# Package 'kmcut'

November 12, 2026

**Type** Package

**Title** Optimized Kaplan Meier analysis and identification and validation of prognostic biomarkers

**Version** 1.6.0

**Description** The purpose of the package is to identify prognostic biomarkers and an optimal numeric cutoff for each biomarker that can be used to stratify a group of test subjects (samples) into two sub-groups with significantly different survival (better vs. worse). The package was developed for the analysis of gene expression data, such as RNA-seq. However, it can be used with any quantitative variable that has a sufficiently large proportion of unique values.

**License** Artistic-2.0

**Encoding** UTF-8

**LazyData** false

**Imports** survival, tools, methods, pracma, doParallel, foreach, parallel, SummarizedExperiment, S4Vectors

**RoxygenNote** 7.2.3

**Suggests** BiocStyle, knitr, rmarkdown,

**VignetteBuilder** knitr

**biocViews** Software, StatisticalMethod, GeneExpression, Survival

**git_url** https://git.bioconductor.org/packages/kmcut

**git_branch** RELEASE_3_23

**git_last_commit** 5bd54fd

**git_last_commit_date** 2026-04-28

**Repository** Bioconductor 3.23

**Date/Publication** 2026-11-11

**Author** Igor Kuznetsov [aut, cre], Javed Khan [aut]

**Maintainer** Igor Kuznetsov <ibkalb@gmail.com>

# Contents

create_se_object . . . . . 2
extract_columns . . . . . 3
extract_rows . . . . . 4
km_opt_pcut . . . . . 5
km_opt_scut . . . . . 6
km_qcut . . . . . 8
km_ucut . . . . . 9
km_val_cut . . . . . 11
transpose_table . . . . . 12
ucox_batch . . . . . 13
ucox_pred . . . . . 14

**Index** **17**

---

`create_se_object` *Create SummarizedExperiment object*

---

## Description

Reads a file with expression data and a file with survival data. Then, uses the data to create a SummarizedExperiment object.

## Usage

```
create_se_object(efile, sfile, wdir = getwd())
```

## Arguments

| | |
|---|---|
| `efile` | a character string (character vector of length 1) that specifies the name of the file with expression data for each sample. The file must be tab-delimited, where genes are in rows and samples are in columns. First column must contain gene names. Column names must contain sample ids. |
| `sfile` | a character string (character vector of length 1) that specifies the name of the file with right-censored survival time data. The file must be tab-delimited, where samples are in rows. First column must contain sample ids that match those in 'efile'. The file must contain columns called 'stime' and 'scens', with survival time and censoring variable (0 or 1), respectively. |
| `wdir` | a character string (character vector of length 1) that specifies the name of the working directory for the input files (defaults to the current R directory). |

## Value

a SummarizedExperiment object that contains expression matrix along with survival data as column data.

## Examples

```
# Example with data files included in the package:

# Load example gene expression data and survival data for 2 genes
# and 93 samples:
fdat <- system.file("extdata", "example_genes.txt", package = "kmcut")
sdat <- system.file("extdata", "survival_data.txt", package = "kmcut")

# Create SummarizedExperiment object
se <- create_se_object(efile = fdat, sfile = sdat)
```

---

`extract_columns` *Extract a sub-set of columns*

---

## Description

Extract a sub-set of columns (such as a sub-set of samples) from a data table. All rows will be preserved.

## Usage

```
extract_columns(fnamein, fids, fnameout, wdir = getwd())
```

## Arguments

| | |
|---|---|
| `fnamein` | a character string (character vector of length 1) that specifies the name of tab-delimited text file with the input data table. |
| `fids` | a character string (character vector of length 1) that specifies the name of text file with column ids (such as sample ids). The file must contain one column id per line, without any trailing spaces or any other additional symbols. |
| `fnameout` | a character string (character vector of length 1) that specifies the name of output file where the new data table will be saved. |
| `wdir` | a character string (character vector of length 1) that specifies the name of the working directory for the input/output files (defaults to the current R directory). |

## Value

no return value

## Examples

```
# Example with built-in data files:

# Load example gene expression data table for 2 genes
fdat <- system.file("extdata", "example_genes.txt", package = "kmcut")
# Load a list that contains column (sample) ids
idlist <- system.file("extdata", "columnids.txt", package = "kmcut")
```

```
# Run the function
extract_columns(fnamein = fdat, fids = idlist,
                fnameout = "example_samples_subset.txt")

# This will create in the current working directory a tab-delimited text file
# "example_samples_subset.txt"
```

---

`extract_rows` *Extract a sub-set of rows*

---

## Description

Extract a sub-set of rows (such as a group of gene ids) from a data table. All columns will be preserved.

## Usage

```
extract_rows(fnamein, fids, fnameout, wdir = getwd())
```

## Arguments

| | |
|---|---|
| `fnamein` | a character string (character vector of length 1) that specifies the name of tab-delimited text file with the input data table. |
| `fids` | a character string (character vector of length 1) that specifies the name of text file with row ids (such as gene ids). The file must contain one row id per line, without any trailing spaces or any other additional symbols. |
| `fnameout` | a character string (character vector of length 1) that specifies the name of output file where the new data table will be saved. |
| `wdir` | a character string (character vector of length 1) that specifies the name of the working directory for the input/output files (defaults to the current R directory). |

## Value

no return value

## Examples

```
# Example with built-in data files:

# Load example gene expression data table for 2 genes
fdat <- system.file("extdata", "example_genes.txt", package = "kmcut")
# Load a list that contains one gene id (MYCN)
idlist <- system.file("extdata", "rowids.txt", package = "kmcut")
# Run the function
extract_rows(fnamein = fdat, fids = idlist,
             fnameout = "example_genes_subset.txt")

# This will create in the current working directory a tab-delimited text file
# "example_genes_subset.txt" with one row "MYCN".
```

km_opt_pcut *Find and evaluate optimal stratification cutoffs*

## Description

For each feature, finds a cutoff that optimally stratifies samples into 2 groups, plots Kaplan-Meier survival curves and observed vs. expected optimization plot. Then, performs the permutation test to estimate the statistical significance of the cutoff.

## Usage

```
km_opt_pcut(
  obj,
  bfname,
  wdir = getwd(),
  min_fraction = 0.1,
  min_up_down = 1,
  n_iter = 100,
  peak_tolerance = 0.1,
  psort = FALSE,
  min_uval = 50,
  wlabels = TRUE,
  wpdf = TRUE,
  verbose = TRUE,
  nproc = 1
)
```

## Arguments

| | |
|---|---|
| obj | SummarizedExperiment object with expression-like data |
| bfname | a character string (character vector of length 1) that specifies the base name used to construct output files, which are created by adding 'KMoptp_minf_.2f_iter_d' and corresponding extension to 'bfname'. |
| wdir | a character string (character vector of length 1) that specifies the name of the working directory for the output files (defaults to the current R directory). |
| min_fraction | numeric value that specifies the minimal fraction of samples in the smaller group (default is 0.1). |
| min_up_down | numeric value that specifies the minimal number of up/down points on either side of the peak for pracma::findpeaks function (default is 1). |
| n_iter | numeric value that specifies the number of iterations for the permutation test. The default is n_iter=100 for fast calculations. Recommended is n_iter=10000 (slow, especially for a large number of samples/features). |
| peak_tolerance | numeric value that specifies the maximal difference in height between top peaks. The peak within 'peak_tolerance' closest to the median value is selected. |

| | |
|---|---|
| psort | logical value whether to sort the output table by p-values in increasing order (default is FALSE). |
| min_uval | numeric value that specifies the minimal percentage of unique values per feature (default is 50). Features that have less than 'min_uval' percent unique values are excluded from the analysis. |
| wlabels | logical value whether to write a CSV file with low/high (below/above the cutoff) group sample labels (default is TRUE). |
| wpdf | logical value whether to write a PDF file with plots (default is TRUE). |
| verbose | logical value whether to print progress (default is TRUE). |
| nproc | integer value that specifies the number of logical processors (default is 1, meaning execute sequentially). |

## Value

no return value

## Examples

```
# Example with data files included in the package:

# Load example gene expression data and survival data for 2 genes
# and 93 samples:
fdat <- system.file("extdata", "example_genes.txt", package = "kmcut")
sdat <- system.file("extdata", "survival_data.txt", package = "kmcut")

#' # Create SummarizedExperiment object
se <- create_se_object(efile = fdat, sfile = sdat)

# Search for optimal cutoffs and run the permutation tests on 1 CPU
km_opt_pcut(obj = se, bfname = "test", wpdf = FALSE, n_iter = 10)

# This will create two output files in the current R working directory:
# 1) Tab-delimited text file with the results:
# "test_KMoptp_minf_0.10_iter_10.txt"
# 2) CSV file with low/high sample labels:
# "test_KMoptp_minf_0.10_iter_10_labels.csv"
```

---

# km_opt_scut *Find optimal stratification cutoffs*

---

## Description

For each feature, finds a cutoff that optimally stratifies samples into 2 groups, plots Kaplan-Meier survival curves and observed vs. expected optimization plot. Does not perform the permutation test to estimate the statistical significance of the cutoff.

## Usage

```
km_opt_scut(
  obj,
  bfname,
  wdir = getwd(),
  min_fraction = 0.1,
  min_up_down = 1,
  peak_tolerance = 0.1,
  min_uval = 50,
  psort = FALSE,
  wlabels = TRUE,
  wpdf = TRUE,
  verbose = TRUE
)
```

## Arguments

| | |
|---|---|
| obj | SummarizedExperiment object with expression-like data |
| bfname | a character string (character vector of length 1) that specifies the base name used to create output file names, which are created by adding "_KMopt_minf_.2f" and corresponding extension to 'bfname'. |
| wdir | a character string (character vector of length 1) that specifies the name of the working directory for the input/output files (defaults to the current R directory). |
| min_fraction | numeric value that specifies the minimal fraction of samples in the smaller group (default is 0.1). |
| min_up_down | numeric value that specifies the minimal number of up/down points on either side of the peak for 'pracma::findpeaks' function (default is 1). |
| peak_tolerance | numeric value that specifies the maximal difference between in heigth between top peaks. The peak within 'peak_tolerance' closest to the median value is selected. |
| min_uval | numeric value that specifies the minimal percentage of unique values per feature (default is 50). Features that have less than 'min_uval' percent unique values are excluded from the analysis. |
| psort | logical value whether to sort the output table by p-values in increasing order (default is FALSE). |
| wlabels | logical value whether to write a CSV file with low/high (below/above the cutoff) group sample labels (default is TRUE). |
| wpdf | logical value whether to write a PDF file with plots (default is TRUE). |
| verbose | logical value whether to print progress (default is TRUE). |

## Value

no return value

## Examples

```
# Example with data files included in the package:

# Load example gene expression data and survival data for 2 genes
# and 93 samples
fdat <- system.file("extdata", "example_genes.txt", package = "kmcut")
sdat <- system.file("extdata", "survival_data.txt", package = "kmcut")

# Create SummarizedExperiment object
se <- create_se_object(efile = fdat, sfile = sdat)

# Search for optimal cutoffs
km_opt_scut(obj = se, bfname = "test", wpdf = FALSE)

# This will create two output files in the current working directory:
# 1) Tab-delimited text file with the results:
# "test_KMopt_minf_0.10.txt"
# 2) CSV file with low/high sample labels:
# "test_KMopt_minf_0.10_labels.csv"
```

---

# km_qcut *Apply quantile-based stratification cutoffs*

---

## Description

For each feature uses the cutoff supplied as quantile (in 0 to 100 range) to stratify samples into 2 groups, plots Kaplan-Meier survival curves, and performs the log-rank test.

## Usage

```
km_qcut(
  obj,
  bfname,
  wdir = getwd(),
  quant = 50,
  min_uval = 50,
  psort = FALSE,
  wlabels = TRUE,
  wpdf = TRUE
)
```

## Arguments

| | |
|---|---|
| obj | SummarizedExperiment object with expression-like data |
| bfname | a character string (character vector of length 1) that specifies the base name used to create output file names, which are created by adding "_KM_quant_d" and corresponding extension to 'bfname'. |

| | |
|---|---|
| wdir | a character string (character vector of length 1) that specifies the name of the working directory for the output files (defaults to the current R directory). |
| quant | numeric value that specifies the cutoff quantile for stratification. The default is 50th quantile (the median). |
| min_uval | numeric value that specifies the minimal percentage of unique values per feature (default is 50). Features that have less than 'min_uval' percent unique values are excluded from the analysis. |
| psort | logical value whether to sort the output table by p-values in increasing order (default is FALSE). |
| wlabels | logical value whether to write a CSV file with low/high (below/above the cutoff) group sample labels (default is TRUE). |
| wpdf | logical value whether to write a PDF file with plots (default is TRUE). |

## Value

no return value

## Examples

```
# Example with data files included in the package:

# Load example gene expression data and survival data for 2 genes and
# 93 samples
fdat <- system.file("extdata", "example_genes.txt", package = "kmcut")
sdat <- system.file("extdata", "survival_data.txt", package = "kmcut")

# Create SummarizedExperiment object
se <- create_se_object(efile = fdat, sfile = sdat)

# Apply quantile-based stratification cutoffs
km_qcut(obj = se, bfname = "test", quant = 50, wpdf = FALSE)

# This will create two output files in the current working directory:
# 1) Tab-delimited text file with the results:
# "test_KM_quant_50.txt"
# 2) CSV file with low/high sample labels:
# "test_KM_quant_50_labels.csv"
```

---

# km_ucut *Apply user-supplied stratification cutoff*

---

## Description

For each feature uses the user-supplied cutoff to stratify samples into 2 groups, plots Kaplan-Meier survival curves, and performs the log-rank test.

## Usage

```
km_ucut(
  obj,
  bfname,
  wdir = getwd(),
  cutoff,
  min_uval = 50,
  psort = FALSE,
  wlabels = TRUE,
  wpdf = TRUE
)
```

## Arguments

| | |
|---|---|
| obj | SummarizedExperiment object with expression-like data |
| bfname | a character string (character vector of length 1) that specifies the base name used to create output file names, which are created by adding "_KM_ucut_.2f" and corresponding extension to 'bfname'. |
| wdir | a character string (character vector of length 1) that specifies the name of the working directory for the output files (defaults to the current R directory). |
| cutoff | numeric value that specifies the cutoff value for stratification.The same cutoff is applied to every feature in the dataset. |
| min_uval | numeric value that specifies the minimal percentage of unique values per feature (default is 50). Features that have less than 'min_uval' percent unique values are excluded from the analysis. |
| psort | logical value whether to sort the output table by p-values in increasing order (default is FALSE). |
| wlabels | logical value whether to write a CSV file with low/high (below/above the cutoff) group sample labels (default is TRUE). |
| wpdf | logical value whether to write a PDF file with plots (default is TRUE). |

## Value

no return value

## Examples

```
# Example with data files included in the package:

# Load example gene expression data and survival data for 2 genes
# and 93 samples
fdat <- system.file("extdata", "example_genes.txt", package = "kmcut")
sdat <- system.file("extdata", "survival_data.txt", package = "kmcut")

# Create SummarizedExperiment object
se <- create_se_object(efile = fdat, sfile = sdat)
```

```
# Apply the cutoff of 5
km_ucut(obj = se, bfname = "test", cutoff = 5, min_uval = 90, wpdf = FALSE)

# This will create two output files in the current working directory:
# 1) Tab-delimited text file with the results:
# "test_KM_ucut_5.txt"
# 2) CSV file with low/high sample labels:
# "test_KM_ucut_5_labels.csv"
```

---

km_val_cut *Validate stratification cutoffs on test data*

---

## Description

Creates Kaplan-Meier survival curves for a validation data set by using a file with previously determined stratification cutoffs and performs the log-rank tests.

## Usage

```
km_val_cut(
  infile,
  obj,
  bfname,
  wdir = getwd(),
  min_uval = 50,
  psort = FALSE,
  wlabels = TRUE,
  wpdf = TRUE
)
```

## Arguments

| | |
|---|---|
| infile | a character string (character vector of length 1) that specifies the name of tab-delimited file with the table that contains features and a stratification threshold for each feature (this table is produced by 'km_opt_scut', 'km_opt_pcut', 'km_qcut' or 'km_ucut'). The file with previously determined stratification thresholds must have first two columns named as 'tracking_id' and 'CUTOFF'. The 'tracking_id' column contains feature names, the 'CUTOFF' column contains stratification threshold for each feature. |
| obj | SummarizedExperiment object with test expression |
| bfname | a character string (character vector of length 1) that specifies the base name used to construct output files, which are created by adding '_KM_val' and corresponding extension to 'bfname'. |
| wdir | a character string (character vector of length 1) that specifies the name of the working directory for the input/output files (defaults to the current R directory). |

| | |
|---|---|
| min_uval | numeric value that specifies the minimal percentage of unique values per feature. Features that have less than 'min_uval' percent unique values are excluded from the analysis. |
| psort | logical value whether to sort the output table by p-values in increasing order (default is FALSE). |
| wlabels | logical value whether to write a CSV file with low/high (below/above the cutoff) group sample labels (default is TRUE). |
| wpdf | logical value whether to write a PDF file with plots (default is TRUE). |

## Value

no return value

## Examples

```
# Example with data files included in the package:

# Load training (fdat1) and validation (fdat2) gene expression data
# files and survival data file (sdat).
fdat1 <- system.file("extdata", "expression_data_1.txt", package = "kmcut")
fdat2 <- system.file("extdata", "expression_data_2.txt", package = "kmcut")
sdat <- system.file("extdata", "survival_data.txt", package = "kmcut")

# Create SummarizedExperiment object with training data
se1 <- create_se_object(efile = fdat1, sfile = sdat)

# Run 'km_qcut' on the training data to create a file
# with thresholds "training_data_KM_quant_50.txt".
km_qcut(obj = se1, bfname = "training_data", quant = 50, min_uval = 40)

# Create SummarizedExperiment object with test data
se2 <- create_se_object(efile = fdat2, sfile = sdat)

# Validate the thresholds from "training_data_KM_quant_50.txt" on
# test data in 'se2'.
km_val_cut(infile = "training_data_KM_quant_50.txt", obj = se2,
           bfname = "test", wpdf = FALSE, min_uval = 40)

# This will create two output files in the current working directory:
# 1) Tab-delimited text file with the results:
# "test_KM_val.txt"
# 2) CSV file with low/high sample labels:
# "test_KM_val_labels.csv"
```

---

# transpose_table *Transpose a data table*

---

## Description

Converts table rows to columns and columns to rows.

## Usage

```
transpose_table(fnamein, fnameout, wdir = getwd())
```

## Arguments

| | |
|---|---|
| fnamein | character vector that specifies the name of tab-delimited text file with the input data table. |
| fnameout | character vector that specifies the name of output file where the transposed data table will be saved. |
| wdir | character vector that specifies the name of the working directory for the input/output files (defaults to the current R directory). |

## Value

no return value

## Examples

```
# Example with data files included in the package:

# Load example gene expression data table for 2 genes
fdat <- system.file("extdata", "example_genes.txt", package = "kmcut")

transpose_table(fnamein = fdat,
                fnameout = "example_genes_transposed.txt")

# This will create in the current working directory a tab-delimited text
# file with the transposed table: "example_genes_transposed.txt"
```

---

# ucox_batch *Fit Cox regression models in batch mode*

---

## Description

For each feature, fits a univariate Cox regression and performs the likelihood ratio test.

## Usage

```
ucox_batch(
  obj,
  bfname,
  wdir = getwd(),
  min_uval = 50,
```

```
  psort = FALSE,
  verbose = TRUE
)
```

## Arguments

| | |
|---|---|
| obj | SummarizedExperiment object with expression-like data |
| bfname | a character string (character vector of length 1) that specifies the base name used to create the output file name, which is created by adding '_ucoxbatch.txt' to 'bfname'. |
| wdir | a character string (character vector of length 1) that specifies the name of the working directory for the output file (defaults to the current R directory). |
| min_uval | numeric value that specifies the minimal percentage of unique values per feature (default is 50) Features that have less than 'min_uval' percent unique values are excluded from the analysis. |
| psort | logical value whether to sort the output table by p-values in increasing order (default is FALSE). |
| verbose | logical value whether to print progress (default is TRUE). |

## Value

no return value

## Examples

```
# Example with data files included in the package:

# Load example gene expression data and survival data for 2 genes
# and 93 samples
fdat <- system.file("extdata", "example_genes.txt", package = "kmcut")
sdat <- system.file("extdata", "survival_data.txt", package = "kmcut")

# Create SummarizedExperiment object
se <- create_se_object(efile = fdat, sfile = sdat)

ucox_batch(obj = se, bfname = "test")

# This will create in the current working directory a tab-delimited text
# file with the results: "test_ucoxbatch.txt"
```

---

# ucox_pred *Fit and validate Cox regression models*

---

## Description

For each feature, fits a univariate Cox regression model on training data and then uses the model to predict the risk score for test data.

## Usage

```
ucox_pred(
  obj1,
  obj2,
  bfname,
  wdir = getwd(),
  min_uval = 50,
  psort = FALSE,
  verbose = TRUE
)
```

## Arguments

| | |
|---|---|
| obj1 | SummarizedExperiment object with training expression |
| obj2 | SummarizedExperiment object with test expression |
| bfname | a character string (character vector of length 1) that specifies the base name used to create the output file names, which are created by adding '_cox_train_sum.txt', '_train_scores.txt', and '_test_scores.txt' to 'bfname'. |
| wdir | a character string (character vector of length 1) that specifies the name of the working directory for the output files (defaults to the current R directory). |
| min_uval | numeric value that specifies the minimal percentage of unique values per feature (default is 50). Features that have less than 'min_uval' percent unique values are excluded from the analysis. |
| psort | logical value whether to sort the output table by p-values in increasing order (default is FALSE). |
| verbose | logical value whether to print progress (default is TRUE). |

## Value

no return value

## Examples

```
# Example with data files included in the package:

# Load training (fdat1) and test (fdat2) gene expression data
# files and survival data file (sdat).
fdat1 <- system.file("extdata", "expression_data_1.txt", package = "kmcut")
fdat2 <- system.file("extdata", "expression_data_2.txt", package = "kmcut")
sdat <- system.file("extdata", "survival_data.txt", package = "kmcut")

# Create SummarizedExperiment object with training data
se1 <- create_se_object(efile = fdat1, sfile = sdat)

# Create SummarizedExperiment object with test data
se2 <- create_se_object(efile = fdat2, sfile = sdat)

# Fit Cox model on the training data and use it to calculate the risk
```

```
# scores for the test data.
ucox_pred(obj1 = se1, obj2 = se2, bfname = "demo", min_uval = 90)

# This will create three output files in the current working directory:
# 1) Tab-delimited text file with Cox summary for the training data:
# "demo_cox_train_sum.txt"
# 2) Tab-delimited text file with the risk scores for training data:
# "demo_train_scores.txt"
# 3) Tab-delimited text file with the risk scores for test data:
# "demo_test_scores.txt"
```

# Index

create_se_object, 2

extract_columns, 3
extract_rows, 4

km_opt_pcut, 5
km_opt_scut, 6
km_qcut, 8
km_ucut, 9
km_val_cut, 11

transpose_table, 12

ucox_batch, 13
ucox_pred, 14