R and Bioconductor solutions for alternative splicing detection

*Tzulip Phang**

Department of Medicine, University of Colorado Denver School of Medicine, Denver, CO, USA
**Correspondence to:* Tel: +1 303 724 6057; E-mail: tzu.phang@ucdenver.edu

Date received (in revised form): 9th September 2009

Abstract

The detection of alternative splicing using microarray technology involves multiple computational steps: normalisation, filtering, detection and visualisation. In this review, these analyses are approached using the R and Bioconductor open-source computation solution. There is some discussion on how to integrate different Bioconductor packages, and some of their major features are demonstrated. In addition, the Xmap Genome Browser is incorporated as an integral part of the analysis and annotation workflow.

Keywords: *software, R, Bioconductor, analysis, programming, alternative splicing*

Introduction

Alternative splicing (AS) is a natural biological process that allows genetic materials from a single locus to produce more than one transcript. The functionality of the resulting isoforms can be grossly different, and could potentially change the landscape of the gene expression network.[1] Furthermore, it has been shown that more than 70 per cent of transcripts in the genome undergo AS.[2] As a result, many human diseases have been directly and indirectly affected by a deficiency in these splice forms.[3] The importance of this phenomenon has triggered the development of biological assays to measure the anomaly, and microarray has become one of the common tools.

Several issues arise in analysing exon microarray data. First, the algorithm developed to detect AS has to account for different AS types.[4] Secondly, unlike gene-centric expression arrays, where most genes are labelled by one identifier (ID), exon arrays involve tracking all known and predicted exons for a gene. As genomic annotation becomes more sophisticated, this information could change frequently. Therefore, a comprehensive database management system is required as a back-end support for the exon array analysis. Lastly, the complex task of integrating the statistical analysis, database support and visualisation into a single software system is unquestionably challenging. Furthermore, these integrations should take future algorithm development into confederation.

In this review, these issues will be discussed, and the R statistical software system introduced as an integrated environment for AS detection analysis. A series of R packages for performing the analysis workflow with example codes will be outlined.

A brief review of exon array design

This review discusses tools used to analyse the Affymetrix GeneChip 1.0 ST array. At the time of writing, the manufacturer has produced arrays for human, mouse and rat species. Briefly, the array design consists of 5.4 million 5-μm probes, which are assembled into 1.4 million probesets. The majority of these probesets are represented by four overlapping probes. In most cases, each probeset represents a single exon. Sometimes, more than one probeset might be necessary to represent a single exon.[5] These probesets can be broken down into three increasingly comprehensive annotated categories, collectively known as the ‘core’, ‘extended’ and ‘full’ sets. At one extreme, the ‘core’ set consists of the probesets with the highest annotation confidence. At the other extreme, the ‘full’ set consists of the more computationally derived probesets, as well as probesets that represent non-exonic regions, such as intronic and intergenic regions.

A brief overview of R and Bioconductor

R (http://www.r-project.org) is a powerful, but yet flexible, statistical computing programming environment. It was developed by Robert Gentleman and Ross Ihaka and is highly regarded by the computational communities. Its object-orientation programming scheme has made algorithm development easy and flexible and has attracted a huge developer community. These packages are freely available at the CRAN section of the website. R is platform independent, and works on all major computer operating systems. In 2001, Robert Gentleman established the Bioconductor consortium[6] (http://www.bioconductor.org) as a repository and distribution centre for genomic computational tools. The consortium has been working hard to keep up with genomic technology development, ranging from proteomic, microarray and the more recent next generation sequencing analyses. Bioconductor has played a major role in the successes of genomic analysis, and continues to contribute to this cause, being cited in hundreds of peer-reviewed publications. Readers who wish to get a quick start on R and Bioconductor are encouraged to visit Thomas Girke’s website for an extensive tutorial.[7]

Workflow for alternative splicing detection using R packages

In general, exon array analysis workflow is broken down into four major steps: data processing and normalisation; genomic annotation and data filtering; AS detection; and AS visualisation. R and Bioconductor statistical packages will be used for each step described below. We will provide sample code when appropriate.

Data processing and normalisation

Due to its popularity, summarisation and normalisation methods for oligonucleotide array have been thoroughly researched.[8] Among the most popular methods are robust multichip average (RMA); probe logarithmic intensity error (PLIER); and GC robust multichip average (GCRMA).[9–11] The Bioconductor packages ‘oligo’ and ‘affy’ are the foundation for importing and processing oligonucleotide microarrays such as the exon array used in this review. The successful processing and normalisation of exon array chips rely on the Platform Design (pd) file. Such files contain the mapping information that assigns the probes to their corresponding probesets. A collection of available pd files can be downloaded from the ‘Download’ and ‘Metadata’ sections of the Bioconductor website.

Here, the procedures for loading the packages and importing the dataset are illustrated. The ‘read.cellfiles’ function requires users to specify the pd file for the chip types using the ‘pkgname’ argument.

```
> library(‘oligo’)
> library(‘pd.huex.1.0.st.v2’)
> celFiles = list.celfiles (‘/path/to/my/celfiles’, full.
  name = TRUE)
> raw = read.cellfiles (celFiles, pkgname = ‘pd.huex.
  1.0.st.v2’)
```

Once the dataset is imported into the R session as an ‘ExpressionSet’ object, it can be normalised using multiple approaches. Here, we illustrate how the dataset can be normalised using three different methods: RMA, GCRMA and PLIER, respectively.

```
R> eset = rma(raw)
R> eset = gcrma(raw)
R> eset = justPlier(raw)
```

Recently, Gaidatzis *et al.*[12] discovered a systematic relationship between gene expression and AS. The authors showed that detection of AS increased significantly as the differential expression of the respective gene increases. This caused an overestimation of AS for the dataset. The authors developed an algorithm, COrrected Splicing Indices for Exon array (COSIE), to correct for this effect. The method was based on a positional dinucleotide modelling approach. The COSIE package can be downloaded from the authors' website (http://www.fmi.ch/groups/gbioinfo). The function from the COSIE package takes the 'ExpressionSet' object as its input and corrected expression values as its output. Here is an example of COSIE codes:

```
R> source('cosie.R')
R> cosieOut = cosie(eset, '/path/to/cosie/output/
     file.txt')
```

Genomic annotation and data filtering

The exon array was designed to measure signals from exonic, intronic and intergenic regions. This ability increases the likelihood of discovering novel splice forms and regulatory elements. Keeping track of these genomic regions can be a daunting task, however, and entails the need for a database management system (DBMS). Okoniewski *et al.*[13] have built a database, Xmap, to capture these gene region entities. From now on, the different representations of a gene (exon, intron, intergenic, etc) will be referred to as gene entities. The database uses the Ensembl ID system to map each probeset to its corresponding gene, transcript, exon and other Ensembl representation (http://www.ensembl.org). For example, the exon array ID, 3564255, is mapped to the gene ID ENSG00000100505, and three transcript IDs, ENST0000029355, ENST00000338969 and ENST00000360392. A browser interface was developed to access the database (http://xmap.picr.man.ac.uk). In addition to the website, an R package, 'exonmap', was developed to provide direct access to the database using the R command line interface. The package provides three types of programmatic routine: translational, annotation and filtering.

The translational routine provides functionalities that perform conversions between different gene entities, such as probe, probeset, exon, transcript, gene, symbol and sequence. These functions assume the form X.to.Y, where X and Y represent different combinations of the entities. For example, the function 'probeset.to.gene' converts a probeset ID to the gene ID it represents. Please refer to the function's help file for the full listing of all the X.to.Y functions. Except for the probeset IDs, all other entities are represented by the Ensembl ID system. Because of this integration, it is relatively easy to extract extended annotation using the Ensembl database. One downside of using 'exonmap', however, is the requirement for installing a local copy of the Ensembl database onto your computer, which can take up a substantial amount of disk space.

The annotation routine is used to attach biological meanings to the different gene entities. At the time of writing, users can perform annotation at three levels: probeset, exon and gene. For example, the function 'probeset.details' takes a probeset ID as an argument and returns a predefined set of annotations for that probeset. This annotation routine returns predefined annotation information for the entities and cannot easily be extended. To perform a more customised annotation, Bioconductor offers the 'biomaRt' package, which provides a direct interface to the Ensemble database. The 'getBM' function from 'biomaRt' takes four input parameters: 'Attributes', 'Filters', 'Values' and 'Mart'. The 'Attributes' are the list of desired annotations to be returned. The 'Values' are the actual input IDs designated by the 'Filters' argument. Finally, 'Mart' indicates the database selected at the time of query. Here is a sample code for annotating the exon array ID, 3564255, using the 'genBM' function. First, we need to select the database, as well as the species, for this analysis. In this case, we have selected to use the Ensemble database for the human species. Then, we use 'getBM' to annotate the 'Affymetrix exon array' ID with two Ensembl IDs.

```
R> library(biomaRt)
R> ensemble = useMart('ensembl', dataset =
     'hsapiens_gene_ensembl')
R> annotation = getBM(Attribute = c('ensembl_
     gene_id', 'ensembl_transcript_id'),
Filters = 'affy_huex_1_0_st_v2',
    Values = '3564255',
    Mart = ensemble)
```

There is also a web version of the Biomart database (http://www.biomart.org), where users can perform point and click database searches. The website is a great companion to the 'biomaRt' package. Users can perform a prototype search before finalising the customised profiles with the 'getBM' function. The precise naming scheme for the 'Attributes' and 'Filters' arguments is mandatory, and can be listed using the 'listAttributes' and 'listFilters' functions.

Finally, the Filtering routine from 'exonmap' serves two purposes: verification and filtering. The verification utility takes the form is.X and is used to verify the identity of a probeset as either exonic, intronic, intergenic or multitarget. This is especially useful for identifying multi-targeted probesets using the 'is.multitarget' function. The removal of these entities will reduce the chances of selecting a false-positive AS. Lastly, the two filtering functions are 'select.probewise' and 'exclude.probewise', which will either select or remove probesets that belong to the selected entity from further consideration.

Alternative splicing detection

The simplest way to detect AS is to use the splicing index (SI) method.[14] The SI is calculated using the log-ratio of the normalised exon level from the comparison group, where the normalised exon level can be calculated by dividing the exon signal by its estimated gene level. One can think of SI as the equivalent of fold change in gene expression microarray analysis. The major pitfall of fold change is that no statistical confidence is computed. Therefore, better approaches, such as ANalysis Of Splice VAriation (ANOSVA), were developed to apply statistical stringency to changes in SI.[15] ANOSVA was based on modelling the interaction effect between the exon and the gene, where, under the null hypothesis, all interaction terms in an exon and a gene in the two-way ANOVA are not significant, and therefore no exon or probesets stand out. By contrast, under the alternative hypothesis, the interaction term is significant, and therefore characteristic of AS. This basic model was further improved by other authors.[16–18] The 'exonmap' package offers two functions for detecting AS: 'si' and 'splanova', which perform the SI and ANOVA modelling analyses, respectively. Please refer to the 'exonmap' vignette for details of their use.

Since exon array analysis deals with a massive amount of data and the current Bioconductor modules require them to be loaded into memory at once, a regular 32-bit computer might be insufficient for this purpose. Instead, users are encouraged to use a 64-bit computer system with sufficient RAM memory, preferably 8 megabytes (MB) or higher. Alternatively, Bengtsson *et al.*[19] have developed the 'aroma.affymetrix' package which has solved this memory problem by using persistent memory technology. These authors claim that the analysis can be done with just 1 MB of RAM, and can process up to 1,000 arrays. Using the 'aroma.affymetrix' framework, Purdom *et al.*[18] have implemented an AS detection algorithm, FIRMA (Finding Isoforms using Robust Multichip Analysis). In this algorithm, the authors modified the popular RMA normalisation method[9] by building an additive model to study the behaviour of an exon relative to its gene expression. Similar to ANOSVA, any deviation from the fitted behaviour would imply AS. The tutorial materials for the 'aroma.affymetrix' package can be found in a Google user group (http://www.braju.com/R/aroma.affymetrix).

AS visualisation

Visualisation is important in AS detection. First, it shows the distribution ratio of probesets per exon. The number of probesets used to represent an exon reflects the exon's size and complexity. Secondly, a visual inspection of the exon's expression value ensures the accuracy of the AS prediction algorithm, and therefore reduces false positives. Thirdly, information without content is meaningless. Therefore, attaching genomic knowledge to the visualisation process allows further evaluation of the AS candidates.

The most comprehensive visualisation tool for the Affymetrix exon array is the Xmap Genome Browser (http://xmap.picr.man.ac.uk). The website uses Google's map navigation technology to provide a smooth scanning of gene regions. The back-end Ensembl database support enables the display of diverse gene information, such as the multiple transcripts that represent different splice forms. To add expression information onto the display, Bioconductor offers the 'XMapBridge' package. The package enables a seamless connection between R and the Xmap Genome Browser via Java technology.

Conclusion

The R statistical software and the Bioconductor consortium offer a wealth of solutions for AS detection. With the support of developers around the world, they provide new and improved algorithms in all areas of exon array analysis. Overall, the 'exonmap' package offers the most comprehensive analysis routines. The Xmap Genome Browser presents a beautiful, yet practical, visualisation utility for the exon array analysis. The website is supported by the Ensembl database and provides a comprehensive annotation solution for AS candidates.

References

1. Pio, R. and Montuenga, L.M. (2009), 'Alternative splicing in lung cancer', *J. Thorac. Oncol.* Vol. 4, pp. 674–678.
2. Johnson, J.M., Castle, J., Garrett-Engele, P., Kan, Z. *et al.* (2003), 'Genome-wide survey of human alternative pre-mRNA splicing with exon junction microarrays', *Science* Vol. 302, pp. 2141–2144.
3. Wang, G.S. and Cooper, T.A. (2007), 'Splicing in disease: Disruption of the splicing code and the decoding machinery', *Nat. Rev. Genet.* Vol. 8, pp. 749–761.
4. Black, D.L. (2003), 'Mechanisms of alternative pre-messenger RNA splicing', *Ann. Rev. Biochem.* Vol. 72, pp. 291–336.
5. Affymetrix_Inc. (2005), 'Affymetrix White Papers: Exon Probeset Annotations and Transcript Cluster Grouping v1.0'. Available from http://www.affymetrix.com/support/technical/whitepapers.affx (accessed October 2009).
6. Gentleman, R.C., Carey, V.J., Bates, D.M., Bolstad, B. *et al.* (2004), 'Bioconductor: Open software development for computational biology and bioinformatics', *Genome Biol.* Vol. 5, p. R80.
7. Girke, T. (2009), 'R and Bioconductor Tutorials'. Available from http://faculty.ucr.edu/~tgirke/Documents/R_BioCond/R_BioCondManual.html.
8. Irizarry, R.A., Wu, Z. and Jaffee, H.A. (2006), 'Comparison of Affymetrix GeneChip expression measures', *Bioinformatics* Vol. 22, pp. 789–794.
9. Irizarry, R.A., Hobbs, B., Collin, F., Beazer-Barclay, Y.D. *et al.* (2003), 'Exploration, normalization, and summaries of high density oligonucleotide array probe level data', *Biostatistics* Vol. 4, pp. 249–264.
10. Wu, Z., Irizarry, R.A., Gentleman, R., Murillo, F.M. *et al.* (2004), 'A model based background adjustment for oligonucleotide expression arrays', *J. Am. Stat. Assoc.* Vol. 99, pp. 909–917.
11. Hubbell, E., Liu, W.M. and Mei, R. (2002), 'Robust estimators for expression analysis', *Bioinformatics* Vol. 18, pp. 1585–1992.
12. Gaidatzis, D., Jacobeit, K., Oakeley, E.J. and Stadler, M.B. (2009), 'Overestimation of alternative splicing caused by variable probe characteristics in exon arrays', *Nucleic Acids Res.* Vol. 1, pp. 1–10.
13. Okoniewski, M.J., Yates, T., Dibben, S. and Miller, C.J. (2007), 'An annotation infrastructure for the analysis and interpretation of Affymetrix exon array data', *Genome Biol.* Vol. 8, p. R79.
14. Affymetrix_Inc. (2008), 'Affymetrix White Papers: Identifying and Validating Alternative Splicing Events'. Available from http://www.affymetrix.com/support/technical/technotes/id_altsplicingevents_technote.pdf (accessed December 2009).
15. Cline, M.S., Blume, J., Cawley, S., Clark, T.A. *et al.* (2005), 'ANOSVA: A statistical method for detecting splice variation from expression data', *Bioinformatics* Vol. 21 (Suppl. 1), pp. i107–i115.
16. Xing, Y., Stoilov, P., Kapur, K., Han, A. *et al.* (2008), 'MADS: A new and improved method for analysis of differential alternative splicing by exon-tiling microarrays', *RNA* Vol. 14, pp. 1470–1479.
17. Li, C. and Wong, W.H. (2001), 'Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection', *Proc. Natl. Acad. Sci. USA*, Vol. 98, pp. 31–36.
18. Purdom, E., Simpson, K.M., Robinson, M.D., Conboy, J.G. *et al.* (2008), 'FIRMA: A method for detection of alternative splicing from exon array data', *Bioinformatics* Vol. 24, pp. 1707–1714.
19. Bengtsson, H., Irizarry, R., Carvalho, B. and Speed, T.P. (2008), 'Estimation and assessment of raw copy numbers at the single locus level', *Bioinformatics* Vol. 24, pp. 759–767.