compareDEtools (v0.1.0)

- *User's Manual* -

Author: BuKyung Baik <back829@unist.ac.kr>

Last updated: 2019. 5. 27.

1. Introduction

compareDEtools is a R package that performs DE analysis on generated dataset and compare performances among DE analysis tools. KIRC, Bottomly and SEQC datasets are given. 14 different DE methods including baySeq, DESeq, DESeq2, edgeR, limma, PoissonSeq, ROTS and SAMSeq are set to run comparison analysis. Based on compcodeR package, datasets and DE analysis results are produced in compData format. Performance boxplots, heatmaps and pca plots can be drawn for analysis. The installation and usage of comareDEtools is described below.

2. Installation (Do only once)

1) Open R program.
2) Type following commands in R console.

```
>> install.packages('devtools')
>> library(devtools)
>>
>> install.packages('BiocManager')
>> BiocManager::install(c('baySeq','Biobase','compcodeR','DESeq','DESeq2','edgeR,','impute','limma','ROTS'))
>> install.packages('gplots','gtools','ggplot2','PoissonSeq','reshape','RColorBrewer','ROCR','samr','SimSeq','statmod','XML')
>>
>> install_github('unistbig/compareDEtools')
```

2. Run

1) Load the compareDEtools R package by typing following line in R console

```
>> library(compareDEtools)
```

2) Generate dataset for DE analysis methods using 'GenerateSyntheticSimulation' or 'GenerateRealSimulation' function.

```
## example code ##
```

```
# 1) Generate working directories for saving dataset, analysis results and result plots.
>> dataset.dir = ‘~/test/dataset/’
>> analysis.dir = ‘~/test/analysis/’
>> figure.dir = ‘~/test/figure/’

# 2-a) Generate synthetic dataset for DE analysis with parameters of given datasets.
>> GenerateSyntheticSimulation(working.dir=dataset.dir, data.types='KIRC',
fixedfold=FALSE, rep=10, nsample=c(3), nvar=1000, nDE=c(50), fraction.upregulated = 0.5,
disp.Types = 'same', modes=c('D'))

# 2-b) Generate datasets with gene counts of given datasets
>> GenerateRealSimulation(working.dir=dataset.dir, fpc=FALSE, data.types='KIRC', rep=10,
nsample=c(3))
>> GenerateRealSimulation(working.dir=dataset.dir, fpc=TRUE, data.types='KIRC', rep=10,
   nsample=c(3))
```

You can choose a dataset to analyze with. Depending on whether the dataset is synthetic dataset or not, further workflow and involved functions change.
As shown in the code above, *GenerateSyntheticSimulation* function takes ten arguments as follows:

① **working.dir**: A character parameter indicating generated dataset save location

② **data.types:** A vector parameter of given datasets to generate synthetic datasets with. (e.g. data.type = c(‘KIRC’, ‘Bottomly’, ‘mBdK’ and ‘mKdB’))

③ **fixedfold:** A logical indicating whether this dataset uses random fold changes following exponential distribution or fixed fold change values. Fixed fold change values are pre-assigned to compare KIRC dataset to given SEQC dataset analysis.

④ **rep:** An integer specifying how many datasets will be generated for each condition to run DE analysis methods.

⑤ **nsample:** An integer vector indicating how many samples are in each sample group.

⑥ **nvar:** An integer vector indicating how many genes are in each dataset.

⑦ **nDE:** An integer indicating how many DE genes are in each dataset.

⑧ **fraction.upregulated:** A numeric vector specifying proportions of upregulated DE genes among total DE genes in the generated dataset.

⑨ **disp.Types:** A vector indicating how is the dispersion parameter assumed to be for each condition to generate a synthetic data. Possible values are 'same' and 'different'.

⑩ **modes:** A character vector specifying test conditions we used for simulation data generation.
"D" for basic simulation (not adding outliers).
"R" for adding 5% of random outlier.
"OS" for adding outlier sample to each sample group.

"DL" for decreasing KIRC simulation dispersion 22.5 times (similar to SEQC data dispersion) to compare with SEQC data.

GenerateRealdataSimulation function with real data workflow takes five arguments as follows

① **working.dir**: A character parameter indicating generated dataset save location

② **fpc:** A logical indicating whether this dataset is generated with samples from single sample group for calculating false positive counts.

③ **data.types:** A character vector of given datasets to generate synthetic datasets with. (e.g. ‘KIRC’, ‘Bottomly’, and ‘SEQC’)

④ **rep:** An integer specifying how many datasets will be generated for each condition to run DE analysis methods.

⑤ **nsample:** An integer vector indicating how many samples are in each sample group.

3) Run DE analysis methods using ‘runSimulationAnalysis’ function

```
## example code ##

# 3) Assign methods for DE analysis.

>> AnalysisMethods=c('edgeR','edgeR.ql','edgeR.rb','DESeq.pc','DESeq2','voom.tmm','
   voom.qn','voom.sw','ROTS','BaySeq','BaySeq.qn','PoissonSeq','SAMseq')dataset.dir
   = ‘~/test/dataset/’

# 4-a) run DE analysis methods with synthetic dataset.

>> runSimulationAnalysis(working.dir=dataset.dir, output.dir=analysis.dir, real=FALSE,
fpc=FALSE, data.types='KIRC', fixedfoldfold=FALSE, rep=10, nsample=c(3), nDE=c(50),
fraction.upregulated=0.5, disp.Types=c('same'), modes=c('D'), AnalysisMethods =
AnalysisMethods, para=list())

# 4-b) run DE analysis methods with real dataset.

>> runSimulationAnalysis(working.dir=dataset.dir, output.dir=analysis.dir, real=TRUE,
fpc=FALSE, data.types='KIRC', rep=10, nsample=c(3), AnalysisMethods = AnalysisMethods,
para=list())

>> runSimulationAnalysis(working.dir=dataset.dir, output.dir=analysis.dir, real=TRUE,
   fpc=TRUE, data.types='KIRC', rep=10, nsample=c(3), AnalysisMethods =
   AnalysisMethods, para=list())
```

runSimulationAnalysis function takes fourteen arguments as follows:

① **working.dir**: A character parameter indicating save location of generated dataset to run DE analysis

② **output.dir:** A character parameter indicating save location of results for each DE analysis methods.

③ **real:** A logical indicating whether this analysis is run with real dataset or synthetic dataset.

④ **fpc:** A logical indicating whether this dataset is generated with samples from same sample group for calculating false positive counts.

⑤ **data.types:** A character vector indicating which given dataset our target dataset is based on. ‘KIRC’, ‘Bottomly’, ‘mBdK’ and ‘mKdB’ are available for synthetic datasets and ‘KIRC’, ‘Bottomly’ and ‘SEQC’ are available for real datasets.

⑥ **fixedfold:** A logical indicating whether target dataset uses random fold changes following exponential distribution or fixed fold change values. This parameter is for synthetic datasets.

⑦ **rep:** An integer specifying iterations for each condition to run DE analysis methods.

⑧ **nsample:** An integer vector indicating how many samples are in each sample group.

⑨ **nDE:** An integer vector indicating how many DE genes are in the generated dataset. This parameter is for synthetic datasets.

⑩ **fraction.upregulated:** A numeric vector specifying proportions of upregulated DE genes among total DE genes in the generated dataset. This parameter is for synthetic datasets.

⑪ **disp.Types:** A vector indicating how is the dispersion parameter assumed to be for each condition in the generated dataset. Possible values are 'same' and 'different'. This parameter is for synthetic datasets.

⑫ **modes:** A character vector specifying test conditions used for generated datasets. This parameter is for synthetic datasets.
"D" for basic simulation (not adding outliers).
"R" for adding 5% of random outlier.
"OS" for adding outlier sample to each sample group.
"DL" for decreasing KIRC simulation dispersion 22.5 times (similar to SEQC data dispersion) to compare with SEQC data.

⑬ **AnalysisMethods:** A character vector indicating which DE tools will be used for analysis.

⑭ **para:** A list parameter indicating the parameters to run each DE analysis methods. It contains lists corresponding each method and each list contain the parameters for each DE analysis methods. The analysis methods not in the para list will be run with default parameters. (e.g. para=list(ROTS=list(transformation=FALSE, normalize=FALSE)))

4) Plot DE analysis results using ‘performance_plot’ or ‘performance_realdata_plot’ function

## example code ##

```
# 5-a) plot synthetic data analysis results
>> performance_plot(working.dir=analysis.dir,figure.dir=figure.dir,fixedfold=FALSE,simul.data='KIRC', rep=10, nsample=c(3), nvar=1000, nDE=50, fraction.upregulated = 0.5, disp.Type = 'same', mode='D', rowType = c('AUC','TPR','trueFDR'), AnalysisMethods=AnalysisMethods)

# 5-b) plot real data analysis results
>>performance_realdata_plot(working.dir=analysis.dir,figure.dir=figure.dir,simul.data='KIRC', rep=10, nsample=c(3), AnalysisMethods=AnalysisMethods, rowType = c("DetectedDE","FP.count"))
```

performance_plot function takes thirteen arguments as follows:

① **working.dir**: A character parameter indicating save location of DE analysis results that will be plotted.

② **output.dir:** A character parameter indicating save location of DE analysis result plots.

③ **fixedfold:** A logical indicating whether analyzed dataset used random fold changes following exponential distribution or fixed fold change values.

④ **simul.data:** A character parameter indicating which given dataset analyzed dataset is based on. ‘KIRC’, ‘Bottomly’, ‘mBdK’ and ‘mKdB’ are available.

⑤ **rep:** An integer specifying iterations DE analysis methods run for each condition.

⑥ **nsample:** An integer vector indicating how many samples are in each sample group.

⑦ **nvar:** An integer vector indicating how many genes are in the analyzed dataset.

⑧ **nDE:** An integer vector indicating how many DE genes are in the analyzed dataset.

⑨ **fraction.upregulated:** A numeric vector specifying proportions of upregulated DE genes among total DE genes in the analyzed dataset.

⑩ **disp.Type:** A character parameter indicating how is the dispersion parameter assumed to be for each condition in the analyzed dataset.

⑪ **mode:** A character parameter specifying test conditions used for analyzed datasets.
"D" for basic simulation (not adding outliers).
"R" for adding 5% of random outlier.
"OS" for adding outlier sample to each sample group.
"DL" for decreasing KIRC simulation dispersion 22.5 times (similar to SEQC data dispersion) to compare with SEQC data.

⑫ **rowType:** A character vector indicating which results are shown in performance plot. Sub-vectors of c(‘AUC’, ‘TPR’, ‘trueFDR’) are available.

⑬ **AnalysisMethods:** A character vector indicating which DE tools are used for analysis and will be plotted.

performance_realdata_plot function with real data workflow takes seven arguments as follows

① **working.dir**: A character parameter indicating save location of DE analysis results that will be plotted.

② **output.dir:** A character parameter indicating save location of DE analysis result plots.

③ **simul.data:** A character parameter indicating which given dataset analyzed dataset is based on. ‘KIRC’, ‘Bottomly’, ‘SEQC’ are available.

④ **rep:** An integer specifying iterations DE analysis methods run for each condition.

⑤ **nsample:** An integer vector indicating how many samples are in each sample group.

⑥ **rowType:** A character vector indicating which results are shown in performance plot. Sub-vectors of c(‘DetectedDE’, ‘FP.count’) are available for ‘KIRC and ‘Bottomly’ datasets. Sub-vectors of c(‘DetectedDE’, ‘FP.count’) are available for ‘SEQC’ datasets.

⑦ **AnalysisMethods:** A character vector indicating which DE tools are used for analysis and will be plotted.

3. Other visualization methods

1) FP.count plot for synthetic dataset.

2) Heatmap and clustering for DE gene rank similarity among DE methods.

```
# Case 1: Calculating FP.count for synthetic dataset analysis
# 1) Generate no DE introduced synthetic datasets with multiple test conditions

>> GenerateSyntheticSimulation(working.dir=dataset.dir, data.types='KIRC', rep=10,
   nsample=c(3,10), nvar=10000, nDE=0, fraction.upregulated = 0.5, disp.Types =
   'same', modes=c('D','R','OS'))

# 2) Run DE analysis with no DE introduced synthetic datasets.

>> runSimulationAnalysis(working.dir=dataset.dir, output.dir=analysis.dir, real=FALSE,
fpc=True, data.types='KIRC', rep=10, nsample=c(3,10), nDE=c(0), disp.Types='same',
modes=c('D','R','OS'), AnalysisMethods = AnalysisMethods, para=list())

# 3) Draw FP.count plot with ‘D’, ‘R’ and ‘OS’ conditions.

>>  fpc_performance_plot(working.dir=analysis.dir,figure.dir=figure.dir,simul.data='KIRC',
rep=10, nsample=c(3,10), disp.Type = 'same', modes=c('D','R','OS'),
AnalysisMethods=AnalysisMethods)
```

fpc_performance_plot function takes eight arguments as follows:

① **working.dir**: A character parameter indicating save location of DE analysis results that will be plotted.

② **output.dir:** A character parameter indicating save location of DE analysis result plots.

③ **simul.data:** A character parameter indicating which given dataset analyzed dataset is based on. ‘KIRC’, ‘Bottomly’, ‘mBdK’ and ‘mKdB’ are available.

④ **rep:** An integer specifying iterations DE analysis methods run for each condition.

⑤ **nsample:** An integer vector indicating how many samples are in each sample group.

⑥ **disp.Type:** A character parameter indicating how is the dispersion parameter assumed to be for each condition in the analyzed dataset.

⑦ **modes:** A character vector specifying test conditions used for analyzed datasets.
"D" for basic simulation (not adding outliers).
"R" for adding 5% of random outlier.
"OS" for adding outlier sample to each sample group.
"DL" for decreasing KIRC simulation dispersion 22.5 times (similar to SEQC data dispersion) to compare with SEQC data.

⑧ **AnalysisMethods:** A character vector indicating which DE tools are used for analysis and will be plotted.

```
# Case 2: heatmap for real data analysis
# Use same results from previous 4-b) realdata analysis example codes
# 1) Draw similarity heatmap for preset methods.
>> correlation_heatmap(working.dir=analysis.dir, figure.dir=figure.dir,simul.data='KIRC',
   nsample=5, topgenes=5000, AnalysisMethods=AnalysisMethods, rep=10)
```

correlation_heatmap function takes seven arguments as follows:

① **working.dir**: A character parameter indicating save location of DE analysis results that will be drawn as a heatmap.

② **output.dir:** A character parameter indicating save location of DE analysis result heatmap.

③ **simul.data:** A character parameter indicating which given dataset analyzed dataset is based on. ‘KIRC’, ‘Bottomly’ are available.

④ **rep:** An integer specifying iterations DE analysis methods run for each condition.

⑤ **nsample:** An integer vector indicating how many samples are in each sample group.

⑥ **topgenes:** An integer parameter indicating the number of genes to generate sub geneset to calculate DE gene ranks and make similarity matrix of DE methods.

⑦ **AnalysisMethods:** A character vector indicating which DE tools are used for analysis and will be plotted.