PathwaySplice: pathway analysis for alternative splicing in RNA-seq datasets that accounts for different number of gene features

Aimin Yan, Xi Chen, Lily Wang

2017-09-28

1 Introduction

In alternative splicing ananlysis of RNASeq data, one popular approach is to first identify gene features (e.g. exons or junctions) significantly associated with splicing using methods such as DEXSeq (Anders, Reyes, and Huber 2012) or JunctionSeq (Hartley and Mullikin 2016), and then perform pathway analysis based on the list of genes associated with the significant gene features.

For DEXSeq results, we use *gene features* to refers to non-overlapping exon counting bins (Anders, Reyes, and Huber 2012, Figure 1), while for JunctionSeq results, *gene features* refers to non-overlapping exon or splicing junction counting bins.

A major challenge is that without explicit adjustment, pathways analysis would be biased toward pathways that include genes with a large number of gene features, because these genes are more likely to be selected as "significant genes" in pathway analysis.

PathwaySplice is an R package that falicitate the folowing analysis:

1. Performing pathway analysis that explicitly adjusts for the number of exons or junctions associated with each gene;
2. Visualizing selection bias due to different number of exons or junctions for each gene and formally tests for presence of bias using logistic regression;
3. Supporting gene sets based on the Gene Ontology terms, as well as more broadly defined gene sets (e.g. MSigDB) or user defined gene sets;
4. Identifing the significant genes driving pathway significance and
5. Organizing significant pathways with an enrichment map, where pathways with large number of overlapping genes are grouped together in a network graph.

2 Quick start on using PathwaySplice

After installation, the PathwaySplice package can be loaded into R using:

```
library(PathwaySplice)
```

The latest version can also be installed by

```
library(devtools)
install_github("SCCC-BBC/PathwaySplice",ref = 'development')
```

The input file of PathwaySplice are p-values for multiple gene features associated with each gene. This information can be obtained from DEXSeq (Anders, Reyes, and Huber 2012) or JunctionSeq (Hartley and Mullikin 2016) output files. As an example, PathwaySplice includes a feature based dataset within the package, based on a RNASeq study of CD34+ cells from myelodysplastic syndrome (MDS) patients with SF3B1 mutations (Dolatshad, et al., 2015). This dataset was downloaded from GEO database (GSE63569), we selected a random subset of 5000 genes here for demonstration.

The example dataset can be loaded directly:

```
data(featureBasedData)
head (featureBasedData)
```

```
#>            geneID countbinID     pvalue
#> 1 ENSG00000279928       E003 0.19792636
#> 2 ENSG00000279457       E002 0.55363528
#> 3 ENSG00000279457       E003 0.12308986
#> 4 ENSG00000279457       E004 0.11887268
#> 5 ENSG00000279457       E005 0.03981720
#> 6 ENSG00000279457       E006 0.07570489
```

Next the **makeGeneTable** function can be used to convert it to a gene based table.

```
gene.based.table <- makeGeneTable(featureBasedData)
head(gene.based.table)
```

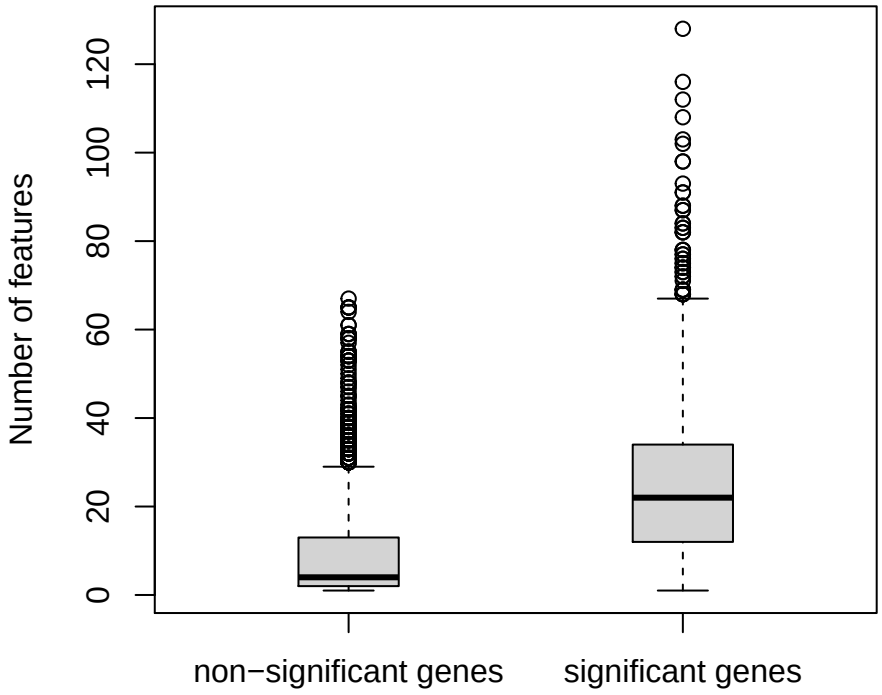
```
#>            geneID geneWisePvalue numFeature         fdr sig.gene
#> 1 ENSG00000000938   4.076135e-02         18 0.104302337        1
#> 2 ENSG00000001497   1.257442e-05         20 0.002168003        1
#> 3 ENSG00000002745   2.477443e-01          1 0.325804471        0
#> 4 ENSG00000002919   1.363500e-03         27 0.024035390        1
#> 5 ENSG00000003393   1.554129e-02         51 0.066815537        1
#> 6 ENSG00000003989   5.960507e-01          1 0.659445796        0
```

Here `geneWisePvalue` is simply the lowest feature based p-value for the gene, `numFeature` is number of features for the gene, `fdr` is false discovery rate for `genewisePvalue`, `sig.gene` indicates if a gene is significant.

To assess selection bias, i.e. whether gene with more features are more likely to be selected as significant genes, **lrTestBias** function fits a logistic regression with `logit (sig.gene) ~ numFeature`

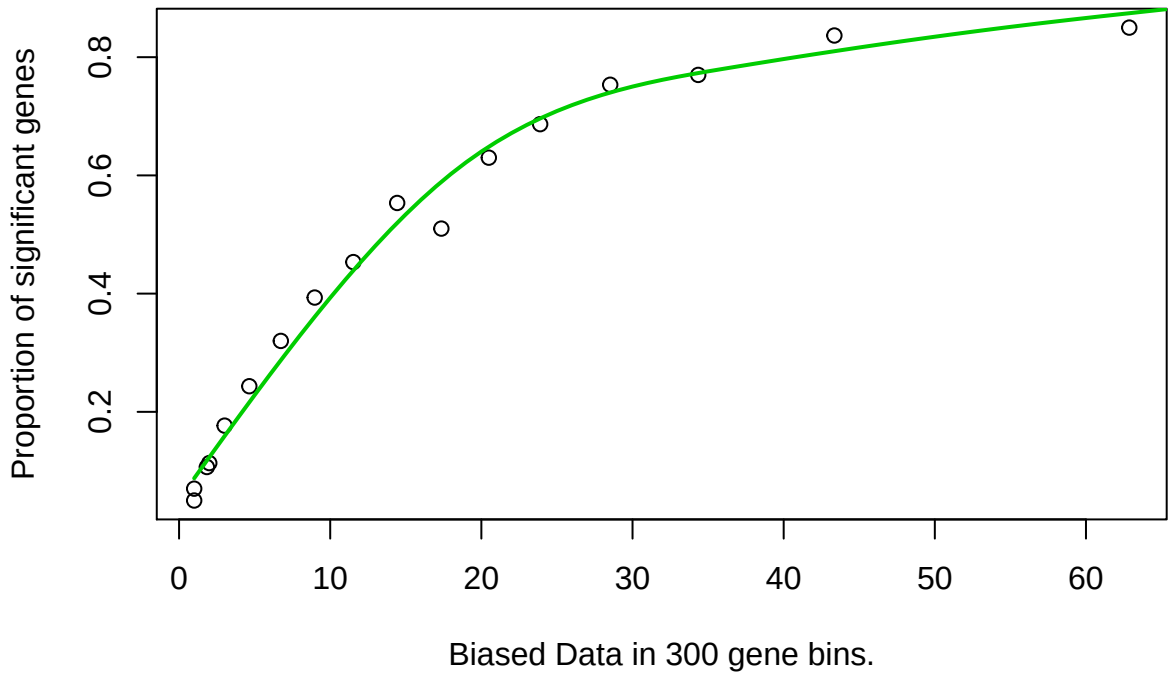
```
lrTestBias(gene.based.table,boxplot.width=0.3)
```

```
#> [1] "P-value from logistic regression is 3.98e-205"
```



To perform pathway analysis that adjusts for the number of gene features, we use the **runPathwangSplice** function, which implements the methodology described in (Young et al. 2010). **runPathwangSplice** returns a tibble dataset with statistical significance of the pathway (`over_represented_pvalue`), as well as the significant genes that drives pathway significance (`SIGgene_ensembl` and `SIGgene_symbol`). An additional bias plot that visualizes the relationship between the proportion of significant genes and the mean number of gene features within gene bins is also generated.

```
result.adjusted <- runPathwaySplice(gene.based.table,genome='hg19',
                        id='ensGene',
                        test.cats=c('GO:BP'),
                        go.size.limit=c(5,30),method='Wallenius')
```




```
head(result.adjusted)
```

```
#> # A tibble: 6 x 12
#>     gene_set over_represented_pvalue under_represented_pvalue numSIGInCat
#>        <chr>                    <dbl>                     <dbl>       <int>
#> 1 GO:0043044              0.003587014                 0.9996795          17
#> 2 GO:0006323              0.003660353                 0.9994637          19
#> 3 GO:0006721              0.007491169                 0.9993466          10
#> 4 GO:0006338              0.010590403                 0.9969542          25
#> 5 GO:0016101              0.011808138                 0.9989078           9
#> 6 GO:0006720              0.011840285                 0.9981355          12
#> # ... with 8 more variables: numInCat <int>, description <chr>,
#> #   ontology <chr>, SIGgene_ensembl <list>, SIGgene_symbol <list>,
#> #   All_gene_ensembl <list>, All_gene_symbol <list>,
#> #   Ave_value_all_gene <dbl>
```

To performe pathway analysis for other user defined databases, one needs to specify the pathway database in .gmt format first and then use the `gmtGene2Cat` function before calling `pathwaySplice` function.

```
dir.name <- system.file('extdata', package='PathwaySplice')
hallmark.local.pathways <- file.path(dir.name,'h.all.v6.0.symbols.gmt.txt')
hlp <- gmtGene2Cat(hallmark.local.pathways, genomeID='hg19')
result.hallmark <- runPathwaySplice(gene.based.table,genome='hg19',id='ensGene',
```

```
                gene2cat=hlp, go.size.limit=c(5,200), method='Wallenius', binsize=20)
```

For example, the results for the MSigDB Hallmark gene sets are

```
head(result.hallmark)
```

```
#> # A tibble: 6 x 10
#>                   gene_set over_represented_pvalue
#>                      <chr>                   <dbl>
#> 1  HALLMARK_MYC_TARGETS_V1             0.004980504
#> 2      HALLMARK_MYOGENESIS             0.082904236
#> 3  HALLMARK_G2M_CHECKPOINT             0.137421221
#> 4 HALLMARK_HEME_METABOLISM             0.154964914
#> 5 HALLMARK_APICAL_JUNCTION             0.199372916
#> 6 HALLMARK_MITOTIC_SPINDLE             0.212276966
#> # ... with 8 more variables: under_represented_pvalue <dbl>,
#> #   numSIGInCat <int>, numInCat <int>, SIGgene_ensembl <list>,
#> #   SIGgene_symbol <list>, All_gene_ensembl <list>,
#> #   All_gene_symbol <list>, Ave_value_all_gene <dbl>
```

Lastly, to visualize pathway analysis results in an enrichment network, we use the **enrichmentMap** function:

```
output.file.dir <- file.path("~/PathwaySplice_output")
enmap <- enrichmentMap(result.adjusted,n=5,
                       output.file.dir=output.file.dir,
                       similarity.threshold=0.3, scaling.factor = 2)
```

In the enrichment map, the `size of the nodes` indicates the number of significant genes within the pathway. The `color of the nodes` indicates pathway significance, where smaller p-values correspond to dark red color. Pathways with Jaccard coefficient > `similarity.thereshold` will be connected on the network. The `thickness of the edges` corresponds to Jaccard similarity coefficient between the two pathways, scaled by `scaling.factor`. A file named "network.layout.for.cytoscape.gml" is generated in the "~/PathwaySplice_output" directory. This file can be used as an input file for cytoscape software(Shannon et al. 2003), which allows users to further maually adjust appearance of the generated network.

3 Reference

Anders, Simon, Alejandro Reyes, and Wolfgang Huber. 2012. "Detecting differential usage of exons from RNA-seq data." *Genome Research* 22 (10): 2008–17. doi:10.1101/gr.133744.111.

Hartley, Stephen W., and James C. Mullikin. 2016. "Detection and visualization of differential splicing in RNA-Seq data with JunctionSeq." *Nucleic Acids Research*, June. Oxford University Press, gkw501. doi:10.1093/nar/gkw501.

Shannon, P., Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. 2003. "Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks." *Genome Research* 13 (11): 2498–2504. doi:10.1101/gr.1239303.

Young, Matthew D., Matthew J. Wakefield, Gordon K. Smyth, and Alicia Oshlack. 2010. "Gene Ontology Analysis for Rna-Seq: Accounting for Selection Bias." *Genome Biology* 11 (2): R14. doi:10.1186/gb-2010-11-2-r14.