Manual for the package samExploreR

Alexey Stupnikov[1], Shailesh Tripathi[1,2], Ricardo de Matos Simoes[1], Darragh McArt[3], Manuel Salto-Tellez[3], Galina Glazko[4] and Frank Emmert-Streib[1,5,6,*]
[1]Computational Biology and Machine Learning Laboratory,
Center for Cancer Research and Cell Biology,
School of Medicine, Dentistry and Biomedical Sciences,
Faculty of Medicine, Health and Life Sciences,
Queen' [2]School of Mathematics and Physics,
Queen's University Belfast, BT7 1NN Belfast, UK
[3]Northern Ireland Molecular Pathology Laboratory,
Centre for Cancer Research and Cell Biology,
Queen's University Belfast, BT9 7AE Belfast, UK
[4]Division of Biomedical Informatics, University of Arkansas for Medical Sciences,
Little Rock, AR 72205, USA
[5]Computational Medicine and Statistical Learning Laboratory,
Department of Signal Processing, Tampere University of Technology, Finland
[6]Institute of Biosciences and Medical Technology,
33520 Tampere, Finland
[*]Corresponding author

Contents

1 Citation

If you use the samExploreR package, please cite [**?**] and [1].

2 Dependencies

SamExploreR depends on some R package available in CRAN that need to be installed before you can use the package. The symbol '>' indicates the R prompt.

1) ggplot2: package available at http://cran.r-project.org/web/packages/ggplot2/index.html

Installation command: call the following command from an R prompt.

```
> install.packages("ggplot2")
```

2) pBrackets: package available at http://cran.r-project.org/web/packages/pBrackets/index.html

Installation command: call the following command from an R prompt.

```
> install.packages("pBrackets")
```

3) colorspace: package available at http://cran.r-project.org/web/packages/colorspace/index.html

Installation command: call the following command from an R prompt.

```
> install.packages("colorspace")
```

2.1 For developers only

If a user wants to make changes to the package and rebuild it again, then the following packages need to be installed additionally. These are necessary for the compilation and the building the package.

1) BiocCheck:

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("BiocCheck")
```

2) BiocGenerics:

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("BiocGenerics")
```

3) RUnit:

```
> install.packages("RUnit")
```

4) Matrix:

```
> install.packages("Matrix")
```

3 Installation

The samExploreR package is available from Bioconductor. To install the package, run the following commands:

```
Remark: To be added after uploaded to Bioconductor.
```

If you install it from a local download, run the following command in a terminal from the same directory where you downloaded the package:

```
R CMD INSTALL samExploreR_1.0.0.tar.gz
```

4 Quick start

In the following, we demonstrate briefly the usage of the samExploreR package and the analysis procedures it provides. These examples require a BAM or SAM file from some RNA-seq experiment containing aligned reads. For the following examples, we provide the all necessary files, namely:

- aligned reads: Test.sam
- annotation file: Annot.gtf

In order to simulate 5 repeats (N_boot) for a virtual sequencing experiment with sequencing depth $f = 0.7$ execute:

```
 # Loading library
>  library(samExploreR)
```

```
 # Performing subsampling

> inpf <- system.file("testdata", package="samExploreR")
> inpf <- paste(inpf,"/","Test.sam",sep="")

> x <- samExplore(files=inpf, subsample_d = 0.85, N_boot=5)
```

This results in a 5-dimensional list, whereas each component corresponds to a subsampled count vectors.

5 The samExplore function

The function $samExplore$, which is our modification of the featureCounts function of [1], performs a summerization of reads to genomic features of annotation.

The procedure of read subsampling works as follows: One of the input parameters is f, which is a fraction of reads that will be subsampled from the original SAM or BAM file,

$$f = \frac{\text{\#subssampled reads}}{\text{\#total reads}}. \tag{1}$$

During the counts computing process every read, or a pair of reads for paired-end sequencing, is taken into account with probability f. This results in a reduction of the amount of reads and, therefore, the overall expression of the genes.

Further input arguments of the function are:

- files: names of SAM/BAM files to process
- annot_ext: annotation file
- isGTFAnnotationFile:
- GTF.featureType:
- N_boot: number of repeated experiments

As an output the function produces a vector of objects - results of featureCounts running.

```
 # Loading library
>  library(samExploreR)

 # Performing subsampling

> inpf <- system.file("testdata", package="samExploreR")
> inpf <- paste(inpf,"/","Test.sam",sep="")

 # Performing robustness analysis for f = 0.7, number of replicates 5,
 #annotation entries 'gene', non-paired reads
 > x <- samExplore(files=inpf,annot.inbuilt="mm9",GTF.featureType="exon",
 GTF.attrType="gene_id", subsample_d = 0.8, N_boot=5)

 # Performing robustness analysis for f = 0.1, number of replicates 10,
 #annotation entries 'exon', paired reads
```

```
> x <- samExplore(files=inpf,annot.inbuilt="mm9",GTF.featureType="gene",
GTF.attrType="gene_id", subsample_d = 0.8, N_boot=5)
```

6 exploreRob

A cornerstone of any scientific study is the question regarding the reproducibility and robustness of obtained results. The function *exploreRob* allows the exploration of the robustness of results. It runs a standard one-way ANOVA test for groups of replicates, corresponding to different f values, for one fixed annotation. In this way, one can measure if the result of the analysis changes significantly with a change in the parameter sequencing depth, f.

The function *exploreRob* takes as input argument a data frame with the format shown in Tab. 1. In this table,

	Label	Variable	Value
.	.	.	.
1	New, Gene	0.40	55
2	New, Exon	0.99	176
3	Old, Gene	0.85	47
4	Old, Gene	0.70	32
5	New, Exon	0.85	128
6	New, Gene	0.20	10
7	New, Exon	0.25	28
8	Old, Gene	0.95	36
9	New, Exon	0.99	173
10	Old, Gene	0.90	42
11	Old, Gene	0.10	1
12	Old, Gene	0.99	46
13	New, Exon	0.20	14
14	New, Exon	0.10	2
15	Old, Gene	0.25	8
16	New, Gene	0.95	137
17	New, Exon	0.95	180
18	New, Gene	0.25	19
19	New, Exon	0.50	101
20	New, Gene	0.20	8
.	.	.	.
.	.	.	.

Table 1: Input argument for the function *exploreRob*.

the first column provides the labels for the annotation (lbl) used for the analysis, the second column gives the f value and the third column provides the value of the metric to be explored, e.g., the number of differentially expressed genes.

The function *exploreRob* splits this data frame up by considering only entries for one specific type of annotation. Then an ANOVA test is run for the groups of replicates that correspond to a given list of f values, see Fig. 1.

For instance, to explore the robustness of the annotation type AnnotB cross the f values $0.8, 0.9, 0.95$ for the data frame df run:

```
> #Loading library
> library(samExploreR)
> data("df_sole")
>  #Performing robustness analysis
> exploreRob(df_sole, lbl = 'New, Gene', f_vect = c(0.85, 0.9, 0.95))
```

```
[1] "ANOVA test for label 'New, Gene' and f values 0.85, 0.9, 0.95"
Call:
   aov(formula = df_d_sub[, 2] ~ df_d_sub[, 1], data = df_d_sub)

Terms:
                df_d_sub[, 1] Residuals
Sum of Squares         669.780  3234.007
Deg. of Freedom              1        73

Residual standard error: 6.655934
Estimated effects may be unbalanced
```

7 exploreRep

Similar to *exploreRob*, the function *exploreRep* allows the exploration of the reproducibility of results. This function runs a one-way ANOVA test for groups of replicates, corresponding to one f value, across various annotation types. Thus, the influence of the annotation or the summarisation method can be explored.

exploreRep takes as input a data frame of form shown in Tab. 2. Here the first column provides the labels for the annotation used for the analysis, the second column gives the f value and the third column provides the value of the metric to be explored, e.g., the number of differentially expressed genes.

exploreRep splits this data frame up to consider only results for one f value. ANOVA test is run for groups of replicates with corresponding to given list of annotation labels , Fig 1.

For instance, to explore the reproducibility for a f value of 0.7 cross the annotation types AnnotA, AnnotB, AnnotC for data frame df run:

```
> #Loading library
> library(samExploreR)
> data("df_sole")
>  #Performing robustness analysis
>  t = exploreRep(df_sole, lbl_vect = c('New, Gene', 'Old, Gene', 'New, Exon'), f = 0.9)
```

```
[1] "ANOVA test for labels 'New, Gene', 'Old, Gene', 'New, Exon' and f value 0.9"
```

```
>
```

	Label	Variable	Value
.	.	.	.
.	.	.	.
1	New, Gene	0.05	3
2	New, Exon	0.05	1
3	Old, Gene	0.05	1
4	New, Gene	0.05	1
5	New, Exon	0.05	2
6	Old, Gene	0.05	0
7	New, Gene	0.05	2
8	New, Exon	0.05	3
9	Old, Gene	0.05	0
10	New, Gene	0.05	2
.	.	.	.
.	.	.	.

Table 2: Input argument for the function $exploreRep$.

8 plotSamExplorer

This function generates boxplots of the metric under investigation, in our case for the number of differentially expressed genes, for different values of f.

The input argument of this function should be a *data.frame* object containing three columns with the names - *Label*, *Variable*, *Value*.

The data should look like in the following example:

```
> require(samExploreR)
> ########## Loading the example data
> data("df_sole")
> data("df_intersect")
> head(df_sole)

      Label Variable Value
1 New, Gene     0.05     3
2 New, Exon     0.05     1
3 Old, Gene     0.05     1
4 New, Gene     0.05     1
5 New, Exon     0.05     2
6 Old, Gene     0.05     0

> #head(df_intersect)
>
```

8.1 Plotting data using the function plotSamExplorer

```
> ### Generation of the plot:
> require(samExploreR)
> data("df_sole")
> plotsamExplorer(df_sole,p.depth=.9,font.size=4, anova=TRUE)
[1] "ANOVA test for label 'New, Exon' and f values 0.9, 0.95, 0.99"
[1] "ANOVA test for label 'New, Gene' and f values 0.9, 0.95, 0.99"
[1] "ANOVA test for label 'Old, Gene' and f values 0.9, 0.95, 0.99"
```

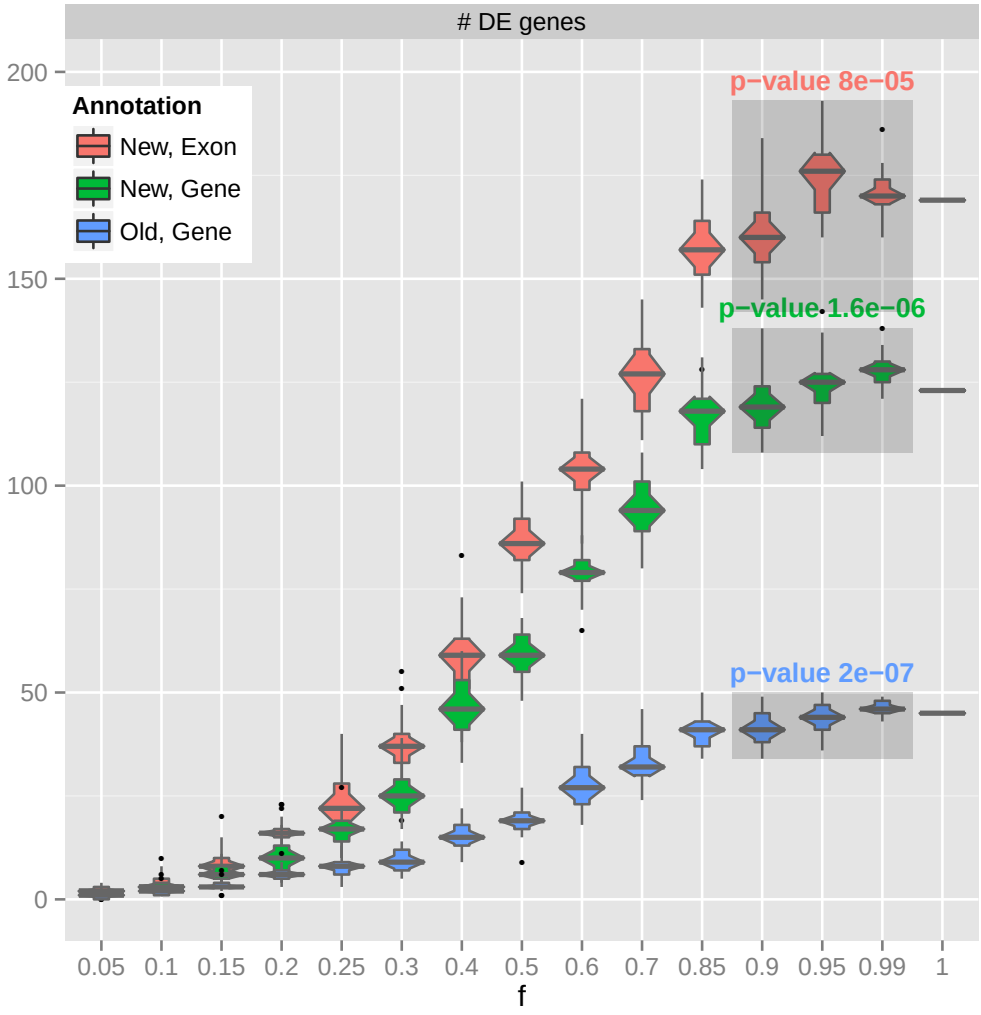

Figure 1: Boxplot of the number of differentially expressed genes for different sequence-depths f.

```
> sessionInfo()

R version 3.1.2 (2014-10-31)
Platform: x86_64-apple-darwin13.4.0 (64-bit)

locale:
[1] C/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8

attached base packages:
```

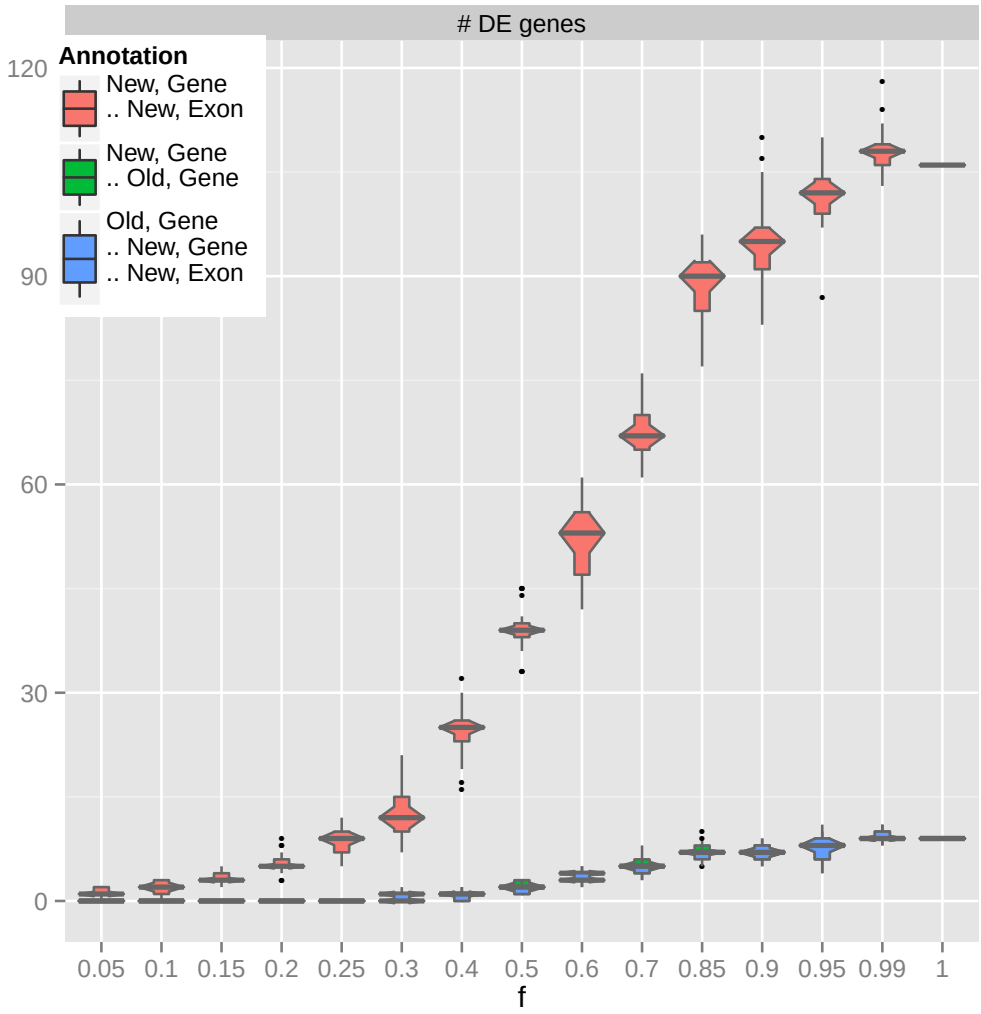

Figure 2: Boxplot of the number of differentially expressed genes for different sequence-depths f.

```
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] samExploreR_1.0.0 colorspace_1.2-6  pBrackets_1.0     ggplot2_1.0.1

loaded via a namespace (and not attached):
 [1] BiocStyle_1.4.1 MASS_7.3-40     Rcpp_0.11.5     digest_0.6.8    grid_3.1.2
 [6] gtable_0.1.2    labeling_0.3    munsell_0.4.2   plyr_1.8.2      proto_0.3-10
[11] reshape2_1.4.1  scales_0.2.4    stringr_0.6.2   tools_3.1.2
```

References

[1] Y Liao, GK Smyth, and W Shi. featurecounts: an efficient general-purpose read summarization program. *arXiv*, 1305:16, 2013.