

# Package ‘PoweREST’

September 3, 2024

**Title** A Bootstrap-Based Power Estimation Tool for Spatial Transcriptomics

**Version** 0.1.0

**Imports** scam, Seurat, dplyr, plotly, resample, xgboost, magrittr, rayshader, ggplot2

**Suggests** patchwork, boot, knitr, rmarkdown, fields, rayrender, tidyr

**Description** Power estimation and sample size calculation for 10X Visium Spatial Transcriptomics data to detect differential expressed genes between two conditions based on bootstrap re-sampling. See Shui et al. (2024) <<https://www.biorxiv.org/content/10.1101/2024.08.30.610564v1.full.pdf+html>> for method details.

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**License** MIT + file LICENSE

**Depends** R (>= 2.10)

**LazyData** true

## Contents

fit_powerest . . . . .	2
fit_XGBoost . . . . .	2
plotly_powerest . . . . .	3
PoweREST . . . . .	4
PoweREST_gene . . . . .	5
PoweREST_subset . . . . .	6
power_example . . . . .	7
pred_powerest . . . . .	7
pred_XGBoost . . . . .	8
result_example . . . . .	9
vis_powerest . . . . .	9
vis_XGBoost . . . . .	10
<b>Index</b>	<b>12</b>

---

fit_powerest	<i>Fit the power surface</i>
--------------	------------------------------

---

### Description

This function loads the power values with corresponding avg\_log2FC and avg\_PCT derived from bootstrap sampling and utilizes the **scam** package to fit two dimensional smoothing splines under monotone constraints: 1.positive relationship between power and avg\_log2FC; 2.positive relationship between power and avg\_PCT. The values of avg\_log2FC and avg\_PCT can be either from the averages of the bootstrap samples or from the original spatial transcriptomics data.

### Usage

```
fit_powerest(power, avg_log2FC, avg_PCT, filter_zero=TRUE)
```

### Arguments

power	The raw power values.
avg_log2FC	The corresponding log2FC values.
avg_PCT	The corresponding PCT values.
filter_zero	Whether the user would like to filter to remove the power values being 0, default=TRUE.

### Value

A 'scam' object is the result of scam function. More information about the content of a 'scam' object can be found at the document of R package **scam**.

### Author(s)

Lan Shui <lshui@mdanderson.org>

### Examples

```
data(result_example)
b<-fit_powerest(result_example$power,result_example$avg_logFC,result_example$avg_PCT)
```

---

fit_XGBoost	<i>Fit with XGBoost</i>
-------------	-------------------------

---

### Description

This function estimates the power values based on XGBoost under 3-dimensional monotone constraints upon avg\_log2FC, avg\_PCT and replicates. This function is recommended when there exist crossings between power surfaces fitted by 'fit\_powerest' and used for estimating local power values.

**Usage**

```
fit_XGBoost(power, avg_log2FC, avg_PCT, replicates, filter_zero=TRUE,
max.depth=6, eta=0.3, nround=100)
```

**Arguments**

power	The raw power values.
avg_log2FC	The corresponding log2FC values.
avg_PCT	The corresponding PCT values.
replicates	The corresponding replicates number.
filter_zero	Whether the user would like to filter to remove the power values being 0. Default=TRUE.
max.depth	Maximum depth of a tree. Default=6.
eta	control the learning rate: scale the contribution of each tree by a factor of $0 < \eta < 1$ when it is added to the current approximation. Used to prevent overfitting by making the boosting process more conservative. Default=0.3.
nround	Max number of boosting iterations.

**Value**

A object of class 'xgb.Booster'. More information about the content of a 'xgb.Booster' object can be found at the document of R package [xgboost](#).

**Author(s)**

Lan Shui <lshui@mdanderson.org>

**Examples**

```
data(power_example)
# Fit the local power surface of avg_log2FC_abs between 1 and 2
avg_log2FC_abs_1_2<-dplyr::filter(power_example, avg_log2FC_abs>1 & avg_log2FC_abs<2)
# Fit the model
bst<-fit_XGBoost(power_example$power, avg_log2FC=power_example$avg_log2FC_abs,
avg_PCT=power_example$mean_pct, replicates=power_example$sample_size)
```

---

plotly\_powerest

3D interactive visualization

---

**Description**

This function creates 3d interactive plot of the power against other parameters based on 'plot\_ly'.

**Usage**

```
plotly_powerest(pred, opacity=0.8, colors='BrBG', fig_title=NULL)
```

**Arguments**

pred	The result from 'pred_powerest'.
opacity	The opacity of the graph, default=0.8.
colors	The color for the graph, default='BrBG'.
fig_title	The title of the graph, default=NULL.

**Value**

A 3d interactive plot of the power surface. Users can also plot multiple surfaces together to compare them.

**Author(s)**

Lan Shui <lshui@mdanderson.org>

**Examples**

```
data(result_example)
b<-fit_powerest(result_example$power,result_example$avg_logFC,result_example$avg_PCT)
pred <- pred_powerest(b,xlim= c(0,6),ylim=c(0,1))
plotly_powerest(pred,fig_title='Power estimation result')
```

---

PoweREST

*Bootstrap resampling and power calculation upon ST data*


---

**Description**

This function performs bootstrap resampling upon a Seurat subject under each condition to resemble the real dataset which allows the exact power calculation, and perform DE analysis. Users can specify the test they would like to perform for the DE analysis in '...' which should not contain min.pct and logfc.threshold or other parameters attempt to pre-filter genes, as we specify min.pct and logfc.threshold as 0s to calculate power for all the genes available. Therefore it may take one night to run if the ST data owns over thousands of genes. To speed up this process, one may want to try function 'PoweREST\_subset' where the pre-filter of genes are included in this process.

**Usage**

```
PoweREST(Seurat_obj,cond,replicates=1,spots_num,
iteration=100,random_seed=1,pvalue=0.05,...)
```

**Arguments**

Seurat_obj	A <b>Seurat</b> object.
cond	The name of the variable that indicates different conditions which is also stored in the meta.data of the Seurat_obj and should be in character type.
replicates	The number of sample replicates per group.
spots_num	The number of spots per replicate.
iteration	The number of iterations of the resampling.
random_seed	To set a random seed.
pvalue	The pvalue that will be considered significant.
...	DE test to use other than the default Wilcoxon test.

**Value**

A list of values containing the power, average log2FC and percentage of spots detecting the gene among the resampling data, the replicate value and the spots number per slice specified by the user and corresponding genes' name.

**Author(s)**

Lan Shui <lshui@mdanderson.org>

---

PoweREST_gene	<i>Bootstrap resampling and power estimation for one single gene</i>
---------------	--

---

**Description**

This function performs bootstrap resampling upon a Seurat subject under each condition to resemble the real dataset which allows the exact power calculation, and perform DE analysis upon one gene specified by the user. Users can specify the test they would like to perform for the DE analysis in '...'. Note that the results are not multiple testing corrected, therefore should be interpreted carefully.

**Usage**

```
PoweREST_gene(Seurat_obj, cond, replicates=1, spots_num,
gene_name, iteration=100, random_seed=1, pvalue=0.05, ...)
```

**Arguments**

Seurat_obj	A <b>Seurat</b> object.
cond	The name of the variable that indicates different conditions which is also stored in the meta.data of the Seurat_obj and should be in character type.
replicates	The number of sample replicates per group.
spots_num	The number of spots per replicate.
gene_name	Specify the name of gene for power calculation.
iteration	The number of iterations of the resampling.
random_seed	To set a random seed.
pvalue	The pvalue that will be considered significant.
...	DE Test to use other than the default Wilcoxon test.

**Value**

A list of values containing the power, average log2FC and percentage of spots detecting the gene among the resampling data, the replicate value and the spots number per slice specified by the user and corresponding gene's name.

**Author(s)**

Lan Shui <lshui@mdanderson.org>

---

PowerREST_subset	<i>Bootstrap resampling and power calculation for a subset of genes</i>
------------------	---

---

## Description

This function performs bootstrap resampling upon a Seurat subject under each condition to resemble the real dataset which allows the exact power calculation, and perform DE analysis. Similar to 'PoweREST', users can specify the test they would like to perform for the DE analysis in '...' (more test options can be referred to [Seurat](#)). Different to 'PoweREST', users can specify the values of 'min.pct' and 'logfc.threshold' to pre-filter the genes based on their minimum detection rate 'min.pct' and at least X-fold difference (log-scale) ('logfc.threshold') across both groups. But this kind of filtering can miss weaker signals.

## Usage

```
PowerREST_subset(Seurat_obj, cond, replicates=1, spots_num,
  iteration=100, random_seed=1, pvalue=0.05, logfc.threshold = 0.1,
  min.pct = 0.01, ...)
```

## Arguments

Seurat_obj	A <a href="#">Seurat</a> object.
cond	The name of the variable that indicates different conditions which is also stored in the meta.data of the Seurat_obj and should be in character type.
replicates	The number of sample replicates per group.
spots_num	The number of spots per replicate.
iteration	The number of iterations of the resampling.
random_seed	To set a random seed.
pvalue	The pvalue that will be considered significant.
logfc.threshold	For every resampling, limit testing to genes which show, on average, at least X-fold difference (log-scale) between the two groups. Default is 0.1 Increasing logfc.threshold speeds up the function, but can miss weaker signals.
min.pct	For every resampling, only test genes that are detected in a minimum fraction of min.pct spots in either of the two populations. Meant to speed up the function by not testing genes that are very infrequently expressed. Default is 0.01.
...	DE test to use other than the default Wilcoxon test.

## Value

A list of values containing the power, average log2FC and percentage of spots detecting the gene among the resampling data, the replicate value and the spots number per slice specified by the user and the filtered.

## Author(s)

Lan Shui <lshui@mdanderson.org>

---

power_example	<i>An example of power results with multiple replicates number</i>
---------------	--

---

### Description

A subset of power results with multiple replicates number from PoweREST

### Usage

```
power_example
```

### Format

power\_example:

A data frame with 844 rows and 5 columns:

**avg\_logFC** average log2FC

**mean\_PCT** percentage of spots detecting the gene

**sample\_size** number of replicates

**power** power values

**avg\_log2FC\_abs** the absolute value of average log2FC

---

pred_powerest	<i>Power value prediction</i>
---------------	-------------------------------

---

### Description

This function provides the prediction from the **Seurat** object which could be used for visualization by 'plotly\_powerest' and 'vis\_powerest' or the power result for your proposal or research. And it is a modified version of the scam library code predict.scam.

### Usage

```
pred_powerest(x,n.grid=30,xlim=NULL,ylim=NULL)
```

### Arguments

x	A <b>Seurat</b> object.
n.grid	The grid note number within 'xlim' and 'ylim', default=30.
xlim	The range of the absolute value of log2FC used for prediction, default=NULL which means the original range.
ylim	The range of the avg_pct used for prediction, default=NULL which means the original range.

### Value

The prediction values of the power.

**Author(s)**

Lan Shui <lshui@mdanderson.org> based partly on 'scam' by Natalya Pya

**Examples**

```
data(result_example)
b<-fit_powerest(result_example$power,result_example$avg_logFC,result_example$avg_PCT)
pred <- pred_powerest(b,xlim= c(0,6),ylim=c(0,1))
```

---

pred_XGBoost	<i>Prediction results from XGBoost</i>
--------------	--

---

**Description**

This function takes the result from 'fit\_XGBoost' and make predictions.

**Usage**

```
pred_XGBoost(x,n.grid=30,xlim,ylim,replicates)
```

**Arguments**

x	A object of class 'xgb.Booster'.
n.grid	The grid note number within 'xlim' and 'ylim', default=30.
xlim	The range of the absolute value of avg_log2FC used for prediction.
ylim	The range of the avg_pct used for prediction.
replicates	The replicates number.

**Value**

The power estimations from XGBoost.

**Author(s)**

Lan Shui <lshui@mdanderson.org>

**Examples**

```
data(power_example)
# Fit the local power surface of avg_log2FC_abs between 1 and 2
avg_log2FC_abs_1_2<-dplyr::filter(power_example,avg_log2FC_abs>1 & avg_log2FC_abs<2)
# Fit the model
bst<-fit_XGBoost(power_example$power,avg_log2FC=power_example$avg_log2FC_abs,
avg_PCT=power_example$mean_pct,replicates=power_example$sample_size)
pred<-pred_XGBoost(bst,n.grid=30,xlim=c(0,1.5),ylim=c(0,0.1),replicates=3)
```



result\_example

*An example of power results from PoweREST***Description**

A subset of power results from PoweREST by running `PoweREST(Peri,cond='Condition', replicates=5,spots_num=80,iteration=2)`

**Usage**

```
result_example
```

**Format**

```
result_example:
```

A data frame with ~20,000 rows and 3 columns:

**power** power values

**avg\_logFC** average log2FC

**avg\_PCT** percentage of spots detecting the gene

vis\_powerest

*Visualization of the power surface***Description**

This function takes the result from 'pred\_powerest' and plots 2D views of it, supply ticktype="detailed" to get proper axis annotation and is a modified version of the 'scam' library code 'vis.scam'.

**Usage**

```
vis_powerest(x,color="heat",contour.col=NULL,
se=-1,zlim=NULL,n.grid=30,col=NA,plot.type="persp",
nCol=50,...)
```

**Arguments**

x	A <b>scam</b> object.
color	The color of the plot which can be one of the "heat", "topo", "cm", "terrain", "gray" or "bw".
contour.col	The color of the contour plot when using plot.type="contour".
se	If less than or equal to zero then only the predicted surface is plotted, but if greater than zero, then 3 surfaces are plotted, one at the predicted values minus se standard errors, one at the predicted values and one at the predicted values plus se standard errors.
zlim	The range of power value the user want to show.
n.grid	The number of grid nodes in each direction used for calculating the plotted surface.

col	The colors for the facets of the plot. If this is NA then if se>0 the facets are transparent, otherwise the color scheme specified in color is used. If col is not NA then it is used as the facet color.
plot.type	One of "contour" or "persp".
nCol	The number of colors to use in color schemes.
...	Other arguments.

**Value**

A 2d plot of the power surface. More details can be seen at [scam](#).

**Author(s)**

Lan Shui <lshui@mdanderson.org> based partly on 'scam' by Natalya Pya

**Examples**

```
data(result_example)
b<-fit_powerest(result_example$power,result_example$avg_logFC,result_example$avg_PCT)
pred <- pred_powerest(b,xlim= c(0,6),ylim=c(0,1))
vis_powerest(pred,theta=-30,phi=30,color='heat',ticktype = "detailed",xlim=c(0,6),nticks=5)
```

vis\_XGBoost

*Visualization of the power estimations from XGBoost***Description**

This function takes the result from 'pred\_XGboost' and plots 2D/3D views of it,

**Usage**

```
vis_XGBoost(x,view='2D',legend_name='Power',
xlab='avg_log2FC_abs',ylab='mean_pct')
```

**Arguments**

x	The result dataframe from 'pred_XGboost'.
view	determines plot 2D/3D view, default='2D'.
legend_name	The name of legend, default='Power'.
xlab	The name of xlab, default='avg_log2FC_abs'.
ylab	The name of ylab, default='mean_pct'.

**Value**

A 2D/3D plot of the power results from XGBoost.

**Author(s)**

Lan Shui <lshui@mdanderson.org>

**Examples**

```
data(power_example)
# Fit the local power surface of avg_log2FC_abs between 1 and 2
avg_log2FC_abs_1_2<-dplyr::filter(power_example,avg_log2FC_abs>1 & avg_log2FC_abs<2)
# Fit the model
bst<-fit_XGBoost(power_example$power,avg_log2FC=power_example$avg_log2FC_abs,
avg_PCT=power_example$mean_pct,replicates=power_example$sample_size)
pred<-pred_XGBoost(bst,n.grid=30,xlim=c(0,1.5),ylim=c(0,0.1),replicates=3)
vis_XGBoost(pred,view='2D',legend_name='Power',xlab='avg_log2FC_abs',ylab='mean_pct')
```

# Index

## \* datasets

- power\_example, [7](#)
- result\_example, [9](#)

- fit\_powerest, [2](#)
- fit\_XGBoost, [2](#)

- plotly\_powerest, [3](#)
- power\_example, [7](#)
- PowerREST, [4](#)
- PowerREST\_gene, [5](#)
- PowerREST\_subset, [6](#)
- pred\_powerest, [7](#)
- pred\_XGBoost, [8](#)

- result\_example, [9](#)

- vis\_powerest, [9](#)
- vis\_XGBoost, [10](#)