

PowerExplorer Manual

Xu Qiao, Laura Elo

2018-01-29

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 R package **PowerExplorer** as a power and sample size estimation tool for RNA-Seq and quantitative proteomics data.

PowerExplorer contains the following main features:

- Estimation of power based on the current data
- Prediction of power corresponding to the increased sample sizes
- Result visualizations

Introduction

Power and sample size estimation is one of the important principles in designing next-generation sequencing experiments to discover differential expressions. **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 feature for simplicity). With the obtained prior distribution of each feature, a specified amount of simulations are executed to generate data (read counts for RNA-Seq and protein 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 statistics will be summarized to calculate the statistical power.

Prepare Input Data

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

```
library(PowerExplorer)
data("exampleRNASeqData")
head(exampleRNASeqData$dataMatrix[,1:6])
```

	<i>Sample_A_1</i>	<i>Sample_A_2</i>	<i>Sample_A_3</i>	<i>Sample_A_4</i>	<i>Sample_A_5</i>	<i>Sample_B_1</i>
#> <i>Gene_1</i>	469	324	38	1059	64	496
#> <i>Gene_2</i>	84	276	263	182	181	737
#> <i>Gene_3</i>	293	173	272	123	475	169
#> <i>Gene_4</i>	310	209	550	212	394	1064
#> <i>Gene_5</i>	82	141	216	202	494	293
#> <i>Gene_6</i>	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"
--------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

The sample groups corresponding to the data:

```
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.

Run Estimation

Here we use a randomly generated RNASeq dataset `exampleRNASeqData` 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 gene will be run 50 times (ST>50 is recommended).
- `seed`: 345; optional, a seed value for the random number generator 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 summarized 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

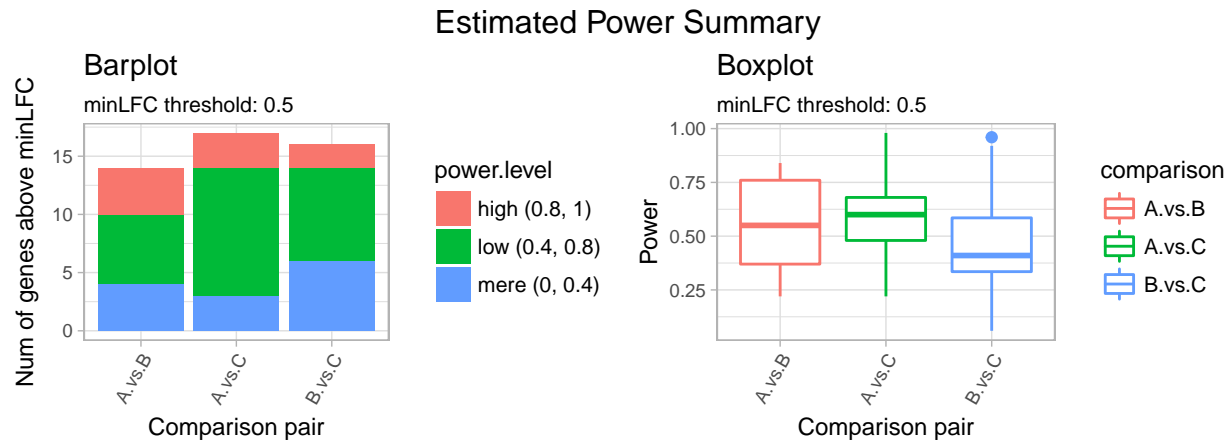
Power Estimation between group A and group B:

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 comparison; 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 different parameter is needed:

- `rangeSimNumRep`: the power of replicate number 5 to 20 will be predicted.

Similar to the estimation process, however, the simulations will be executed 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 Positive 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

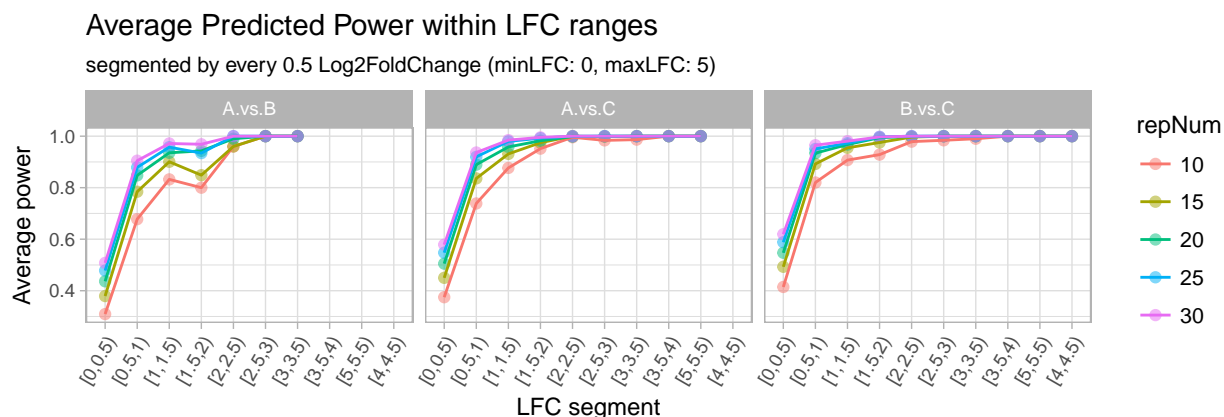
The predicted results can be summarized using `plotPredictedPower`. The input should be the predicted power object returned from `predictSampleSizePower`, the summary can be optionally visualized by setting the following parameters:

- `plotType`: power-sample-size-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)
```



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 every Log2 Fold Change segment 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 each colour indicates the average power of a certain sample size (as shown in the legend besides the plot). In addition, a summary table below displays the average power of each comparison across the sample sizes.

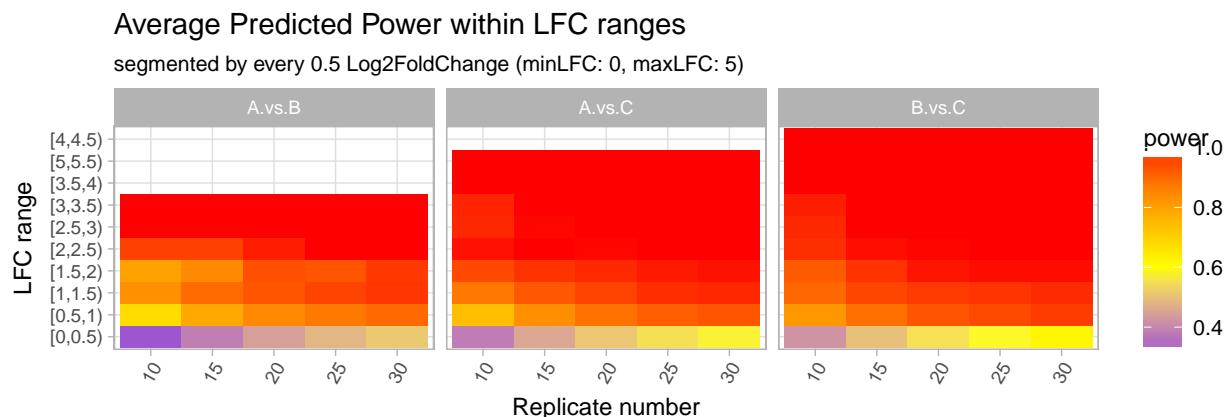
For instance, the line plot here shows the average power of four sample sizes (5 to 30, with increment of 5) in LFCscale of 0.5. The LFC ranges from 0 to 5, and within each LFC segment, the graph shows the

average power of the features. Here, the higher LFC shows higher power, the average power of each LFC range increases with the larger sample sizes, as expected.

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 a similar way, as previous visualizations. Each heatmap displays overall power of each LFC range and sample size. The average power of each LFC range is scaled with colours between blue and red, as shown in the colour bar on the right. For example, this graph shows the power increasing with larger sample sizes as expected. The same summary table is also shown at the bottom.