadArrays (Dunning et al., 2006).

BeadArrays can be used for various applications, including gene expression studies (Kuhn et al., 2004), SNP genotyping, methylation profiling and array CGH. Arrays are processed in parallel as a SAM (Sentrix Array Matrix) or BeadChip. A SAM is a plate of 96 uniquely prepared hexagonal BeadArrays, each of which contains around 1500 bead types. The BeadChip technology comprises a series of rectangular strips on a slide, each strip containing about 24,000 bead types. For example, there are six pairs of strips on each Human-6 BeadChip. BeadArrays can be one or two colour depending on the application.

After hybridisation and washing, each array is scanned by Illumina scanning software (BeadScan) to produce a TIFF image. The latest version of BeadScan can also output a text file giving the identity and position of each bead on the array.

The Bioconductor project (Gentleman et al., 2004) is an online repository of open source software written using the R programming language. The project aims to provide a range of statistical and graphical tools for analysing genomics data. *beadarray* was the first Bioconductor package written specifically for Illumina data. Other packages such as *lumi*, *BeadExplorer* and *beadarraySNP* are now available. IlluminaGUI (Eggle et al., 2007) provides a graphical interface allowing summarised Illumina data to be analysed via selected Bioconductor packages.

*to whom correspondence should be addressed

2 DESCRIPTION

2.1 Bead level data

We refer to the collection of TIFF images and text files as the *bead level data* for an experiment. Bead level data can be read into memory using the *readIllumina* function. By default, this function will find all images and text files within the current working directory and apply the image processing steps used by Illumina (Kuhn et al., 2004). Other image processing options are also available. Users can also choose between different background correction methods. Due to the random nature of the technology, each array has a variable number of rows of intensity data. An R environment object is used to store this information in a memory efficient manner. The same environment may be used for single or two channel data from SAMs or BeadChips.

Typical quality assessment for microarrays involves looking for systematic differences between arrays within an experiment as well as spatial artifacts on each array. Boxplots, density plots and image plots can be generated automatically and summarised in an HTML report and used to identify outlier arrays. Figure 1 shows some of the bead level plotting options available in the *beadarray* package. These plots can be used to identify problematic arrays (A) or to view the raw bead intensities for particular genes (B) or SNPs (C).

2.2 Bead Summary Data

After quality assessment has taken place, the replicate beads on each array are summarised to give an average intensity value and variance for each bead type. We refer to this as the *bead summary data* for an experiment. We use the Illumina default method for calculating these summary values, by removing outliers greater than 3 median absolute deviations (MADs) from the median and calculating the mean and variance of the remaining beads. Different MAD cut-offs are possible and users may define their own functions to obtain robust summary values and choose between calculating their summary values on the original or logged scale. Alternatively, the bead summary output produced by BeadStudio may be used.

The contents of the class object used to store bead summary data depend on the type of Illumina technology being analysed. The

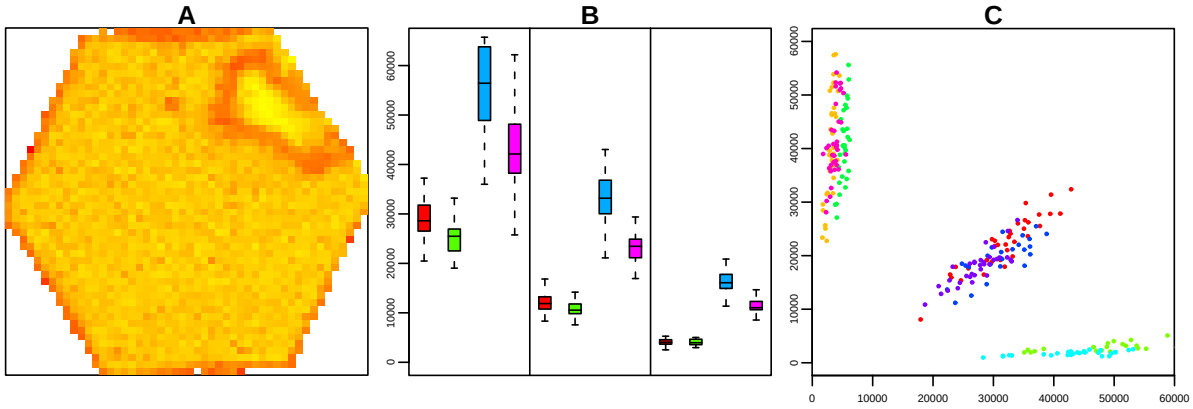

Fig. 1. **(A)** Image plot showing the variation in $\log_2$ foreground intensity across the surface of a BeadArray. An obvious spatial effect which can only be identified from bead level data can be seen. **(B)** Using bead level information to assess the distributions of particular bead types across a set of arrays. Here we show the bead level intensities (y axis) of four bead types across three arrays from a spike-in experiment in which the concentration of each probe decreases on each array. **(C)** Plot of the raw data for allele A versus allele B for a particular SNP across eight arrays. Each colour denotes a different array. Using the bead level data allows three distinct genotypes (AA, AB, BB) to be identified.

class is an extension of the *eSet* class, written by the Bioconductor core development team and designed to store and manipulate data from high-throughput genomic experiments. Using a common class to store data means that *beadarray* users can interact with other Bioconductor packages. Examples include using *affy* (Gautier et al., 2004) to normalise the data or *limma* (Smyth, 2005) to find differentially expressed genes.

3 DISCUSSION

BeadArray technology will become increasingly popular and we anticipate that *beadarray* will become an important tool in the analysis of Illumina data. The main benefit of *beadarray* is its flexibility. The package offers a variety of image processing and background correction methods, rather than the default method used by Illumina, and a choice of scale at all stages of the analysis. Having access to the raw data provides scope for users to develop their own analysis methods, such as genotype calling. Also, *beadarray* is able to read and process raw data from gene expression, SNP genotyping or methylation arrays. All other Bioconductor packages for Illumina analysis only handle summarised data from specific platforms.

beadarray allows the analysis of Illumina data to be performed entirely in R and on any operating system. A simple script can be used to read raw data, produce diagnostic plots and create summarised data. Therefore, the package is amenable for use in core facilities producing large numbers of arrays where processing data using BeadStudio may not be feasible and reproducible research is required.

Users should be aware that using bead level data requires large amounts of computer memory. For example, the raw data for a Human-6 BeadChip consists of twelve 80Mb TIFF images and twelve 40Mb text files. Reading these data into memory and analysing them using *beadarray* currently requires at least 2Gb of RAM and uses around 1 Gb of disk space.

4 ACKNOWLEDGEMENTS

We thank Martin Morgan and Vince Carey for useful advice on the classes used within the package; Roman Sasik for providing code to read TIFF images; Gary Nunn for the spike-in data; Semyon Krugylak for advice on Illumina algorithms; Matthew Forrest and Barbara Stranger for the example data used in Figure 1; Inma Spiteri for the example data in the package and Natalie Thorne and Andy Lynch for useful discussions. The authors were supported in part by grants from the MRC (MJD), CRUK (MLS, ST) and the Isaac Newton Trust (MER).

REFERENCES

Dunning MJ, *et al.* Quality control and low-level statistical analysis of Illumina BeadArrays. *Revstat*, 4:1–30, 2006.

Eggle D, *et al.* IlluminaGUI: Graphical User Interface for analyzing gene expression data generated on the Illumina platform. *Bioinformatics*, 2007.

Gautier L, *et al.* affy–analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics*, 20(3):307–15, 2004.

Gentleman RC, *et al.* Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol*, 5:R80, 2004.

Gunderson KL, *et al.* Decoding randomly ordered DNA arrays. *Genome Res*, 14:870–877, 2004.

Kuhn K, *et al.* A novel, high-performance random array platform for quantitative gene expression profiling. *Genome Res*, 14:2347–2356, 2004.

Smyth GK. Limma: linear models for microarray data. In Gentleman R, Carey V, Huber W, *et al*. *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, pages 397–420. Springer, New York, 2005.