# Package 'HVP'

November 12, 2026

**Title** Hierarchical Variance Partitioning

**Version** 1.2.0

**Description** HVP is a quantitative batch effect metric that estimates the proportion of variance associated with batch effects in a data set.

**Imports** Matrix, methods, stats

**Suggests** SingleCellExperiment, SummarizedExperiment, Seurat, SeuratObject, ggplot2, progress, testthat, splatter, scater, devtools, knitr, rmarkdown, BiocStyle, ExperimentHub

**License** MIT + file LICENSE

**VignetteBuilder** knitr

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**biocViews** SingleCell, Transcriptomics, GeneExpression, BatchEffect

**Collate** 'utils.R' 'HVP.R' 'simulate.R'

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

**git_branch** RELEASE_3_23

**git_last_commit** 3958a85

**git_last_commit_date** 2026-04-28

**Repository** Bioconductor 3.23

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

**Author** Wei Xin Chan [aut, cre] (ORCID: <https://orcid.org/0000-0003-3193-9195>)

**Maintainer** Wei Xin Chan <cweixin@gmail.com>

## Contents

HVP . . . . . 2
hvp-class . . . . . 4
plot,hvp,missing-method . . . . . 4
sigmoid . . . . . 5
simulateMicroarray . . . . . 5
splitCols . . . . . 7

**Index** 8

---

HVP *Hierarchical variance partitioning (HVP)*

---

## Description

'HVP' calculates the proportion of variance associated with batch effects in a data set (the "HVP" value of a data set). To determine whether batch effects are statistically significant in a data set, a permutation test can be performed by setting 'nperm' to a number above 100. 'HVP' is an S4 generic function; methods can be added for new classes. S4 methods for class: array-like objects, 'SummarizedExperiment', 'SingleCellExperiment' and 'Seurat' are provided.

## Usage

```
HVP(x, ...)

## S4 method for signature 'matrix'
HVP(x, batch, cls = NULL, nperm = 0, use.sparse = FALSE, ...)

## S4 method for signature 'Matrix'
HVP(x, batch, cls = NULL, nperm = 0, use.sparse = FALSE, ...)

## S4 method for signature 'data.frame'
HVP(x, ...)

## S4 method for signature 'Seurat'
HVP(x, batchname, classname = NULL, nperm = 0, use.sparse = FALSE, ...)

## S4 method for signature 'SummarizedExperiment'
HVP(
  x,
  batchname,
  classname = NULL,
  assayname = NULL,
  nperm = 0,
  use.sparse = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| x | object to calculate HVP for. |
| ... | additional arguments to pass to S4 methods. |
| batch | vector, indicating the batch information of samples. |
| cls | vector or list of vectors with class information of samples. |

| | |
|---|---|
| nperm | numeric indicating number of permutations to simulate in the Monte Carlo permutation test. We recommend a value no less than 1000. By default, no permutation test is performed. |
| use.sparse | logical indicating whether to use sparse matrices when computing HVP. N.B. Using sparse matrices may lead to slight increase in run time. |
| batchname | character, name of column in metadata indicating batch. |
| classname | character, name of column/s in metadata indicating class. |
| assayname | character, name of assay to use. By default the first assay is used. |

## Details

S4 method for class data frame or matrix takes in array with dimensions (nfeatures, nsamples).

S4 method for 'SummarizedExperiment' is applicable for the 'SingleExperiment' class as well, as it inherits from the 'SummarizedExperiment' class.

## Value

hvp S4 object with the following slots:

**'HVP'** the proportion of variance associated with batch effects.

**'sum.squares'** matrix of sum of squares between batch and total sum of squares for all features.

**'p.value'** p-value of permutation test

**'null.distribution'** numeric, null distribution of HVP values.

Last two components are only present if permuation test is performed.

## Author(s)

Wei Xin Chan

## Examples

```
X <- matrix(rnorm(1000), 50, 20)
batch <- factor(rep(1:2, each = 10))
class <- factor(rep(LETTERS[1:2], 10))

res <- HVP(X, batch, class)
```

## hvp-class *HVP results class*

### Description

An S4 class to store the results from Hierarchical variance partitioning (HVP).

### Slots

`HVP` numeric indicating the proportion of variance associated with batch effects.

`sum.squares` matrix containing sum of squares between batches and total sum of squares for all features.

`p.value` optional numeric of P-value from permutation test.

`null.distribution` optional numeric vector of null distribution of HVP values.

## plot,hvp,missing-method *Plot results of permutation test*

### Description

Plot results of permutation test

### Usage

```
## S4 method for signature 'hvp,missing'
plot(x, y, ...)
```

### Arguments

| | |
|---|---|
| `x` | hvp S4 class containing HVP results after permutation testing. |
| `y` | ignored argument for compatibility with generic plot function. |
| `...` | ignored argument for compatibility with generic plot function. |

### Details

Plots the null distribution of the permutation test.

### Value

ggplot object of null distribution of permutation test.

## sigmoid *Sigmoid function*

### Description

Sigmoid function

### Usage

```
sigmoid(x, r = 1, s = 0)
```

### Arguments

| | |
|---|---|
| x | numeric scalar/vector/matrix |
| r | inverse scale parameter of the sigmoid function |
| s | midpoint parameter of the sigmoid function |

### Value

A numeric scalar/vector/matrix of the same dimensions containing the transformed values.

### Examples

```
p <- sigmoid(0.5)
```

## simulateMicroarray *Simulate log-transformed microarray gene expression data*

### Description

Simulate log-transformed microarray gene expression data

### Usage

```
simulateMicroarray(
  crosstab,
  m,
  delta = 1,
  gamma = 0.5,
  phi = 0.2,
  c = 10,
  d = 6,
  epsilon = 0.5,
  kappa = 0.2,
```

```
  a = 40,
  b = 5,
  dropout = FALSE,
  r = 2,
  s = -6
)
```

## Arguments

| | |
|---|---|
| crosstab | matrix of contingency table specifying number of samples in each class-batch condition, with classes as rows and batches as columns. |
| m | number of genes. |
| delta | magnitude of additive batch effects (i.e. standard deviation of normal distribution modelling batch log fold change means of all genes). |
| gamma | magnitude of multiplicative batch effects (i.e. standard deviation of normal distribution modelling log batch effect terms of all samples in a batch). |
| phi | percentage of differentially expressed genes. |
| c | shape parameter of Gamma distribution modelling class log fold change means of all genes. |
| d | rate parameter of Gamma distribution modelling class log fold change means of all genes. |
| epsilon | magnitude of random noise across samples (i.e. standard deviation of normal distribution modelling log expression values with class effects only). |
| kappa | standard deviation of normal distribution modelling log scaling factors of all samples. |
| a | shape parameter of Gamma distribution modelling basal log mean expression of all genes. |
| b | rate parameter of Gamma distribution modelling basal log mean expression of all genes. |
| dropout | logical indicating whether to perform dropout |
| r | inverse scale parameter of the sigmoid function used to calculate probability of dropout for each value. |
| s | midpoint parameter of the sigmoid function used to calculate probability of dropout for each value. |

## Value

A list containing the following components:

**‘X‘** matrix with dimensions ‘(m, n)‘ of log expression values.

**‘metadata‘** data frame with ‘n‘ rows of sample metadata.

**‘diff.genes‘** character vector, names of differentially expressed genes.

**‘Y‘** matrix with dimensions ‘(m, n)‘ of log expression values with class effects only.

**‘batch.terms‘** matrix with dimensions ‘(m, n)‘ of log batch effect terms.

**'class.logfc'** matrix of class log fold change means for each gene in each class

**'batch.logfc'** matrix of batch log fold change means for each gene in each batch.

**'params'** list of parameters supplied.

## Author(s)

Wei Xin Chan

## Examples

```
crosstab <- matrix(10, 3, 2)
data <- simulateMicroarray(crosstab, 100)
```

---

`splitCols` *Splits subsettable objects according to their columns*

---

## Description

Splits subsettable objects according to their columns

## Usage

```
splitCols(x, f, drop = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `x` | subsettable object to be split |
| `f` | vector or list of vectors indicating the grouping of columns |
| `drop` | logical indicating if levels that do not occur should be dropped |
| `...` | optional arguments to [split()] |

## Value

List of objects split by columns

## Examples

```
X <- matrix(1:60, 10, 6)
cond <- rep(1:3, each = 2)
splitCols(X, cond)
```

# Index

HVP, 2
HVP,data.frame-method *(HVP)*, 2
HVP,Matrix-method *(HVP)*, 2
HVP,matrix-method *(HVP)*, 2
HVP,Seurat-method *(HVP)*, 2
HVP,SummarizedExperiment-method *(HVP)*, 2
hvp-class, 4

plot,hvp,missing-method, 4

sigmoid, 5
simulateMicroarray, 5
splitCols, 7