PowerExplorer Manual

Xu Qiao

2018-01-16

Introduction

Power and sample size estimation are non-trival PowerExplorer is a power estimation and prediction tool currently designed for RNA-Seq and Quantitative Proteomics analyses. The estimation is based on simulation method which extend the sample size accordinng to the distribution parameters estimated from the input data.

Prepare Input Data

For datasets of both RNA-Seq (gene expression levels) and Quantitative Proteomics (peptide abundance levels), the data matrix should arrange gene/protein entries as rows and samples as columns, for example:

```
library(PowerExplorer)
data("exampleRNASeqData")
head(exampleRNASeqData$dataMatrix[,1:5])
#>        Sample_A_1 Sample_A_2 Sample_A_3 Sample_A_4 Sample_A_5
#> Gene_1        469        324         38       1059         64
#> Gene_2         84        276        263        182        181
#> Gene_3        293        173        272        123        475
#> Gene_4        310        209        550        212        394
#> Gene_5         82        141        216        202        494
#> Gene_6        583         98        137        179        214
```

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 and then predict the power according to desired sample sizes. The input dataset is named as `dataMatrix` and the grouping vector as `groupVec`.

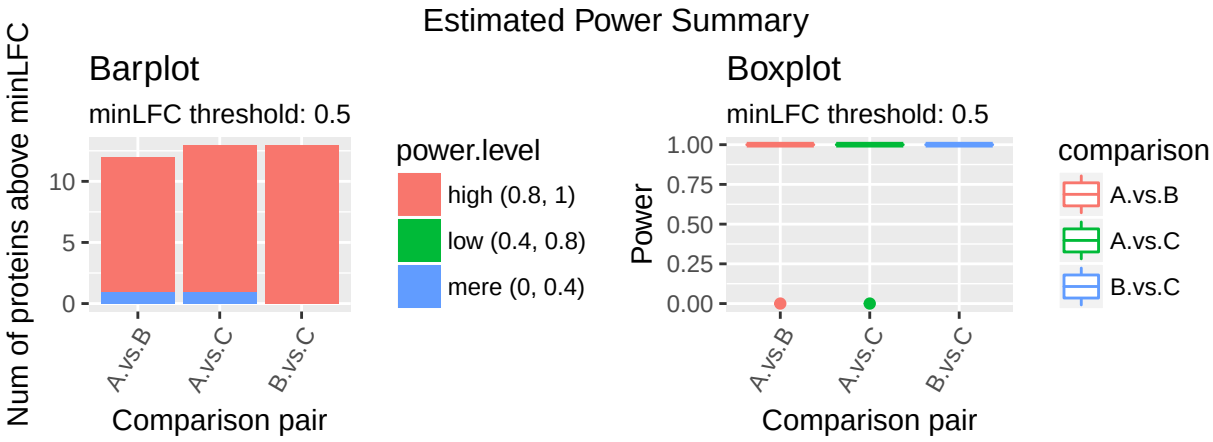
To run the estimation, apart from the input, we still need to specify the following parameters:

- `isLogTransformed`: FALSE, the input data is not log-transformed
- `dataType`: "Proteomics", declare the datatype as "Proteomics"
- `minLFC`: 0.5, the threshold of Log2 Fold Change, proteins with lower LFC will be discarded
- `alpha`: 0.05, the controlled false positive rate
- `ST`: 100, the statistical test and data simulation of each protein entry will be run 100 times
- `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`: TRUE, save the simulated data in root/savedData directory

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

```
library(PowerExplorer)
data("exampleProteomicsData")
estimatedPower <- estimateCurrentPower(inputDataMatrix = exampleProteomicsData$dataMatrix,
                                       groupVec = exampleProteomicsData$groupVec,
                                       isLogTransformed = FALSE,
                                       dataType = "Proteomics",
                                       minLFC = 0.5,
                                       alpha = 0.05,
                                       ST = 1,
                                       seed = 345,
                                       showProcess = FALSE,
                                       saveSimulatedData = FALSE)
```

The estimated results can be summarized using `plotCurrentPower`, the input data should be the estimated power object produced from `estimateCurrentPower`, the result graph can be saved if wanted.



Comp.	Protein Num.	Avg. Power	H (0.8, 1)	L (0.4, 0.8)	M (0, 0.4)
A.vs.B	12	0.92	11 (91.67%)	0 (0%)	1 (8.33%)
A.vs.C	13	0.92	12 (92.31%)	0 (0%)	1 (7.69%)
B.vs.C	13	1.00	13 (100%)	0 (0%)	0 (0%)

Run Predictions

Here we load the same dataset used in the previous estimation, to run a prediction, a few more parameters are needed:

- `isLogTransformed`: FALSE, the input data is not log-transformed
- `dataType`: "Proteomics", declare the datatype as "Proteomics"
- `minLFC`: 0.5, the threshold of Log2 Fold Change, proteins with lower LFC will be discarded
- `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, recommend to be more than 50
- `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`: TRUE, save the simulated data in root/savedData directory

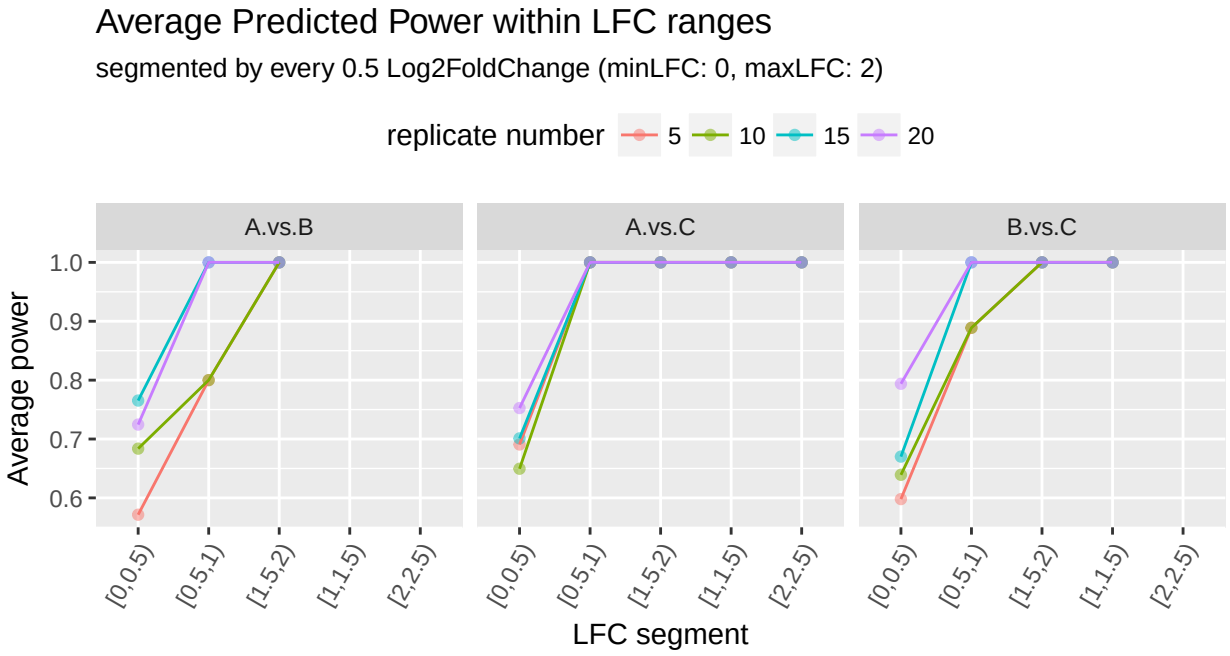
```
data("exampleProteomicsData")
predictedPower <- predictSampleSizePower(inputDataMatrix = exampleProteomicsData$dataMatrix,
                                         groupVec = exampleProteomicsData$groupVec,
                                         isLogTransformed = FALSE,
                                         dataType = "Proteomics",
                                         rangeSimNumRep = c(5, 10, 15, 20),
                                         alpha = 0.05,
                                         ST = 1,
                                         seed = 345,
                                         showProcess = FALSE,
                                         saveSimulatedData = FALSE)
```

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
- `savePlot`: whether the graph will be saved as a picture file

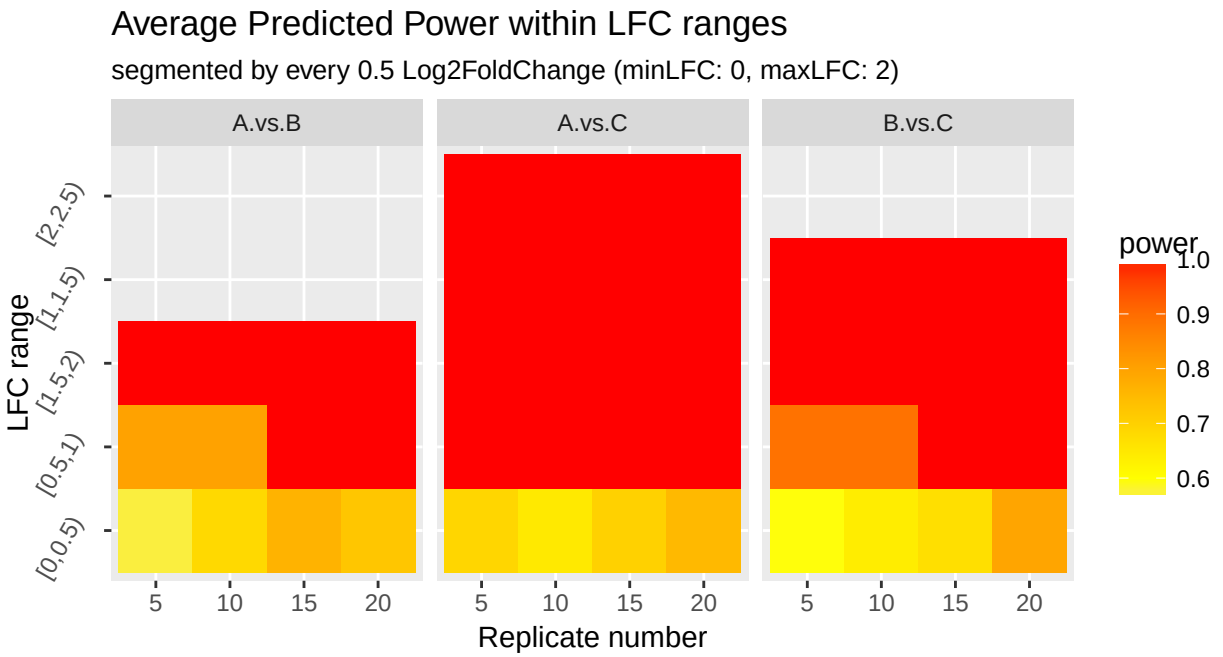
Lineplot (LFCscale = 0.5):

```
plotPredictedPower(predictedPower = predictedPower,
                   plotType = "lineplot",
                   LFCscale = 0.5,
                   savePlot = FALSE)
```



Heatmap (LFCscale = 0.5):

```
plotPredictedPower(predictedPower = predictedPower,
                   plotType = "heatmap",
                   LFCscale = 0.5,
                   savePlot = FALSE)
```


Average Predicted Power within LFC ranges
segmented by every 0.5 Log2FoldChange (minLFC: 0, maxLFC: 2)
A.vs.B
A.vs.C
B.vs.C
LFC range
[2,2.5)
[1,1.5)
[1.5,2)
[0.5,1)
[0,0.5)
5
10
15
20
Replicate number
power
1.0
0.9
0.8
0.7
0.6