BIOINFORMATICS APPLICATIONS NOTE
Vol. 30 no. 17 2014, pages 2517–2518
doi:10.1093/bioinformatics/btu324

Gene expression
Advance Access publication May 9, 2014

# compcodeR—an R package for benchmarking differential expression methods for RNA-seq data

Charlotte Soneson
Bioinformatics Core Facility, SIB Swiss Institute of Bioinformatics, Quartier Sorge, CH-1015 Lausanne, Switzerland

Associate Editor: Ivo Hofacker

**ABSTRACT**

**Summary:** `compcodeR` is an `R` package for benchmarking of differential expression analysis methods, in particular, methods developed for analyzing RNA-seq data. The package provides functionality for simulating realistic RNA-seq count datasets, an interface to several of the most commonly used differential expression analysis methods and extensive functionality for evaluating and comparing different approaches on real and simulated data.

**Availability and implementation:** `compcodeR` is available from http://www.bioconductor.org/packages/release/bioc/html/compcodeR.html

**Contact:** Charlotte.Soneson@isb-sib.ch or charlottesoneson@gmail.com

Received on December 4, 2013; revised on April 3, 2014; accepted on May 2, 2014

## 1 INTRODUCTION

Transcriptome profiling studies using RNA-seq with the goal of finding genes that are differentially expressed (DE) between conditions are abundant in the current scientific literature and can be expected to become even more so, as next-generation sequencing technology becomes cheaper and more accessible. RNA-seq experiments generate millions of short reads, which are aligned to a reference sequence to yield a quantitative measure of the expression levels of a collection of genes or other features. Typically, the processed data are represented as a *count matrix*, which constitutes the input for many differential expression methods.

In the past few years, many novel differential expression methods applicable to count matrices obtained from RNA-seq experiments have been presented (for example, Anders and Huber, 2010; Hardcastle and Kelly, 2010; Robinson *et al*., 2010; Tarazona *et al*., 2011). At this point, both users and developers of such methods would thus greatly benefit from objective and standardized benchmarking and characterization of new and existing approaches. A few comparison studies have been published (e.g.Robles *et al*., 2012; Soneson and Delorenzi, 2013). However, the fast pace at which new and updated methods are being presented and the varying objectives of the users create a need for a tool that makes it easy to evaluate and compare a collection of methods from many different aspects in a standardized way. In this applications note, we present such a tool. `compcodeR` (COMParison of COunt-based Differential Expression analysis methods with R) is a benchmarking `R` package that, in a few steps, lets the user evaluate and compare differential expression methods using an approach similar to the one used by Soneson and Delorenzi (2013). Accompanying the package is a large collection of simulated and real-world benchmarking datasets, together with differential expression results obtained by >20 different approaches (available from http://bcf.isb-sib.ch/data/compcodeR). The exact `R` code used to run each differential expression analysis is included and can be rerun to reproduce the results. Taken together, the package provides users with a pedagogical interface to understand and compare differential expression methods, and gives developers an accessible tool for standardized benchmarking of newly developed approaches.

## 2 EXAMPLE

This section outlines the three major functionalities of `compcodeR`. First, the package contains a function for generating synthetic RNA-seq count matrices, using the approach described by Robles *et al.* (2012) and Soneson and Delorenzi (2013). The user defines the properties of the dataset, such as the number of genes and samples, the fraction of truly DE genes and their effect size distribution, several parameters governing the inclusion of outlier counts and filter thresholds. The following code simulates a dataset consisting of Negative Binomially distributed data for five samples from each of two conditions and 12,500 genes, 10% of which are truly DE.

```
> library(compcodeR)
> dat <- generateSyntheticData(dataset = "mydata",
      n.vars = 12500, samples.per.cond = 5,
      n.diffexp = 1250, fraction.upregulated = 0.5,
      repl.id = 1, output.file = "mydata_5spc.rds")
```

The code generates an object of the class `compData`, which is saved to a file named `mydata_5spc.rds`. By rerunning the data simulation with `repl.id` set to different values, it is possible to generate multiple replicate datasets for a given simulation setting, which can make method comparisons more robust and informative.

Second, the package provides an interface to several of the most commonly used methods for differential expression analysis of RNA-seq data. The intention is not to cover all available methods or exploit all of their possibilities, and new methods can easily be included by the user. The code below applies the differential expression test implemented in edgeR (Robinson *et al*., 2010) to the data simulated above, and saves a new `compData` object containing also the test results. A list of the differential expression methods to which `compcodeR` provides an interface can be obtained using the function `listcreateRmd`.

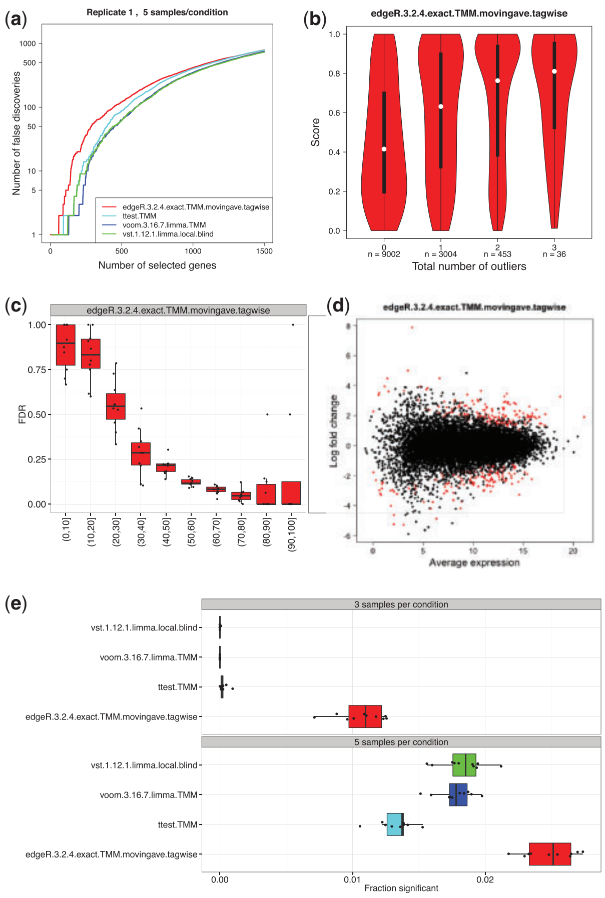

**Fig. 1.** Excerpts from five example figures generated by compcodeR in a characterization of and comparison among four differential expression methods. The comparison is based on results from the four methods applied to 10 replicates of a synthetic dataset for two sample sizes (three and five samples per condition). (**a**) False discovery curves, depicting the number of false positives encountered when stepping through the list of genes ranked by significance. (**b**) The distribution of the gene-wise scores (in this case, defined as 1 minus the nominal $P$-value) assigned by one of the evaluated methods, as a function of the number of outlier counts introduced in the simulation. (**c**) Observed false discovery rates as a function of the binned average expression level. (**d**) An MA plot, depicting the log-fold change against the average expression level, with significantly DE genes marked in color. (**e**) The fraction of genes called significantly DE by each of the methods. Each boxplot summarizes the results from all 10 dataset replicates

The `runDiffExp` function automatically includes the executed code as well as the output from the `R` console in the result object. The code can be written to an HTML file by the `generateCodeHTMLs` function.

```
> runDiffExp(data.file = "mydata_5spc.rds",
    result.extent = "edgeR.exact",
    Rmdfunction = "edgeR.exact.createRmd",
    output.directory = ".",
    norm.method = "TMM", disp.type = "tagwise",
    trend.method = "movingave")
```

The third pillar of the package is the large number of metrics for comparison of differential expression results obtained by different methods. Many of the metrics are general and can also be applied to test results from other types of data, such as microarrays. Some, such as the overlap between the sets of DE genes found by different methods, are independent of knowledge about the true differential expression status of the genes and can thus be applied to any dataset. Others, such as comparisons of the observed false discovery rates for the different methods, are only applicable when the true differential expression status for each gene is known. The 15 comparison metrics are described in more detail in the package vignette. The code below launches the comparison.

```
> runComparisonGUI(input.directories = ".",
    output.directory = ".", recursive = FALSE)
```

The function `runComparisonGUI` scans the provided input directories for result objects and opens a graphical user interface (GUI) where the user can select the dataset on which to base the comparison, which methods to compare and which comparison metrics to use. To facilitate inclusion in an automated analysis pipeline, the GUI can be circumvented, and the comparison can be directly performed using the function `runComparison`. The results of the comparison are written to an HTML report that is saved in the designated output directory. Figure 1 shows excerpts from five example plots from such a report.

## ACKNOWLEDGEMENT

The author would like to thank Mauro Delorenzi, Sarah Gerster, Edoardo Missiaglia and Giovanni d'Ario for valuable input.

*Conflict of Interest*: none declared.

## REFERENCES

Anders,S. and Huber,W. (2010) Differential expression analysis for sequence count data. *Genome Biol.*, **11**, R106.

Hardcastle,T.J. and Kelly,K.A. (2010) baySeq: empirical Bayesian methods for identifying differential expression in sequence count data. *BMC Bioinformatics*, **11**, 422.

Robinson,M.D. *et al.* (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139–140.

Robles,J.A. *et al.* (2012) Efficient experimental design and analysis strategies for the detection of differential expression using RNA-sequencing. *BMC Genomics*, **13**, 484.

Soneson,C. and Delorenzi,M. (2013) A comparison of methods for differential expression analysis of RNA-seq data. *BMC Bioinformatics*, **14**, 91.

Tarazona,S. *et al.* (2011) Differential expression in RNA-seq: a matter of depth. *Genome Res.*, **21**, 2213–2223.