PowerExplorer Manual

Xu Qiao

2018-01-18

Contents

Abstract **2**

Introduction **2**

Prepare Input Data **3**

Run Estimation **4**
Visualization 5

Run Predictions **6**
Visualization 7
Line Plot 7
Heatmap 8

Abstract

This vignette demonstrates the applications of R package `PowerExplorer` as the power and sample size estimation tool for RNA-Seq and quantitative proteomics data.

`PowerExplorer` contains following main features:

- Estimation of power based on the current setting
- Prediction of power according to the future settings (e.g. increasing sample size)
- Visualizations of estimation and prediction results

Introduction

Power and sample size estimation is still one of the important principles in designing next-generation sequencing experiments to discover differential expressions, a few methods on power estimation for RNA-Seq data have been studied, while the one specialized for proteomics data has not yet been developed. `PowerExplorer`is a power estimation and prediction tool currently applicable to RNA-Seq and quantitative proteomics experiments. The calculation procedure starts with estimating the distribution parameters of each gene or protein (following referred as entry for simplicity) accordingly, with the obtained prior distribution of each entry, a specified amount of simulations are executed to generate data (read counts for RNA-Seq and peptide abundance for proteomics) repetitively for each entry based on null and alternative hypotheses. Furthermore, the corresponding statistical tests (t-test or Wald-test) are performed and the test statistics are collected, eventually the result statistics will be summarized to calculate the statistical power.

Prepare Input Data

For both RNA-Seq (gene expression levels) and quantitative proteomics (peptide abundance levels) datasets, the data matrix should be arranged as entries in rows and samples in columns, for example:

```
library(PowerExplorer)
data("exampleRNASeqData")
head(exampleRNASeqData$dataMatrix[,1:6])
#>        Sample_A_1 Sample_A_2 Sample_A_3 Sample_A_4 Sample_A_5 Sample_B_1
#> Gene_1        469        324         38       1059         64        496
#> Gene_2         84        276        263        182        181        737
#> Gene_3        293        173        272        123        475        169
#> Gene_4        310        209        550        212        394       1064
#> Gene_5         82        141        216        202        494        293
#> Gene_6        583         98        137        179        214        884
```

A grouping vector indicating the sample groups to which all the samples belong should also be created, for example:

```
show(exampleProteomicsData$groupVec)
#>  [1] "A" "A" "A" "A" "A" "B" "B" "B" "B" "B" "C" "C" "C" "C" "C"
```

```
colnames(exampleProteomicsData$dataMatrix)
#>  [1] "Sample_A_1" "Sample_A_2" "Sample_A_3" "Sample_A_4" "Sample_A_5"
#>  [6] "Sample_B_1" "Sample_B_2" "Sample_B_3" "Sample_B_4" "Sample_B_5"
#> [11] "Sample_C_1" "Sample_C_2" "Sample_C_3" "Sample_C_4" "Sample_C_5"
```

Note that the grouping vector length should be equal to the column number of the data matrix, all groups conventionally should have the same number of samples, otherwise the tool will automatically even all the sample numbers to the least number to achieve equal groups.

Run Estimation

Here we use a randomly generated proteomics dataset `exampleProteomicsData` as an example to estimate the current power of the dataset. The input dataset is named as `dataMatrix` and the grouping vector as `groupVec`.

To run the estimation, apart from the input, we still need to specify the following parameters:

- `isLogTransformed`: FALSE; the input data is not log-transformed.
- `dataType`: "RNA-Seq"; the datatype can be declared as "Proteomics" or "RNA-Seq".
- `minLFC`: 0.5; the threshold of Log2 Fold Change, proteins with lower LFC will be discarded.
- `alpha`: 0.05; the controlled false positive (Type I Error) rate.
- `ST`: 50; the simulation of each protein entry will be run 50 times (ST>50 is recommended).
- `seed`: 345; the seed of the random variables to maintain the reproducibility.
- `showProcess`: FALSE; no detailed processes will be shown, set to TRUE if debug is needed.
- `saveSimulatedData`: FALSE; if TRUE, save the simulated data in ~/savedData directory.

The results will be summaried in barplot, boxplot and summary table.

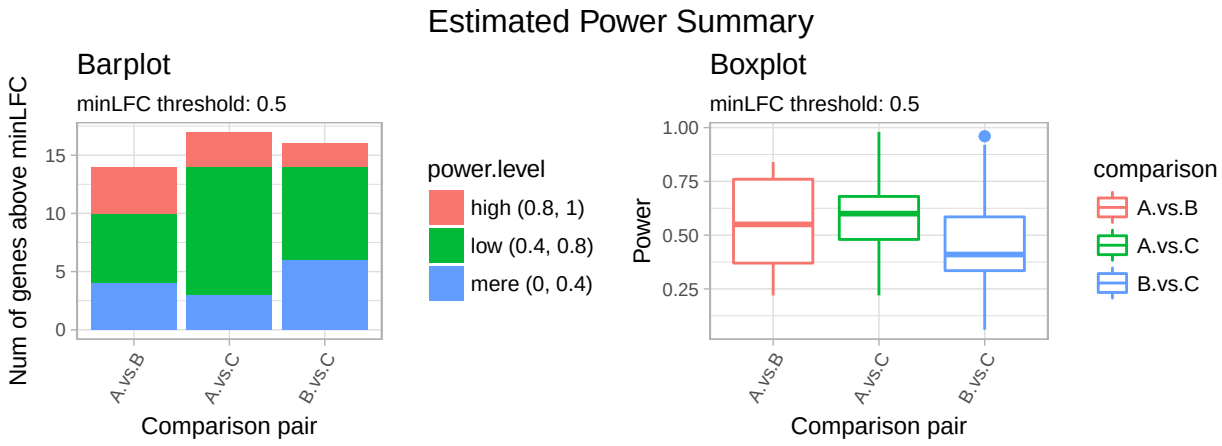
```
library(PowerExplorer)
data("exampleRNASeqData")
estimatedPower <- estimateCurrentPower(inputDataMatrix = exampleRNASeqData$dataMatrix,
                                       groupVec = exampleRNASeqData$groupVec,
                                       isLogTransformed = FALSE,
                                       dataType = "RNA-Seq",
                                       minLFC = 0.5,
                                       alpha = 0.05,
                                       ST = 50,
                                       seed = 345,
                                       showProcess = FALSE,
                                       saveSimulatedData = FALSE)
```

A part of the output should look like this:

```
##------ Wed Jan 17 16:50:25 2018 ------##
0 of 110 entries are filtered due to excessive zero counts
[!]START ESTIMATION
Estimating NB parameters by DESeq2...
Number of groups:        3
Number of replicates:    5
Number of simulations:   50
Min. Log Fold Change:    0.5
False Postive Rate:      0.05
Transformed:     FALSE


Power Estimation between group A and group B:

Quantiles of absolute Log2 Fold Change:
     0%     10%     20%     30%     40%     50%     60%     70%     80%     90%    100%
0.00000 0.05000 0.08000 0.13000 0.16000 0.20000 0.26525 0.31550 0.39000 0.57800 1.80000

14 of 110 genes are over minLFC threshold 0.5:
>> [==================================================] 100% of All Simulations Completed...
##------ Wed Jan 17 16:50:34 2018 ------##
##------ Wed Jan 17 16:50:34 2018 ------##
Estimating cut-off statistics with false positive rate: 0.05
OVERALL ESTIMATED POWER: 0.5486
```

Visualization

The estimated results can be summarized using `plotEstimatedPower`, the only input needed is the `estimatedPower`, which should be the estimated power object returned from `estimateCurrentPower`.



Comp.	Gene Num.	Avg. Power	H (0.8, 1)	L (0.4, 0.8)	M (0, 0.4)
A.vs.B	14	0.55	4 (28.57%)	6 (42.86%)	4 (28.57%)
A.vs.C	17	0.59	3 (17.65%)	11 (64.71%)	3 (17.65%)
B.vs.C	16	0.47	2 (12.5%)	8 (50%)	6 (37.5%)

The graph contains 3 plots, the `barplot` vertically shows the number of genes/proteins above the minLFC threshold, columns indicates the comparison pairs, each column presents the proportions of three power levels in three colours as indicated in the legend `power.level`; The boxplot shows the overall power distribution of each comparsion; And the summary table summarize the power in a numerical way with the same information shown in the previous two plots.

Run Predictions

With the same dataset, to run a prediction, a few more parameters are needed:

- `isLogTransformed`: FALSE; the input data is not log-transformed.
- `dataType`: "RNA-Seq"; the datatype can be declared as "Proteomics" or "RNA-Seq".
- `rangeSimNumRep`: the power of replicate number 5 to 20 will be predicted.
- `alpha`: 0.05; the controlled false positive rate.
- `ST`: 30; the statistical test and data simulation of each protein entry will be run 30 times (ST>50 is recommended).
- `seed`: 345; specify the seed of the random variables to maintain the reproducibility.
- `showProcess`: FALSE; no detailed processes will be shown, set to TRUE if debug is needed.
- `saveSimulatedData`: FALSE; if TRUE, save the simulated data in root/savedData directory.

Similar to the estimation process, however, the simulations will be excuted with each sample size specified in `rangeSimNumRep`. (Note: the term sample size in this vignette refers to the replicate number of each group/case)

A part of the output should look like this:

```
##------ Thu Jan 18 10:27:46 2018 ------##
0 of 110 entries are filtered due to excessive zero counts
[!]START ESTIMATION
Estimating NB parameters by DESeq2...
Number of groups:        3
Number of replicates:    5, 10, 15, 20
Number of simulations:   30
False Postive Rate:      0.05
Transformed:     FALSE


(1 / 4) Simulation with 5 replicates per group:

Power Estimation between group A and group B:
>> [==================================================] 100% of All Simulations Completed...
##------ Thu Jan 18 10:28:21 2018 ------##

Power Estimation between group A and group C:
>> [==================================================] 100% of All Simulations Completed...
##------ Thu Jan 18 10:28:56 2018 ------##

Power Estimation between group B and group C:
>> [====================================              ] 74% of All Simulations Completed...
```

Visualization

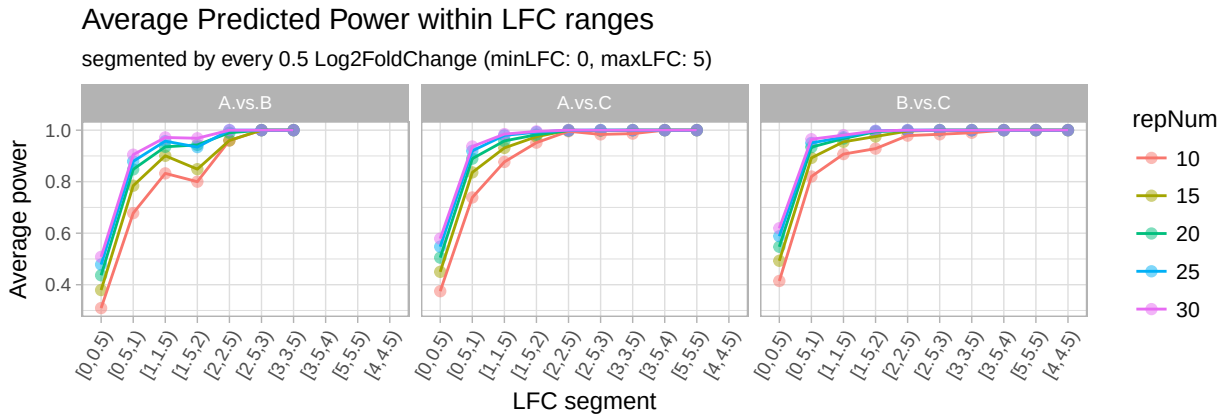
The predicted results can be summaried using `plotPredictedPower`. The input should be the predicted power object returned from `predictSampleSizePower`, the summary can be optionally visualized by setting the following parameters:

- `plotType`: power-samplesize-foldchange relationship can be visualized optionally between "lineplot" and "heatmap".
- `minLFC` and `maxLFC`: to observe power in a specific range of LFC
- `LFCscale`: to determine the LFC scale of the observation

Line Plot

Lineplot (LFCscale = 0.5):

```
data("examplePredictedPower")
plotPredictedPower(examplePredictedPower, plotType = "lineplot", LFCscale = 0.5)
```



	repNum:10	repNum:15	repNum:20	repNum:25	repNum:30
A.vs.B	0.8	0.84	0.88	0.89	0.91
A.vs.C	0.88	0.91	0.93	0.94	0.94
B.vs.C	0.9	0.93	0.94	0.95	0.96

Lineplot is one of the optional outputs of `plotPredictedPower`, the output contains a lineplot and a summary table. For each comparison, the lineplot shows the power tendency across each Log2 Fold Change segment, which resulted from a complete LFC list divided by a specified `LFCscale`. Each dot on the lines stands for the average power (y-axis) of the genes within the LFC range (x-axis), and the colours indicate the average power of the certain sample size as shown in the legend besides the plot. In addition, a summary table below shows the average power of each comparison across the sample sizes.

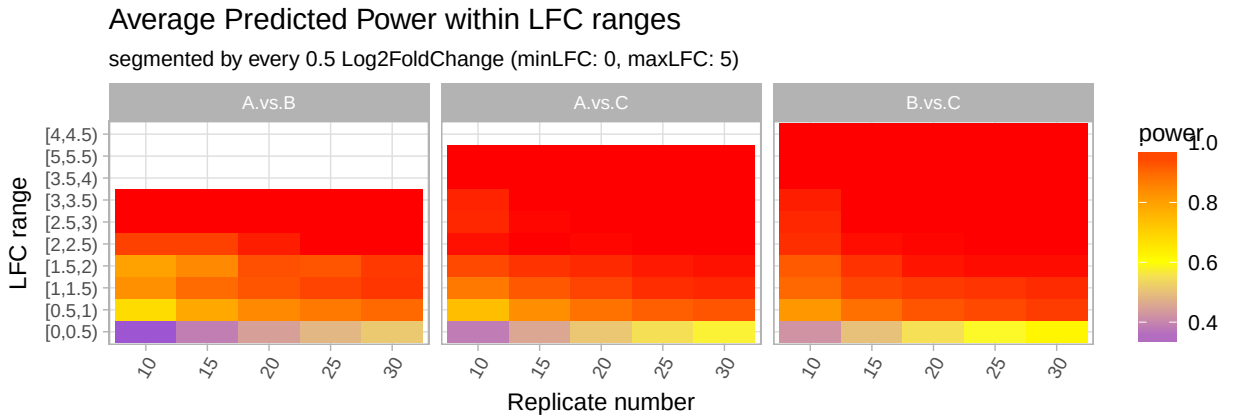
For instance, the line plot here shows the average power of four sample sizes (5 to 30, step=5) in LFCscale of 0.5, the LFC range is between 0 and 5, each LFC segment shows the average power of the entries with the

LFC in this range, here the higher LFC has higher power, additionally as the repNum (sample size) shown with different colours, the average power of each LFC range increases with the larger sample sizes.

Heatmap

Heatmap (LFCscale = 0.5):

```
data("examplePredictedPower")
plotPredictedPower(examplePredictedPower, plotType = "heatmap", LFCscale = 0.5)
```



	repNum:10	repNum:15	repNum:20	repNum:25	repNum:30
A.vs.B	0.8	0.84	0.88	0.89	0.91
A.vs.C	0.88	0.91	0.93	0.94	0.94
B.vs.C	0.9	0.93	0.94	0.95	0.96

The heatmap option presents the power predictions in the similar way, vertically each heatmap shows overall LFC level of a comparison, sometimes a certain range shows blank space, since the result LFC vary in different comparison pairs. The average power of each LFC range is scaled with colours between blue and red, the middle value (0.6) is coloured as yellow, as shown in the colour bar on the right. For example, this graph shows the power increases with larger sample sizes. The same summary table is also shown on the bottom.