GGtools: analysis of genetics of gene expression in bioconductor

Vincent J. Carey1,*, Martin Morgan2, Seth Falcon2, Ross Lazarus1 and Robert Gentleman2

1Channing Laboratory, Brigham and Women’s Hospital, 75 Francis Street, Boston 02115, USA and 2Program in Computational Biology, Fred Hutchinson Cancer Research Center, 1100 Fairview Avenue N, Seattle, WA 02115, USA

Received on August 18, 2006; revised on November 7, 2006; accepted on December 5, 2006

Advance Access publication December 8, 2006

Associate Editor: Keith A. Crandall

ABSTRACT

Summary: This paper reviews the central concepts and implementation of data structures and methods for studying genetics of gene expression with the GGtools package of Bioconductor. Illustration with a HapMap expression dataset is provided.

Availability: Package GGtools is part of Bioconductor 1.9 (http://bioconductor.org). Open source with Artistic License.

Contact: stvjc@channing.harvard.edu

INTRODUCTION

When each member of a group of individuals is subjected to both SNP array genotyping and microarray-based gene transcript profiling, joint analysis of genetic sequence variation and variation in gene expression becomes feasible. The fundamental paper on this approach (termed ‘genetical genomics’) is Jansen and Nap (2001); see also Li and Burmeister (2005). Cheung et al. (2005) is an essential example for the software developed here. Key topics addressed by these authors include (1) assessment of existence and determination, and (2) assessment of the distribution of locations of cis determinants between flanking regions that are either 5′ or 3′ to the gene. This paper describes GGtools, an R/Bioconductor package that facilitates (1) creation of a unified representation of expression array and SNP array results for R/Bioconductor (Gentleman et al., 2005), (2) development of convenience interfaces that simplify execution of relevant statistical analyses, (3) establishment of interoperability with members of a growing family of genetics-related software packages provided for R at CRAN (http://lib.stat.cmu.edu/R/CRAN/src/contrib/Views/Genetics.html).

DESCRIPTION

GGtools data structures

Three basic data structures are used. They are composed using the S4 formalism for object-oriented programming in R (Chambers, 1998).

Class racExSet is used to manage combinations of SNP genotyping and expression profiling results obtained on the same individuals. The prefix rac denotes the ‘rare allele count’ encoding of genotype. SNP data are held in an S × N matrix, where S is the number of SNPs (each identified by dbSNP rs number) and N is the number of samples. The ij element of the matrix is the number of copies of the minor allele (relative to a population) present for SNP i in sample j. Two S-vectors are also present encoding the SNP alleles in the form ‘A/B’ (no ordering assumed), and identifying the minor allele. Expression data are held in a G × N matrix, where G is the number of reporter elements on the expression array platform. The package includes examples of racExSet objects created to represent results of human and mouse experiments. Presently, human data are provided in a chromosome-specific format to reduce data object size; the example mouse data are provided in a whole-genome format.

Class snpMeta encodes rs number, position and strand orientation for each SNP on a chromosome-by-chromosome basis.

Class snpScreenResult is used to manage the results of statistical model fits over sequences of SNPs. An R list of returned fit objects (e.g. instances of S3 class lm, if least squares regression is in use) is maintained, along with information on names and physical locations of SNPs tested for association with expression on a given gene.

GGtools methods

There are three basic activities supported at present.

Data import Given a matrix of expression data, an associated phenoData instance describing samples, a matrix of rare allele count data with vectors identifying the biallelic content and rare base for each SNP, and an instance of Bioconductor’s MIAME experiment metadata class, the function make_racExSet will bundle together a unified object of class racExSet. The SNP-related genotyping counts and metadata can be created from gzipped HapMapConsortium (2003) chromosome-specific archive files using Hapworkflow; a helper function for creating genotype components from Wellcome Trust archives on inbred mouse strains is called INBREDSworkflow.

Interrogation Expression values can be queried using the exprs method with subscripting on probe set name and/or sample identifier. A snps method works similarly.

*To whom correspondence should be addressed.
Flexible screening  The `snpScreen` method operates on instances of the `racExSet` class. It is important to allow variations on the additive genetic model commonly in use. The interface takes the form `snpScreen(x, m, g, f, mod)`, where `x` is an instance of the `racExSet` class, `m` is a `snpMeta` instance, providing location information, `g` is a gene symbol object denoting the gene for which expression is modeled, `f` is a formula template (either `.` or `factor(.)` to choose between an additive regression model and a more general ANOVA), and `mod` is an R model fitting function that works with formula and data frame inputs. A special function `fastAGM` is available for binding to `mod` for rapid fitting of the additive genetic model for SNPs with complete data.

EXAMPLES
The following code can be used to approximately replicate the finding of Cheung *et al.* (2005) Table 1 regarding gene IRF5 on chromosome 7. The second statement generates Figure 1.

```r
sCPNE1 = snpScreen(chr20Gdem, chr20meta, genesym('CPNE1'), ..., fastAGM)
plot_mlp(sCPNE1, chr20meta, plotf=plot)
```

On a 1.5 GHz G4 powerbook with 1.25 GB RAM, this screen processed ~ 1400 SNP/s. The SNP with lowest *P*-value in this screen was rs6058296, with a regression *P*-value of $1.4 \times 10^{-14}$. This SNP is separated from the one identified in Table 1 of Cheung *et al.* (2005) by only 33 703 bases. A whole-genome screen of the sample of BXD mice collected in GEO GSE2031 (see lower panel of Fig. 1) is executed with similar code and processes 2600 SNP/s.

DISCUSSION
When infrastructure for managing results of multiple high-throughput experiments is carefully and portably designed, it becomes possible to answer focused questions very simply. By allowing all the results of a genetics of gene expression experiment to be bound into a single R language variable, methods for screening and visualization of genotype-expression relationships can have simple interfaces. Ongoing work addresses performance improvements (particularly parallelization) and more flexible statistical inference with such data.

ACKNOWLEDGEMENTS

Conflict of Interest: none declared.

REFERENCES