

# Package ‘ZetaSuite’

September 24, 2025

**Type** Package

**Title** Analyze High-Dimensional High-Throughput Dataset and Quality Control Single-Cell RNA-Seq

**Version** 1.0.2

**Date** 2025-09-24

**Maintainer** Junhui Li <ljh.biostat@gmail.com>

**Description** The advent of genomic technologies has enabled the generation of two-dimensional or even multi-dimensional high-throughput data, e.g., monitoring multiple changes in gene expression in genome-wide siRNA screens across many different cell types (E Robert McDonald 3rd (2017) <[doi:10.1016/j.cell.2017.07.005](https://doi.org/10.1016/j.cell.2017.07.005)> and Tsherniak A (2017) <[doi:10.1016/j.cell.2017.06.010](https://doi.org/10.1016/j.cell.2017.06.010)>) or single cell transcriptomics under different experimental conditions. We found that simple computational methods based on a single statistical criterion is no longer adequate for analyzing such multi-dimensional data. We herein introduce 'ZetaSuite', a statistical package initially designed to score hits from two-dimensional RNAi screens. We also illustrate a unique utility of 'ZetaSuite' in analyzing single cell transcriptomics to differentiate rare cells from damaged ones (Vento-Tormo R (2018) <[doi:10.1038/s41586-018-0698-6](https://doi.org/10.1038/s41586-018-0698-6)>). In 'ZetaSuite', we have the following steps: QC of input datasets, normalization using Z-transformation, Zeta score calculation and hits selection based on defined Screen Strength.

**BugReports** <https://github.com/JunhuiLi1017/ZetaSuite/issues>

**Imports** RColorBrewer, Rtsne, e1071, ggplot2, reshape2, gridExtra, mixtools, shinyjs, shinydashboard, shiny, plotly, DT

**License** MIT + file LICENSE

**Depends** R (>= 2.10)

**RoxygenNote** 7.3.2

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Author** Yajing Hao [aut] (ORCID: <<https://orcid.org/0000-0003-1384-4176>>), Shuyang Zhang [ctb] (ORCID: <<https://orcid.org/0000-0002-8428-1828>>), Junhui Li [cre] (ORCID: <<https://orcid.org/0000-0003-3973-1700>>), Guofeng Zhao [ctb], Xiang-Dong Fu [cph, fnd] (ORCID: <<https://orcid.org/0000-0001-5499-8732>>)

**NeedsCompilation** no  
**Repository** CRAN  
**Date/Publication** 2025-09-24 21:00:02 UTC  
**Encoding** UTF-8

**Contents**

countMat . . . . .	2
countMatSC . . . . .	3
EventCoverage . . . . .	4
FDRcutoff . . . . .	5
negGene . . . . .	7
nonExpGene . . . . .	8
posGene . . . . .	8
QC . . . . .	9
SVM . . . . .	10
SVMcurve . . . . .	12
Zeta . . . . .	12
ZetaSuiteApp . . . . .	14
ZetaSuitSC . . . . .	15
Zscore . . . . .	16
ZseqList . . . . .	17
<b>Index</b>	<b>19</b>

---

countMat	<i>Subsampled data from in-house HTS2 screening for global splicing regulators.</i>
----------	---

---

**Description**

A data frame with 1609 individual screened genes and 100 functional readouts. The data was generated from a siRNA screen for global splicing regulators. In this screen, we interrogated ~400 endogenous alternative splicing (AS) events by using an oligo ligation-based strategy to quantify 18,480 pools of siRNAs against annotated protein-coding genes in the human genome.

**Usage**

```
data("countMat")
```

**Format**

A data frame with 1609 observations on the following 100 variables  
A data frame with 1609 observations on the following 100 maker variables.Each row represents gene with specific knocking-down siRNA pool, each column is an AS event. The values in the matrix are the processed foldchange values between included exons and skipping exons read counts.

**Details**

This data frame is the raw output data from large-scale screening.

**Examples**

```
data(countMat)
```

---

countMatSC	<i>The cell x gene matrix from single-cell RNA-seq.</i>
------------	---

---

**Description**

A scRNA-seq dataset generated from placenta that has been analyzed with CellRanger and used to develop EmptyDrops. We have subsampled the genes from the real datasets to generate the matrix.

**Usage**

```
data("countMatSC")
```

**Format**

A data frame with 1090 cells and 10000 genes. This is the subset of data obtained from single-cell RNAseq for package testing. Each row represents one cell detected in single-cell RNA-seq, each column is one gene in detected cells. The values in the matrix are the raw read counts from single-cell RNAseq.

A data frame with 1090 cells and 10000 genes. This is the subset of data obtained from single-cell RNAseq for package testing.

**Details**

This data frame is generated by single-cell RNA-seq.

**Examples**

```
data(countMatSC)
```

---

EventCoverage	<i>Generate event coverage analysis and visualization for alternative splicing data.</i>
---------------	--

---

## Description

This function analyzes event coverage across Z-score thresholds and generates visualizations to compare positive and negative control samples. It calculates the proportion of readouts that exceed different Z-score thresholds for each gene, creating the foundation for zeta score calculations.

## Usage

```
EventCoverage(ZscoreVal, negGene, posGene, binNum, combine = TRUE)
```

## Arguments

ZscoreVal	A matrix of Z-scores where rows represent genes and columns represent readouts/conditions. This is typically the output from the Zscore() function.
negGene	A data frame or matrix containing negative control gene/siRNA identifiers. The first column should contain gene/siRNA names that match the row names in ZscoreVal.
posGene	A data frame or matrix containing positive control gene/siRNA identifiers. The first column should contain gene/siRNA names that match the row names in ZscoreVal.
binNum	The number of bins to divide the Z-score range. Recommended value is 100. The function creates Z-score thresholds from the 0.00001 to 0.99999 quantiles of the data.
combine	Logical. Whether to combine the negative and positive Z-score ranges. Default is TRUE. When TRUE, uses symmetric ranges around zero; when FALSE, uses separate ranges for negative and positive values.

## Details

The function performs the following steps:

1. Determines Z-score thresholds based on data quantiles and binNum
2. For each gene and threshold, calculates the proportion of readouts that exceed (increase) or fall below (decrease) the threshold
3. Separates data into negative and positive control groups
4. Generates jitter plots comparing event coverage between control groups
5. Returns both data matrices and visualization plots

The event coverage matrices can be used as input for SVM analysis and zeta score calculations.

**Value**

A list containing two sublists:

- |            |  |
|------------|--|
| ECdataList | <p>A list with the following components:</p> <ul style="list-style-type: none"> <li>• ZseqList: A data frame with Z-score thresholds for decrease (Zseq_D) and increase (Zseq_I) directions</li> <li>• EC_N_I: Event coverage matrix for negative controls in increase direction</li> <li>• EC_N_D: Event coverage matrix for negative controls in decrease direction</li> <li>• EC_P_I: Event coverage matrix for positive controls in increase direction</li> <li>• EC_P_D: Event coverage matrix for positive controls in decrease direction</li> </ul> |
| ECplotList | <p>A list with two ggplot objects:</p> <ul style="list-style-type: none"> <li>• EC_jitter_D: Jitter plot showing event coverage for decrease direction</li> <li>• EC_jitter_I: Jitter plot showing event coverage for increase direction</li> </ul>  |

**Author(s)**

Yajing Hao, Shuyang Zhang, Junhui Li, Guofeng Zhao, Xiang-Dong Fu

**Examples**

```
data(countMat)
data(negGene)
data(posGene)
ZscoreVal <- Zscore(countMat, negGene)
ECList <- EventCoverage(ZscoreVal, negGene, posGene, binNum=100, combine=TRUE)
```

---

FDRcutoff	<i>Determine optimal cutoff thresholds based on Screen Strength analysis.</i>
-----------	---

---

**Description**

This function calculates optimal cutoff thresholds for identifying significant hits in high-throughput screening data using Screen Strength (SS) analysis. It evaluates the trade-off between sensitivity and specificity by calculating the ratio of apparent FDR to baseline FDR across different zeta score thresholds.

**Usage**

```
FDRcutoff(zetaData, negGene, posGene, nonExpGene, combine = FALSE)
```

## Arguments

zetaData	A data frame containing zeta scores calculated by the Zeta() function. Should have columns 'Zeta_D' and 'Zeta_I' representing decrease and increase direction scores, respectively.
negGene	A data frame or matrix containing negative control gene/siRNA identifiers. The first column should contain gene/siRNA names that match the row names in zetaData.
posGene	A data frame or matrix containing positive control gene/siRNA identifiers. The first column should contain gene/siRNA names that match the row names in zetaData.
nonExpGene	A data frame or matrix containing non-expressed gene/siRNA identifiers. These genes are used to estimate the baseline false discovery rate. The first column should contain gene/siRNA names that match the row names in zetaData.
combine	Logical. Whether to combine decrease and increase direction zeta scores. Default is FALSE. When TRUE, uses the sum of Zeta_D and Zeta_I; when FALSE, analyzes each direction separately.

## Details

The function performs the following analysis:

1. Categorizes genes into types: "Gene" (test genes), "Positive" (positive controls), "NS\_mix" (negative controls), and "non\_exp" (non-expressed genes)
2. Calculates baseline FDR (bFDR) as the proportion of non-expressed genes in the entire dataset
3. For each zeta score threshold, calculates apparent FDR (aFDR) as the proportion of non-expressed genes among hits
4. Computes Screen Strength:  $SS = 1 - (aFDR / bFDR)$
5. Generates plots showing zeta score distributions and SS curves

Higher Screen Strength values indicate better separation between true hits and false positives. Users can select appropriate thresholds based on desired sensitivity/specificity trade-offs.

## Value

A list containing:

FDR_cutOff	<p>A data frame with 6 columns:</p> <ul style="list-style-type: none"> <li>• Cut_Off: Zeta score threshold</li> <li>• aFDR: Apparent false discovery rate at this threshold</li> <li>• SS: Screen Strength = <math>1 - (aFDR / bFDR)</math></li> <li>• TotalHits: Total number of hits at this threshold</li> <li>• Num_nonExp: Number of non-expressed genes among hits</li> <li>• Type: Direction ("Decrease", "Increase", or "Combine")</li> </ul>
plotList	<p>A list with two ggplot objects:</p> <ul style="list-style-type: none"> <li>• Zeta_type: Jitter plots showing zeta score distributions by gene type</li> <li>• SS_cutOff: Screen Strength curves showing SS vs. zeta score threshold</li> </ul>

**Author(s)**

Yajing Hao, Shuyang Zhang, Junhui Li, Guofeng Zhao, Xiang-Dong Fu

**Examples**

```
data(nonExpGene)
data(negGene)
data(posGene)
data(ZseqList)
data(countMat)
ZscoreVal <- Zscore(countMat, negGene)
zetaData <- Zeta(ZscoreVal, ZseqList, SVM=FALSE)
cutoffval <- FDRcutoff(zetaData, negGene, posGene, nonExpGene, combine=TRUE)
```

---

negGene

*Input negative file.*

---

**Description**

A data frame with 510 different well IDs in which the cells treated with non-specific siRNAs. If users did not have the build-in negative controls, the non-expressed genes should be provided here.

**Usage**

```
data("negGene")
```

**Format**

A data frame with 510 different well IDs in which the cells treated with non-specific siRNAs.

A data frame with 510 different well IDs in which the cells treated with non-specific siRNAs.  
These wells were served as negative control.

**Details**

These wells were designed by the authors in the large-scale screen.

**Examples**

```
data(negGene)
```

---

nonExpGene	<i>Input internal negative control file.</i>
------------	--

---

### Description

A data frame with 722 different well IDs in which the cells treated with siRNAs targeting to non-expressed genes in HeLa cells. It is the subset of total non-expressed genes in HeLa cells.

### Usage

```
data("nonExpGene")
```

### Format

A data frame with 722 different well IDs in which the cells treated with siRNAs targeting to non-expressed genes in HeLa cells.

A data frame with 722 different well IDs in which the cells treated with siRNAs targeting to non-expressed genes in HeLa cells. These wells were served as internal negative controls.

### Details

These non-expressed genes can be obtained from a prior expression profile.

### Examples

```
data(nonExpGene)
```

---

posGene	<i>Input positive file.</i>
---------	-----------------------------

---

### Description

A data frame with 299 different well IDs in which the cells treated with siRNAs targeting to PTB. If users didn't have the build-in positive controls, choose the parameters `-withoutsvm` and the filename can use any name such as 'NA'.

### Usage

```
data("negGene")
```

### Format

A data frame with 299 different well IDs in which the cells treated with siRNAs targeting to PTB.

A data frame with 299 different well IDs in which the cells treated with siRNAs targeting to PTB. These wells were served as positive control.



**Details**

These wells were designed by the authors in the large-scale screen.

**Examples**

```
data(posGene)
```

---

QC

---

*Perform quality control analysis for high-throughput screening data.*


---

**Description**

This function performs comprehensive quality control analysis on high-throughput screening data to evaluate experimental design and data quality. It generates multiple diagnostic plots and calculates SSMD (Strictly Standardized Mean Difference) scores to assess the separation between positive and negative controls.

**Usage**

```
QC(countMat, negGene, posGene)
```

**Arguments**

countMat	A matrix of raw count data where rows represent genes/siRNAs and columns represent readouts/conditions. The matrix should have row names corresponding to gene/siRNA identifiers.
negGene	A data frame or matrix containing negative control gene/siRNA identifiers. The first column should contain gene/siRNA names that match the row names in countMat.
posGene	A data frame or matrix containing positive control gene/siRNA identifiers. The first column should contain gene/siRNA names that match the row names in countMat.

**Details**

The function performs the following quality control analyses:

1. Creates jitter plots to visualize score distributions across readouts
2. Performs t-SNE dimensionality reduction to assess global sample separation
3. Generates boxplots to compare score distributions between control groups
4. Calculates SSMD scores for each readout:  $SSMD = (\mu_{pos} - \mu_{neg}) / \sqrt{\sigma_{pos}^2 + \sigma_{neg}^2}$
5. Reports the percentage of readouts with  $|SSMD| \geq 2$  (considered high quality)

SSMD scores  $\geq 2$  indicate good separation between positive and negative controls, suggesting high-quality readouts.

**Value**

A list containing four diagnostic plots:

score_qc	A jitter plot showing the distribution of raw scores across all readouts for positive and negative controls
tSNE_QC	A t-SNE plot showing the global separation of positive and negative control samples in 2D space
QC_box	Side-by-side boxplots showing the distribution of scores for positive and negative controls across all readouts
QC_SSMD	A density plot showing the distribution of SSMD scores across readouts, with a threshold line at SSMD=2 and the percentage of high-quality readouts displayed

**Author(s)**

Yajing Hao, Shuyang Zhang, Junhui Li, Guofeng Zhao, Xiang-Dong Fu

**References**

Laurens van der Maaten & Geoffrey Hinton: Visualizing Data using t-SNE. Journal of Machine Learning Research 2008, 9(2008):2579-2605.

Zhang XD: A pair of new statistical parameters for quality control in RNA interference high-throughput screening assays. Genomics 2007, 89:552-561.

**Examples**

```
data(countMat)
data(negGene)
data(posGene)
QC(countMat, negGene, posGene)
```

---

SVM

*Generate SVM decision boundaries for positive and negative control separation.*

---

**Description**

This function constructs Support Vector Machine (SVM) models to find optimal decision boundaries that separate positive and negative control samples in event coverage space. It uses radial kernel SVM to create non-linear decision boundaries for both decrease and increase directions.

**Usage**

```
SVM(ECdataList)
```

## Arguments

- ECdataList** A list containing event coverage data from the EventCoverage() function. The list should contain:
- EC\_N\_D: Event coverage matrix for negative controls in decrease direction
  - EC\_P\_D: Event coverage matrix for positive controls in decrease direction
  - EC\_N\_I: Event coverage matrix for negative controls in increase direction
  - EC\_P\_I: Event coverage matrix for positive controls in increase direction
  - ZseqList: Z-score thresholds for both directions

## Details

The function performs the following steps:

1. Prepares training data by combining positive and negative control event coverage data
2. Trains separate SVM models for decrease and increase directions using radial kernel
3. Uses pre-tuned hyperparameters (cost=20, gamma=3 for decrease; cost=50, gamma=2 for increase)
4. Generates prediction grids across the Z-score and event coverage space
5. Identifies decision boundary points where the SVM prediction changes from negative to positive
6. Returns the optimal threshold points for each Z-score bin

The resulting SVM curves can be used for background correction in zeta score calculations to improve the accuracy of hit identification.

## Value

A list containing two data frames:

- cutOffD** A data frame with SVM decision boundary points for decrease direction. Each row contains a Z-score threshold and the corresponding event coverage threshold that separates positive and negative controls.
- cutOffI** A data frame with SVM decision boundary points for increase direction. Each row contains a Z-score threshold and the corresponding event coverage threshold that separates positive and negative controls.

## Author(s)

Yajing Hao, Shuyang Zhang, Junhui Li, Guofeng Zhao, Xiang-Dong Fu

## Examples

```
data(countMat)
data(negGene)
data(posGene)
ZscoreVal <- Zscore(countMat, negGene)
ECdataList <- EventCoverage(ZscoreVal, negGene, posGene, binNum=10, combine=TRUE)
SVM(ECdataList)
```

---

SVMcurve

*The SVM curve lines in Zeta-plot.*


---

### Description

The SVM curves were calculated from raw input matrix files. They were designed to maximally separate the positive and negative genes.

### Usage

```
data("SVMcurve")
```

### Format

A data frame with 24 rows and 4 features.

A data frame with 24 rows and 4 features. The first column is the bins cut-offs for decreased direction. The second column is the values of percentage with different cut-offs in column 1. The third column is the bins cut-offs for increased direction. The fourth column is the values of percentage with different cut-offs in column 3.

### Details

This data frame is generated by SVM.R.

### Examples

```
data(SVMcurve)
```

---

Zeta

*Calculation of zeta and weighted zeta score.*


---

### Description

This function calculates zeta scores for genes based on their Z-score profiles across different thresholds. The zeta score quantifies the regulatory effect of gene knockdown on alternative splicing events by measuring the area under the curve of event coverage across Z-score thresholds.

### Usage

```
Zeta(ZscoreVal, ZseqList, SVMcurve = NULL, SVM = FALSE)
```

**Arguments**

ZscoreVal	A matrix of Z-scores where rows represent genes and columns represent read-outs/conditions. This is typically the output from the Zscore() function.
ZseqList	A list containing two vectors: 'Zseq_D' (decrease direction thresholds) and 'Zseq_I' (increase direction thresholds). These define the Z-score bins for calculating event coverage.
SVMcurve	Optional. A matrix containing SVM curve data for decrease and increase directions. Required only when SVM=TRUE. The matrix should have 4 columns: Z-score and coverage for decrease direction (columns 1-2), and Z-score and coverage for increase direction (columns 3-4).
SVM	Logical. Whether to use SVM curves for background correction. Default is FALSE. When TRUE, the function subtracts SVM-predicted background from the event coverage before calculating zeta scores.

**Details**

The function calculates zeta scores as follows:

1. For each Z-score threshold, calculates the proportion of readouts that exceed (increase) or fall below (decrease) the threshold
2. Computes the area under the event coverage curve using trapezoidal integration
3. If SVM=TRUE, subtracts SVM-predicted background coverage before area calculation
4. Returns separate scores for decrease (Zeta\_D) and increase (Zeta\_I) directions

Higher zeta scores indicate stronger regulatory effects on alternative splicing.

**Value**

A data frame with two columns:

Zeta_D	Zeta score for decrease direction (exon skipping events)
Zeta_I	Zeta score for increase direction (exon inclusion events)

Each row corresponds to a gene, and the zeta scores represent the cumulative regulatory effect across all Z-score thresholds.

**Author(s)**

Yajing Hao, Shuyang Zhang, Junhui Li, Guofeng Zhao, Xiang-Dong Fu

**Examples**

```
data(ZseqList)
data(SVMcurve)
data(countMat)
data(negGene)
ZscoreVal <- Zscore(countMat, negGene)
zetaData <- Zeta(ZscoreVal, ZseqList, SVM=FALSE)
```

---

**ZetaSuiteApp***Launch ZetaSuite Shiny Application*

---

**Description**

Launches the ZetaSuite Shiny web application for interactive analysis of high-throughput screening data and single-cell RNA-seq quality control.

**Usage**

```
ZetaSuiteApp(launch.browser = TRUE, port = NULL, host = "127.0.0.1")
```

**Arguments**

<code>launch.browser</code>	Logical. Should the app launch in the default browser? Default is TRUE.
<code>port</code>	Integer. Port number for the Shiny app. Default is NULL (random port).
<code>host</code>	Character. Host address. Default is "127.0.0.1" (localhost).

**Details**

The Shiny app provides a user-friendly interface for:

- Quality Control Analysis
- Z-score Normalization
- Event Coverage Analysis
- Zeta Score Calculation
- SVM-based Background Correction
- Screen Strength Analysis
- Single Cell Quality Control
- Interactive visualizations and data export

**Value**

Launches the Shiny application in a web browser.

**Examples**

```
## Not run:  
# Launch the ZetaSuite Shiny app  
ZetaSuiteApp()  
  
# Launch without opening browser automatically  
ZetaSuiteApp(launch.browser = FALSE)  
  
# Launch on a specific port  
ZetaSuiteApp(port = 3838)
```

```
## End(Not run)
```

---

ZetaSuitSC

---

*Calculate zeta score for single cell RNA-seq quality control.*


---

## Description

This function evaluates the quality of cells detected in single-cell RNA-seq data by calculating a zeta score for each cell. The zeta score is based on the distribution of gene expression across different expression thresholds. A cutoff value is automatically determined using a two-component Gaussian mixture model to separate high-quality cells from low-quality or damaged cells.

## Usage

```
ZetaSuitSC(countMatSC, binNum = 10, filter = TRUE)
```

## Arguments

countMatSC	A matrix of single-cell RNA-seq count data where rows represent cells and columns represent genes.
binNum	The number of bins for zeta score calculation. Default is 10. The function creates expression thresholds from 0 to the 80th percentile of non-zero expression values, divided into binNum intervals.
filter	Logical. Whether to filter out cells with total read counts less than 100. Default is TRUE. This helps remove extremely low-quality cells before analysis.

## Details

The function works as follows:

1. Filters cells based on total read count if filter=TRUE
2. Samples a subset of cells and genes for computational efficiency
3. Creates expression thresholds (bins) from 0 to the 80th percentile of non-zero expression values
4. For each cell, counts how many genes exceed each threshold
5. Calculates the zeta score as a weighted sum of these counts
6. Fits a two-component Gaussian mixture model to log10-transformed zeta scores
7. Determines an optimal cutoff to separate high-quality from low-quality cells

**Value**

- A list containing:
- zetaData            A data frame with two columns: 'Cell' (cell identifiers) and 'Zeta' (calculated zeta scores)
  - p\_cutoff           A ggplot object showing the distribution of log10-transformed zeta scores with fitted Gaussian mixture components and the determined cutoff threshold

**Author(s)**

Yajing Hao, Shuyang Zhang, Junhui Li, Guofeng Zhao, Xiang-Dong Fu

**Examples**

```
data(countMatSC)
zetaDataSC <- ZetaSuitSC(countMatSC, binNum=50, filter=TRUE)
```

---

Zscore	<i>Z-score normalization for high-throughput screening data.</i>
--------	--

---

**Description**

This function performs Z-score normalization on high-throughput screening data using negative control samples as reference. The Z-score transformation standardizes the data by centering and scaling each column (readout) based on the mean and standard deviation of negative control samples.

**Usage**

```
Zscore(countMat, negGene)
```

**Arguments**

- countMat           A matrix of raw count data where rows represent genes/siRNAs and columns represent readouts/conditions. The matrix should have row names corresponding to gene/siRNA identifiers.
- negGene            A data frame or matrix containing negative control gene/siRNA identifiers. The first column should contain the gene/siRNA names that match the row names in countMat.



## Details

The function performs Z-score normalization as follows:

1. Extracts negative control samples from the input matrix using the identifiers provided in negGene
2. For each column (readout), calculates the mean and standard deviation using only the negative control samples
3. Applies Z-score transformation:  $Z_{ij} = (X_{ij} - \mu_j) / \sigma_j$  where  $X_{ij}$  is the raw value for gene  $i$  in readout  $j$ ,  $\mu_j$  is the mean of negative controls in readout  $j$ , and  $\sigma_j$  is the standard deviation of negative controls in readout  $j$

This normalization allows for comparison across different readouts and identifies genes/siRNAs that show significant deviation from the negative control distribution.

## Value

A Z-score normalized matrix with the same dimensions as the input countMat (excluding the Type column added during processing). Each value represents how many standard deviations away from the negative control mean that particular gene/readout combination is.

## Author(s)

Yajing Hao, Shuyang Zhang, Junhui Li, Guofeng Zhao, Xiang-Dong Fu

## Examples

```
data(countMat)
data(negGene)
ZscoreVal <- Zscore(countMat, negGene)
ZscoreVal[1:5, 1:5]
```

---

ZseqList

*The bin size for Zeta calculation.*


---

## Description

A data frame with 11 different cut-offs and 2 directions. We divided the ranges of input values into bins. The number of bins is determined by the users.

## Usage

```
data("ZseqList")
```

## Format

A data frame with 11 different cut-offs and 2 directions.

A data frame with 11 different cut-offs and 2 directions. We divided the ranges of input values into bins. The number of bins is determined by the users.

**Details**

This data frame is the generated by EventCoverage.R.

**Examples**

```
data(ZseqList)
```

# Index

- \* **FDR**
    - FDRcutoff, [5](#)
  - \* **SSMD**
    - QC, [9](#)
  - \* **SVM**
    - SVM, [10](#)
  - \* **Z-score**
    - Zscore, [16](#)
  - \* **ZetaSuite**
    - EventCoverage, [4](#)
    - FDRcutoff, [5](#)
    - QC, [9](#)
    - SVM, [10](#)
    - Zeta, [12](#)
    - ZetaSuitSC, [15](#)
    - Zscore, [16](#)
  - \* **alternative**
    - EventCoverage, [4](#)
    - Zeta, [12](#)
  - \* **boundary**
    - SVM, [10](#)
  - \* **cell**
    - ZetaSuitSC, [15](#)
  - \* **control**
    - QC, [9](#)
    - ZetaSuitSC, [15](#)
  - \* **coverage**
    - EventCoverage, [4](#)
  - \* **cutoff**
    - FDRcutoff, [5](#)
  - \* **datasets**
    - countMat, [2](#)
    - countMatSC, [3](#)
    - negGene, [7](#)
    - nonExpGene, [8](#)
    - posGene, [8](#)
    - SVMcurve, [12](#)
    - ZseqList, [17](#)
  - \* **decision**
    - SVM, [10](#)
  - \* **event**
    - EventCoverage, [4](#)
  - \* **machine**
    - SVM, [10](#)
  - \* **normalization**
    - Zscore, [16](#)
  - \* **quality**
    - QC, [9](#)
    - ZetaSuitSC, [15](#)
  - \* **score**
    - Zeta, [12](#)
  - \* **screen**
    - FDRcutoff, [5](#)
  - \* **single**
    - ZetaSuitSC, [15](#)
  - \* **splicing**
    - EventCoverage, [4](#)
    - Zeta, [12](#)
  - \* **strength**
    - FDRcutoff, [5](#)
  - \* **support**
    - SVM, [10](#)
  - \* **t-SNE**
    - QC, [9](#)
  - \* **vector**
    - SVM, [10](#)
  - \* **zeta**
    - Zeta, [12](#)
- countMat, [2](#)  
countMatSC, [3](#)  
  
EventCoverage, [4](#)  
  
FDRcutoff, [5](#)  
  
negGene, [7](#)  
nonExpGene, [8](#)  
  
posGene, [8](#)

QC, [9](#)

SVM, [10](#)

SVMcurve, [12](#)

Zeta, [12](#)

ZetaSuiteApp, [14](#)

ZetaSuitSC, [15](#)

Zscore, [16](#)

ZseqList, [17](#)