Using ReportingTools in an Analysis of RNA-seq Data

Jessica L. Larson and Christina Chaivorapol

October 2, 2012

Contents

1 Introduction

The ReportingTools package can be used with differential gene expression results from RNA-sequencing analysis. In this vignette we show how to publish output from an edgeR, Gene Ontology (GO) and/or Protein family (PFAM) analysis. In the final section we publish all our pages onto one, creating a comprehensive output page.

2 Differential expression analysis

In this section we demonstrate how to use the ReportingTools package to generate a table of differentially expressed genes. We begin by loading our library and data set. The mockRnaSeqData contains random RNA-seq output for random mouse genes.

```
> library(ReportingTools)
> data(mockRnaSeqData)
```

Next, we run edgeR to find differentially expressed genes.

```
> library(edgeR)
> conditions <- c(rep("case",3), rep("control", 3))
> d <- DGEList(counts = mockRnaSeqData, group = conditions)
> d <- calcNormFactors(d)
> d <- estimateCommonDisp(d)
> ## Get an edgeR object
> edgeR.de <- exactTest(d)
```

Now the results can be written to a report using the edgeR object. Currently, only DGEExact objects returned from the exactTest function in the edgeR package are supported.

```
> library(lattice)
> rep.theme <- reporting.theme()
> ## Change symbol colors in plots
> rep.theme$superpose.symbol$col <- c("blue", "red")
> rep.theme$superpose.symbol$fill <- c("blue", "red")
> lattice.options(default.theme = rep.theme)
> deReport <- HTMLReport(shortName = 'RNAseq_analysis_with_edgeR',
+     title = 'RNA-seq analysis of differential expression using edgeR',
+     reportDirectory = "./reports/",
+     baseUrl = "")
> ## Publish a report of the top 10 genes with p-values < 0.05 and log-fold change > 2
> publish(edgeR.de, deReport, mockRnaSeqData,
+         conditions, annotation.db = 'org.Mm.eg',
+         pvalueCutoff = .05, lfc = 2, n = 10)
> finish(deReport)
```

3 GO analysis using GOstats

In this section, we show how to use ReportingTools to write a table of GO analysis results to an html file. First we select or genes of interest and then we run the hyperGTest.

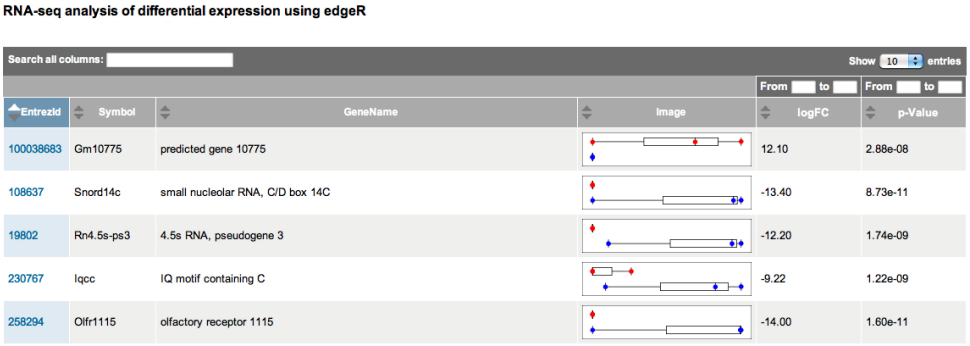


Figure 1: Resulting page created by `publish` for `edgeR.de`.

```
> library(GOstats)
> library(org.Mm.eg.db)
> tt<-topTags(edgeR.de, n = 1000, adjust.method = 'BH', sort.by = 'p.value')
> selectedIDs<-rownames(tt$table)
> universeIDs<-rownames(mockRnaSeqData)
> goParams <- new("GOHyperGParams",
+     geneIds = selectedIDs,
+     universeGeneIds = universeIDs,
+     annotation ="org.Mm.eg" ,
+     ontology = "MF",
+     pvalueCutoff = 0.01,
+     conditional = TRUE,
+     testDirection = "over")
> goResults <- hyperGTest(goParams)
```

With these results, we then make the GO report. Here we set `makePlot=TRUE` to get a large image of the relationship between our significant ontologies.

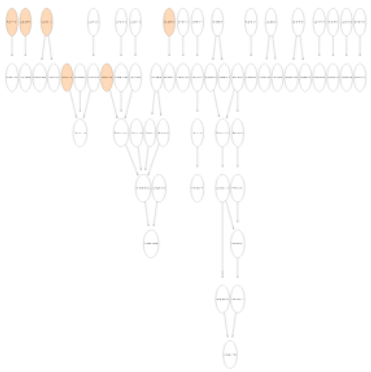
```
> goReport <- HTMLReport(shortName = 'go_analysis_rnaseq',
+         title = "GO analysis of mockRnaSeqData",
+         reportDirectory = "./reports",
+         baseUrl = "")
> publish(goResults, goReport, selectedIDs, annotation.db="org.Mm.eg",
+         pvalueCutoff= 0.05, makePlot=TRUE)
> finish(goReport)
```

4 PFAM analysis

In this section, we show how to use `ReportingTools` to write a table of PFAM analysis results to an html file. First we run the `hyperGTest` using our genes of interest from the previous section.

```
> library(Category)
> params <- new("PFAMHyperGParams",
+         geneIds= selectedIDs,
+         universeGeneIds=universeIDs,
+         annotation="org.Mm.eg",
```

GO analysis of mockRnaSeqData



Search all columns: Show 10 entries

From to From to

Accession	GO Term	Category Size	Image	Overlap	Odds Ratio	P-value
GO:0000049	tRNA binding	11		3	5.94	0.02430
GO:0003682	chromatin binding	83		10	2.19	0.02520

Figure 2: Resulting page created by `publish` for `goResults`.

PFAM analysis of mockRnaSeqData

Search all columns: Show 10 entries

PFAM ID	PFAM Term	PFAM Size	Image	Overlap	Odds Ratio	P-value
PF00057	Low-density lipoprotein receptor domain class A	16		5	7.42	0.00168
PF00413	Matrixin	9		4	13.00	0.00114
PF00433	Protein kinase C terminal domain	8		3	9.75	0.00885

Figure 3: Resulting page created by `publish` for `PFAMResults`.

```
+         pvalueCutoff= 0.01,
+         testDirection="over")
> PFAMResults <- hyperGTest(params)
```

Then we make the PFAM report.

```
> PFAMReport <- HTMLReport(shortName = 'pfam_analysis_rnaseq',
+         title = "PFAM analysis of mockRnaSeqData",
+         reportDirectory = "./reports",
+         baseUrl = "")
> publish(PFAMResults, PFAMReport, selectedIDs, annotation.db="org.Mm.eg",categorySize=5)
> finish(PFAMReport)
```

5 Putting it all together

Here, we make an index page that puts all three analyses together for easy navigation.

```
> indexPage <- HTMLReport(shortName = "indexRNASeq",
+     title = "Analysis of mockRnaSeqData",
+     reportDirectory = "./reports",
+     baseUrl = "")
> publish(c(deReport,goReport, PFAMReport), indexPage)
> finish(indexPage)
```

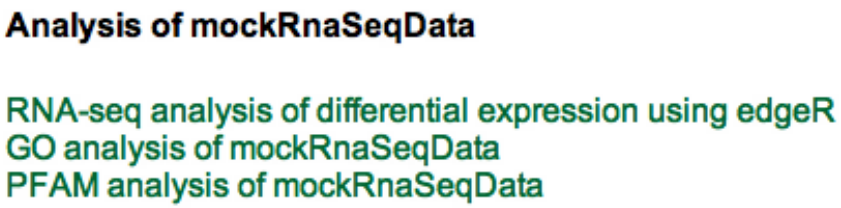


Figure 4: Resulting page created by calling `publish` on all our analysis pages.