# Package 'ruv'

October 14, 2022

**Title** Detect and Remove Unwanted Variation using Negative Controls

**Description** Implements the 'RUV' (Remove Unwanted Variation) algorithms. These algorithms attempt to adjust for systematic errors of unknown origin in high-dimensional data. The algorithms were originally developed for use with genomic data, especially microarray data, but may be useful with other types of high-dimensional data as well. These algorithms were proposed in Gagnon-Bartsch and Speed (2012) <doi:10.1093/nar/gkz433>, Gagnon-Bartsch, Jacob and Speed (2013), and Molania, et. al. (2019) <doi:10.1093/nar/gkz433>. The algorithms require the user to specify a set of negative control variables, as described in the references. The algorithms included in this package are 'RUV-2', 'RUV-4', 'RUV-inv', 'RUV-rinv', 'RUV-I', and RUV-III', along with various supporting algorithms.

**Version** 0.9.7.1

**Date** 2019-08-30

**Imports** stats, ggplot2, scales, gridExtra

**Suggests** shiny, colourpicker

**Author** Johann Gagnon-Bartsch <johanngb@umich.edu>

**Maintainer** Johann Gagnon-Bartsch <johanngb@umich.edu>

**License** GPL

**URL** http://www-personal.umich.edu/~johanngb/ruv/

**LazyLoad** yes

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2019-08-30 21:50:02 UTC

## R topics documented:

---

`ruv-package` *Detect and Remove Unwanted Variation using Negative Controls*

---

## Description

Implements the 'RUV' (Remove Unwanted Variation) algorithms. These algorithms attempt to adjust for systematic errors of unknown origin in high-dimensional data. The algorithms were originally developed for use with genomic data, especially microarray data, but may be useful with other types of high-dimensional data as well. These algorithms were proposed in Gagnon-Bartsch and Speed (2012) <doi:10.1093/nar/gkz433>, Gagnon-Bartsch, Jacob and Speed (2013), and Molania, et. al. (2019) <doi:10.1093/nar/gkz433>. The algorithms require the user to specify a set of negative control variables, as described in the references. The algorithms included in this package are 'RUV-2', 'RUV-4', 'RUV-inv', 'RUV-rinv', 'RUV-I', and RUV-III', along with various supporting algorithms.

## Details

| | |
|---|---|
| Package: | ruv |
| Type: | Package |
| Version: | 0.9.7.1 |
| Date: | 2019-08-30 |
| License: | GPL |
| LazyLoad: | yes |
| URL: | http://www-personal.umich.edu/~johanngb/ruv/ |

## Note

Additional resources can be found at http://www-personal.umich.edu/~johanngb/ruv/.

## Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

## References

Gagnon-Bartsch, J.A. and T.P. Speed (2012). Using control genes to correct for unwanted variation in microarray data. Biostatistics. <doi:10.1093/biostatistics/kxr034>

Gagnon-Bartsch, J.A., L. Jacob, and T.P. Speed (2013). Removing Unwanted Variation from High Dimensional Data with Negative Controls. Technical report. Available at: http://statistics.berkeley.edu/tech-reports/820

Molania, R., J. A. Gagnon-Bartsch, A. Dobrovic, and T. P. Speed (2019). A new normalization for the Nanostring nCounter gene expression assay. Nucleic Acids Research. <doi:10.1093/nar/gkz433>

## See Also

RUV2, RUV4, RUVinv, RUVrinv, variance_adjust, RUVI, RUVIII

---

`collapse.replicates` *Collapse Replicates*

---

## Description

This function is intended for use in conjunction with RUVIII, specifically when using the `average=TRUE` option. When using the `average=TRUE` option, the adjusted data matrix has only one row for each set of replicates. In other words, each set of replicate rows in the orginal data matrix is "collapsed" into a single row in the adjusted data matrix. This function similarly collapses the rows of a dataframe of covariates. Only covariates that are constant within each set of replicates are retained.

## Usage

```
collapse.replicates(df, M)
```

### Arguments

| | |
|---|---|
| df | A dataframe. |
| M | The replicate structure. See RUVIII for details. |

### Value

A sub-dataframe of df.

### Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

### See Also

RUVIII

---

## design.matrix *Design Matrix*

---

### Description

Creates a design matrix.

### Usage

```
design.matrix(a, name = "X", remove.collinear = TRUE, include.intercept = TRUE)
```

### Arguments

| | |
|---|---|
| a | Object from which to create a design matrix. Can be a vector, matrix, factor, or dataframe. |
| name | Name of the design matrix. Used to name columns that aren't already named (e.g. X1, X2, etc.) |
| remove.collinear | Will remove columns that are collinear, to ensure the design matrix is full rank. |
| include.intercept | Add an intercept to the matrix if one is not included already. |

### Details

Numerical vectors are not modified. Factors are converted to dummy variables. Character vectors are converted to factors, and then to dummy variables.

### Value

A matrix.

### Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

---

getK *Get K*

---

### Description

Finds an often-suitable value of K for use in RUV-4.

### Usage

```
getK(Y, X, ctl, Z = 1, eta = NULL, include.intercept = TRUE,
     fullW0 = NULL, cutoff = NULL, method="select", l=1, inputcheck = TRUE)
```

### Arguments

| | |
|---|---|
| Y | The data. A m by n matrix, where m is the number of samples and n is the number of features. |
| X | The factor(s) of interest. A m by p matrix, where m is the number of samples and p is the number of factors of interest. Note that X should be only a single column, i.e. p = 1; if X has more than one column, only column l will be used (see below). |
| ctl | An index vector to specify the negative controls. Either a logical vector of length n or a vector of integers. |
| Z | Any additional covariates to include in the model, typically a m by q matrix. Factors and dataframes are also permissible, and converted to a matrix by design.matrix. Alternatively, may simply be 1 (the default) for an intercept term. May also be NULL. |
| eta | Gene-wise (as opposed to sample-wise) covariates. These covariates are adjusted for by RUV-1 before any further analysis proceeds. Can be either (1) a matrix with n columns, (2) a matrix with n rows, (3) a dataframe with n rows, (4) a vector or factor of length n, or (5) simply 1, for an intercept term. |
| include.intercept | Applies to both Z and eta. When Z or eta (or both) is specified (not NULL) but does not already include an intercept term, this will automatically include one. If only one of Z or eta should include an intercept, this variable should be set to FALSE, and the intercept term should be included manually where desired. |
| fullW0 | Can be included to speed up execution. Is returned by previous calls of getK, RUV4, RUVinv, or RUVrinv (see below). |
| cutoff | Specify an alternative cut-off. Default is the (approximate) 90th percentile of the distribution of the first singular value of an m by n gaussian matrix. |

| | |
|---|---|
| method | Can be set to either leave1out, fast, or select. leave1out is the preferred method but may be slow, fast is an approximate method that is faster but may provide poor results if n_c is not much larger than m, and select (the default) tries to choose for you. |
| l | Which column of X to use in the getK algorithm. |
| inputcheck | Perform a basic sanity check on the inputs, and issue a warning if there is a problem. |

## Value

A list containing

| | |
|---|---|
| k | the estimated value of k |
| cutoff | The cutoff value used |
| sizeratios | A measure of the relative sizes of the rows of alpha. |
| fullW0 | Can be used to speed up future calls of RUV4. |

## Warning

This value of K will not be the best choice in all cases. Moreover, it will often be a poor choice of K for use with RUV2. See Gagnon-Bartsch and Speed (2012) for commentary on estimating k.

## Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

## References

Using control genes to correct for unwanted variation in microarray data. Gagnon-Bartsch and Speed, 2012. Available at: http://biostatistics.oxfordjournals.org/content/13/3/539.full.

Removing Unwanted Variation from High Dimensional Data with Negative Controls. Gagnon-Bartsch, Jacob, and Speed, 2013. Available at: http://statistics.berkeley.edu/tech-reports/820.

## See Also

RUV4

## Examples

```
## Create some simulated data
m = 50
n = 10000
nc = 1000
p = 1
k = 20
ctl = rep(FALSE, n)
ctl[1:nc] = TRUE
X = matrix(c(rep(0,floor(m/2)), rep(1,ceiling(m/2))), m, p)
beta = matrix(rnorm(p*n), p, n)
```

```
beta[,ctl] = 0
W = matrix(rnorm(m*k),m,k)
alpha = matrix(rnorm(k*n),k,n)
epsilon = matrix(rnorm(m*n),m,n)
Y = X%*%beta + W%*%alpha + epsilon

## Run getK
temp = getK(Y, X, ctl)
K = temp$k
```

---

`get_empirical_variances` *Get empirical variances*

---

## Description

This method implements the method of empirical variances as described in Gagnon-Bartsch, Jacob, and Speed (2013). This function is normally called from the function `variance_adjust`, and is not normally intended for stand-alone use.

## Usage

```
get_empirical_variances(sigma2, betahat, bin = 10,
                        rescaleconst = NULL)
```

## Arguments

| | |
|---|---|
| `sigma2` | Estimates of sigma^2 |
| `betahat` | Estimates of beta |
| `bin` | The bin size |
| `rescaleconst` | The expected value of the average of the smallest `bin - 1` of `bin` independent chi-square random variables. This can be specified to save computational time (otherwise, it is calculated by simulation). |

## Value

A vector of the empirical variances.

## Author(s)

Johann Gagnon-Bartsch

## References

Removing Unwanted Variation from High Dimensional Data with Negative Controls. Gagnon-Bartsch, Jacob, and Speed, 2013. Available at: http://statistics.berkeley.edu/tech-reports/820.

## See Also

variance_adjust

---

`google_search` *Google Search URL*

---

## Description

Converts a string to URL for a goolge search of that string.

## Usage

```
google_search(a)
```

## Arguments

| | |
|---|---|
| a | A string. |

## Value

A string that is a URL.

---

`inputcheck1` *Input Check One*

---

## Description

Performs a basic sanity check on the arguments passed to RUV-2, RUV-4, RUV-inv, and RUV-rinv.

## Usage

```
inputcheck1(Y, X, Z, ctl, check.na=TRUE)
```

## Arguments

| | |
|---|---|
| Y | The data. A m by n matrix, where m is the number of samples and n is the number of features. |
| X | The factor(s) of interest. A m by p matrix, where m is the number of samples and p is the number of factors of interest. Very often p = 1. |
| Z | Any additional covariates to include in the model. Either a m by q matrix of covariates, or simply 1 (the default) for an intercept term. |
| ctl | The negative controls. A logical vector of length n. |
| check.na | Whether to check for missing values. |

## Value

Returns NULL. The function is only called to check for problems in the arguments and to issue warnings if any problems are found.

## See Also

RUV2, RUV4, RUVinv, RUVrinv

---

`invvar` *Inverse Method Variances*

---

## Description

Estimate the features' variances using the inverse method. This function is usually called from RUVinv and not normally intended for stand-alone use.

## Usage

```
invvar(Y, ctl, XZ = NULL, eta = NULL, lambda = NULL,
       invsvd = NULL)
```

## Arguments

| | |
|---|---|
| `Y` | The data. A m by n matrix, where m is the number of samples and n is the number of features. |
| `ctl` | The negative controls. A logical vector of length n. |
| `XZ` | A m by (p + q) matrix containing both the factor(s) of interest (X) and known covariates (Z). |
| `eta` | Gene-wise (as opposed to sample-wise) covariates. These covariates are adjusted for by RUV-1 before any further analysis proceeds. A matrix with n columns. |
| `lambda` | Ridge parameter. If specified, the ridged inverse method will be used. |
| `invsvd` | Can be included to speed up execution. Generally used when calling invvar many times with different values of lambda. |

## Value

A list containing

| | |
|---|---|
| `sigma2` | Estimates of the features' variances. A vector of length n. |
| `df` | The "effective degrees of freedom" |
| `invsvd` | Can be used to speed up future calls of invvar. |

## Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

## References

Removing Unwanted Variation from High Dimensional Data with Negative Controls. Gagnon-Bartsch, Jacob, and Speed, 2013. Available at: http://statistics.berkeley.edu/tech-reports/820.

## See Also

RUVinv, RUVrinv

---

`projectionplotvariables` *Projection Plot Variables*

---

## Description

Calculates the variables necessary to produce a projection plot.

## Usage

```
projectionplotvariables(Y, X, W)
```

## Arguments

| | |
|---|---|
| `Y` | The data. A m by n matrix, where m is the number of samples and n is the number of features. |
| `X` | A m by p matrix containing the factor(s) of interest. |
| `W` | A m by k matrix containing the estimated unwanted factors. |

## Details

Typically intended for internal use, and called after adjustment for known covariates (Z).

## Value

A list containing

| | |
|---|---|
| `byx` | Regression coefficients from regressing Y on X. |
| `bwx` | Regression coefficients from regressing W on X. |
| `projectionplotalpha` | A reparameterization of alpha. |
| `projectionplotW` | A reparameterization of W. |

randinvvar *(Randomization) Inverse Method Variances*

## Description

Estimate the features' variances using a stochastic version of the inverse method. This function is usually called from RUVinv and not normally intended for stand-alone use.

## Usage

```
randinvvar(Y, ctl, XZ = NULL, eta = NULL, lambda = NULL,
           iterN = 1e+05)
```

## Arguments

| | |
|---|---|
| Y | The data. A m by n matrix, where m is the number of samples and n is the number of features. |
| ctl | The negative controls. A logical vector of length n. |
| XZ | A m by (p + q) matrix containing both the factor(s) of interest (X) and known covariates (Z). |
| eta | Gene-wise (as opposed to sample-wise) covariates. These covariates are adjusted for by RUV-1 before any further analysis proceeds. A matrix with n columns. |
| lambda | Ridge parameter. If specified, the ridged inverse method will be used. |
| iterN | The number of random "factors of interest" to generate. |

## Value

A list containing

| | |
|---|---|
| sigma2 | Estimates of the features' variances. A vector of length n. |
| df | The "effective degrees of freedom" |

## Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

## References

Removing Unwanted Variation from High Dimensional Data with Negative Controls. Gagnon-Bartsch, Jacob, and Speed, 2013. Available at: http://statistics.berkeley.edu/tech-reports/820.

## See Also

RUVinv, RUVrinv, invvar

`replicate.matrix` *Replicate (Mapping) Matrix*

## Description

For use with RUVIII, generates a mapping matrix that describes the replicate structure.

## Usage

```
replicate.matrix(a, burst=NULL, return.factor=FALSE, name="M", sep="_", burstsep = "_")
```

## Arguments

| | |
|---|---|
| a | An object that describes the replicate structure. Can be a vector, matrix, factor, or dataframe. If a vector, it is converted to a factor. If a factor, each level of the factor is taken to represent a set of replicates. If a matrix: First it is determined whether it is already a mapping matrix; if so, the matrix is returned unchanged; if not, the matrix is converted to a dataframe. If a dataframe: Each column is converted to a factor. A new factor is then created with levels for every possible combination of factor levels in the dataframe. For example, if the dataframe contains two factors, patientID and sampleDate, the new factor will have a unique level for each (observed) combination of patientID and sampleDate. Thus observations will be considered replicates if they have identical values for BOTH patientID and sampleDate. |
| burst | A character vector, containing the names of factor levels to be "burst." When a factor level is burst, the corresponding observations are no longer replicates; they become singletons. |
| return.factor | Return a factor instead of the mapping matrix. This may be useful in two situations: (1) When the input is a mapping matrix, and it is desired to convert it back to a factor; (2) When making repeated calls to `replicate.matrix` in order to define the replicates in several steps. Example of (2): Suppose there are 4 patients and 3 sample dates. We wish to designate as replicates observations that have the same patient ID and sample date, but only for the first two sample dates; none of the observations in the third sample date should be considered replicates. We would first call `replicate.matrix` using only the sampleDate factor, bursting the third level, and returning another factor. We would then call `replicate.matrix` again, this time with a dataframe containing patientID and the bursted sampleDate. See below for example code. |
| name | Name of the mapping matrix. Used to name columns that aren't already named (e.g. M1, M2, etc.) |
| sep | Text separating the level names of different factors when they are combined. |
| burstsep | Text appended to factor level names when bursting a factor. This text is then followed by a number. Example: if the factor level to be burst is "June29", and burstsep is the default value of "_", then the new levels will be "June29_1", "June29_2", etc. |

## Value

A matrix or a factor, depending on the value of `return.factor`.

## Warning

Be sure to change the default values of `sep` and `burstsep` if there is any risk of factor level naming conflicts (e.g. if existing factors already have level names like "patient_1", "patient_2", etc.

## Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

## See Also

RUVIII

## Examples

```
# Define patientID and sampleDate
patientID = paste("patient", rep(1:4, each=6), sep="")
#print(patientID)
sampleDate = paste("June", rep(c(12,17,29), 8), sep="")
#print(sampleDate)
# Create a mapping matrix, where every unique
# patientID / sampleDate combination define a set of replicates
M = replicate.matrix(data.frame(patientID, sampleDate))
#print(M)
# Convert M back to a factor
M = replicate.matrix(M, return.factor=TRUE)
#print(M)
# Create a factor for sampleDate, but burst the third date
temp = replicate.matrix(sampleDate, burst="June29", return.factor=TRUE)
#print(temp)
# Create a mapping matrix as described above in the description of return.factor
M = replicate.matrix(data.frame(temp, patientID))
#print(M)
```

---

# residop *Residual Operator*

---

## Description

Applies the residual operator of a matrix B to a matrix A.

## Usage

```
residop(A, B)
```

### Arguments

| | |
|---|---|
| A | A matrix with m rows. |
| B | Another matrix with m rows. |

### Details

The columns of B must be linearly independent.

### Value

The matrix A after projection into the orthogonal complement of the column space of B.

---

## RUV2 *Remove Unwanted Variation, 2-step*

---

### Description

The RUV-2 algorithm. Estimates and adjusts for unwanted variation using negative controls.

### Usage

```
RUV2(Y, X, ctl, k, Z=1, eta=NULL, include.intercept=TRUE,
fullW=NULL, svdyc=NULL, do_projectionplot=TRUE, inputcheck=TRUE)
```

### Arguments

| | |
|---|---|
| Y | The data. A m by n matrix, where m is the number of samples and n is the number of features. |
| X | The factor(s) of interest. A m by p matrix, where m is the number of samples and p is the number of factors of interest. Very often p = 1. Factors and dataframes are also permissible, and converted to a matrix by design.matrix. |
| ctl | An index vector to specify the negative controls. Either a logical vector of length n or a vector of integers. |
| k | The number of unwanted factors to use. Can be 0. |
| Z | Any additional covariates to include in the model, typically a m by q matrix. Factors and dataframes are also permissible, and converted to a matrix by design.matrix. Alternatively, may simply be 1 (the default) for an intercept term. May also be NULL. |
| eta | Gene-wise (as opposed to sample-wise) covariates. These covariates are adjusted for by RUV-1 before any further analysis proceeds. Can be either (1) a matrix with n columns, (2) a matrix with n rows, (3) a dataframe with n rows, (4) a vector or factor of length n, or (5) simply 1, for an intercept term. |

include.intercept
: Applies to both Z and eta. When Z or eta (or both) is specified (not NULL) but does not already include an intercept term, this will automatically include one. If only one of Z or eta should include an intercept, this variable should be set to FALSE, and the intercept term should be included manually where desired.

fullW
: Can be included to speed up execution. Is returned by previous calls of RUV2 (see below).

svdyc
: Can be included to speed up execution. For internal use; please use fullW instead.

do_projectionplot
: Calculate the quantities necessary to generate a projection plot.

inputcheck
: Perform a basic sanity check on the inputs, and issue a warning if there is a problem.

## Details

Implements the RUV-2 algorithm as described in Gagnon-Bartsch and Speed (2012), using the SVD as the factor analysis routine. Unwanted factors W are estimated using control genes. Y is then regressed on the variables X, Z, and W.

## Value

A list containing

betahat
: The estimated coefficients of the factor(s) of interest. A p by n matrix.

sigma2
: Estimates of the features' variances. A vector of length n.

t
: t statistics for the factor(s) of interest. A p by n matrix.

p
: P-values for the factor(s) of interest. A p by n matrix.

Fstats
: F statistics for testing all of the factors in X simultaneously.

Fpvals
: P-values for testing all of the factors in X simultaneously.

multiplier
: The constant by which sigma2 must be multiplied in order get an estimate of the variance of betahat

df
: The number of residual degrees of freedom.

W
: The estimated unwanted factors.

alpha
: The estimated coefficients of W.

byx
: The coefficients in a regression of Y on X (after both Y and X have been "adjusted" for Z). Useful for projection plots.

bwx
: The coefficients in a regression of W on X (after X has been "adjusted" for Z). Useful for projection plots.

X
: X. Included for reference.

k
: k. Included for reference.

ctl
: ctl. Included for reference.

Z
: Z. Included for reference.

eta
: eta. Included for reference.

`fullW` Can be used to speed up future calls of RUV2.

`projectionplotW`
A reparameterization of W useful for projection plots.

`projectionplotalpha`
A reparameterization of alpha useful for projection plots.

`include.intercept`
`include.intercept`. Included for reference.

`method` Character variable with value "RUV2". Included for reference.

## Note

Additional resources can be found at http://www-personal.umich.edu/~johanngb/ruv/.

## Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

## References

Using control genes to correct for unwanted variation in microarray data. Gagnon-Bartsch and Speed, 2012. Available at: http://biostatistics.oxfordjournals.org/content/13/3/539.full.

Removing Unwanted Variation from High Dimensional Data with Negative Controls. Gagnon-Bartsch, Jacob, and Speed, 2013. Available at: http://statistics.berkeley.edu/tech-reports/820.

## See Also

RUV4, RUVinv, RUVrinv, variance_adjust

## Examples

```
## Create some simulated data
m = 50
n = 10000
nc = 1000
p = 1
k = 20
ctl = rep(FALSE, n)
ctl[1:nc] = TRUE
X = matrix(c(rep(0,floor(m/2)), rep(1,ceiling(m/2))), m, p)
beta = matrix(rnorm(p*n), p, n)
beta[,ctl] = 0
W = matrix(rnorm(m*k),m,k)
alpha = matrix(rnorm(k*n),k,n)
epsilon = matrix(rnorm(m*n),m,n)
Y = X%*%beta + W%*%alpha + epsilon

## Run RUV-2
fit = RUV2(Y, X, ctl, k)

## Get adjusted variances and p-values
fit = variance_adjust(fit)
```

RUV4 *Remove Unwanted Variation, 4-step*

## Description

The RUV-4 algorithm. Estimates and adjusts for unwanted variation using negative controls.

## Usage

```
RUV4(Y, X, ctl, k, Z = 1, eta = NULL, include.intercept=TRUE,
fullW0=NULL, inputcheck=TRUE)
```

## Arguments

| | |
|---|---|
| Y | The data. A m by n matrix, where m is the number of samples and n is the number of features. |
| X | The factor(s) of interest. A m by p matrix, where m is the number of samples and p is the number of factors of interest. Very often p = 1. Factors and dataframes are also permissible, and converted to a matrix by design.matrix. |
| ctl | An index vector to specify the negative controls. Either a logical vector of length n or a vector of integers. |
| k | The number of unwanted factors to use. Can be 0. |
| Z | Any additional covariates to include in the model, typically a m by q matrix. Factors and dataframes are also permissible, and converted to a matrix by design.matrix. Alternatively, may simply be 1 (the default) for an intercept term. May also be NULL. |
| eta | Gene-wise (as opposed to sample-wise) covariates. These covariates are adjusted for by RUV-1 before any further analysis proceeds. Can be either (1) a matrix with n columns, (2) a matrix with n rows, (3) a dataframe with n rows, (4) a vector or factor of length n, or (5) simply 1, for an intercept term. |
| include.intercept | Applies to both Z and eta. When Z or eta (or both) is specified (not NULL) but does not already include an intercept term, this will automatically include one. If only one of Z or eta should include an intercept, this variable should be set to FALSE, and the intercept term should be included manually where desired. |
| fullW0 | Can be included to speed up execution. Is returned by previous calls of RUV4, RUVinv, or RUVrinv (see below). |
| inputcheck | Perform a basic sanity check on the inputs, and issue a warning if there is a problem. |

## Details

Implements the RUV-4 algorithm as described in Gagnon-Bartsch, Jacob, and Speed (2013), using the SVD as the factor analysis routine. Unwanted factors W are estimated using control genes. Y is then regressed on the variables X, Z, and W.

## Value

A list containing

| | |
|---|---|
| betahat | The estimated coefficients of the factor(s) of interest. A p by n matrix. |
| sigma2 | Estimates of the features' variances. A vector of length n. |
| t | t statistics for the factor(s) of interest. A p by n matrix. |
| p | P-values for the factor(s) of interest. A p by n matrix. |
| Fstats | F statistics for testing all of the factors in X simultaneously. |
| Fpvals | P-values for testing all of the factors in X simultaneously. |
| multiplier | The constant by which sigma2 must be multiplied in order get an estimate of the variance of betahat |
| df | The number of residual degrees of freedom. |
| W | The estimated unwanted factors. |
| alpha | The estimated coefficients of W. |
| byx | The coefficients in a regression of Y on X (after both Y and X have been "adjusted" for Z). Useful for projection plots. |
| bwx | The coefficients in a regression of W on X (after X has been "adjusted" for Z). Useful for projection plots. |
| X | X. Included for reference. |
| k | k. Included for reference. |
| ctl | ctl. Included for reference. |
| Z | Z. Included for reference. |
| eta | eta. Included for reference. |
| fullW0 | Can be used to speed up future calls of RUV4. |
| include.intercept | include.intercept. Included for reference. |
| method | Character variable with value "RUV4". Included for reference. |

## Note

Additional resources can be found at http://www-personal.umich.edu/~johanngb/ruv/.

## Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

## References

Using control genes to correct for unwanted variation in microarray data. Gagnon-Bartsch and Speed, 2012. Available at: http://biostatistics.oxfordjournals.org/content/13/3/539.full.

Removing Unwanted Variation from High Dimensional Data with Negative Controls. Gagnon-Bartsch, Jacob, and Speed, 2013. Available at: http://statistics.berkeley.edu/tech-reports/820.

## See Also

RUV2, RUVinv, RUVrinv, variance_adjust

## Examples

```
## Create some simulated data
m = 50
n = 10000
nc = 1000
p = 1
k = 20
ctl = rep(FALSE, n)
ctl[1:nc] = TRUE
X = matrix(c(rep(0,floor(m/2)), rep(1,ceiling(m/2))), m, p)
beta = matrix(rnorm(p*n), p, n)
beta[,ctl] = 0
W = matrix(rnorm(m*k),m,k)
alpha = matrix(rnorm(k*n),k,n)
epsilon = matrix(rnorm(m*n),m,n)
Y = X%*%beta + W%*%alpha + epsilon

## Run RUV-4
fit = RUV4(Y, X, ctl, k)

## Get adjusted variances and p-values
fit = variance_adjust(fit)
```

---

# RUVI *RUV-I*

---

## Description

The RUV-I algorithm. Generally used as a preprocessing step to RUV-2, RUV-4, RUV-inv, RUV-rinv, or RUVIII. RUV1 is an alias of (identical to) RUVI.

## Usage

```
RUVI(Y, eta, ctl, include.intercept = TRUE)

RUV1(Y, eta, ctl, include.intercept = TRUE)
```

## Arguments

| | |
|---|---|
| Y | The data. A m by n matrix, where m is the number of samples and n is the number of features. |
| eta | Gene-wise (as opposed to sample-wise) covariates. A matrix with n columns. |
| ctl | The negative controls. A logical vector of length n. |
| include.intercept | Add an intercept term to eta if it does not include one already. |

## Details

Implements the RUV-I algorithm as described in Gagnon-Bartsch, Jacob, and Speed (2013). Most often this algorithm is not used directly, but rather is called from RUV-2, RUV-4, RUV-inv, or RUV-rinv. Note that RUV1 and RUVI are two different names for the same (identical) function.

## Value

An adjusted data matrix (i.e., an adjusted Y)

## Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

## References

Using control genes to correct for unwanted variation in microarray data. Gagnon-Bartsch and Speed, 2012. Available at: http://biostatistics.oxfordjournals.org/content/13/3/539.full.

Removing Unwanted Variation from High Dimensional Data with Negative Controls. Gagnon-Bartsch, Jacob, and Speed, 2013. Available at: http://statistics.berkeley.edu/tech-reports/820.

## See Also

RUV2, RUV4, RUVinv, RUVrinv, RUVIII

---

# RUVIII *RUV-III*

---

## Description

Globally adjust data matrix using both negative controls and replicates.

## Usage

```
RUVIII(Y, M, ctl, k = NULL, eta = NULL, include.intercept = TRUE,
       average = FALSE, fullalpha = NULL, return.info = FALSE, inputcheck = TRUE)
```

## Arguments

| | |
|---|---|
| Y | The data. A m by n matrix, where m is the number of observations and n is the number of features. |
| M | The replicate structure. Represented internally as a mapping matrix. The mapping matrix has m rows (one for each observation), and each column represents a set of replicates. The (i, j)-th entry of the mapping matrix is 1 if the i-th observation is in replicate set j, and 0 otherwise. Each observation must be in exactly one set of replicates (some replicate sets may contain only one observation), and thus each row of M must sum to 1. M may be the mapping matrix itself. Alternatively, M may be a vector, factor, or dataframe, in which case it is converted to the mapping matrix by the replicate.matrix function. |

| | |
|---|---|
| ctl | An index vector to specify the negative controls. Either a logical vector of length n or a vector of integers. |
| k | The number of unwanted factors to use. Can be 0, in which case no adjustment is made. Can also be NULL (the default value), in which case the maximum possible value of k is used; note that in this case no singular value decomposition is necessary and execution is faster. |
| eta | Gene-wise (as opposed to sample-wise) covariates. These covariates are adjusted for by RUV-1 before any further analysis proceeds. Can be either (1) a matrix with n columns, (2) a matrix with n rows, (3) a dataframe with n rows, (4) a vector or factor of length n, or (5) simply 1, for an intercept term. |
| include.intercept | When eta is specified (not NULL) but does not already include an intercept term, this will automatically include one. |
| average | Average replicates after adjustment. |
| fullalpha | Can be included to speed up execution. Is returned by previous calls of RUVIII (see below). |
| return.info | If FALSE, only the adjusted data matrix is returned. If TRUE, additional information is returned (see below). |
| inputcheck | Perform a basic sanity check on the inputs, and issue a warning if there is a problem. |

## Value

If codereturn.info is TRUE, a list is returned that contains:

| | |
|---|---|
| newY | The adjusted data matrix. |
| M | The replicate mapping matrix. Included for reference. |
| fullalpha | Can be used to speed up future calls to RUVIII |

Otherwise, if return.info is FALSE, only the adjusted data matrix is returned.

## Note

Additional resources can be found at http://www-personal.umich.edu/~johanngb/ruv/.

## Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

RUVinv *Remove Unwanted Variation, inverse method*

## Description

The RUV-inv algorithm. Estimates and adjusts for unwanted variation using negative controls.

## Usage

```
RUVinv(Y, X, ctl, Z=1, eta=NULL, include.intercept=TRUE,
       fullW0=NULL, invsvd=NULL, lambda=NULL,
       randomization=FALSE, iterN=100000, inputcheck=TRUE)
```

## Arguments

| | |
|---|---|
| Y | The data. A m by n matrix, where m is the number of samples and n is the number of features. |
| X | The factor(s) of interest. A m by p matrix, where m is the number of samples and p is the number of factors of interest. Very often p = 1. Factors and dataframes are also permissible, and converted to a matrix by design.matrix. |
| ctl | An index vector to specify the negative controls. Either a logical vector of length n or a vector of integers. |
| Z | Any additional covariates to include in the model, typically a m by q matrix. Factors and dataframes are also permissible, and converted to a matrix by design.matrix. Alternatively, may simply be 1 (the default) for an intercept term. May also be NULL. |
| eta | Gene-wise (as opposed to sample-wise) covariates. These covariates are adjusted for by RUV-1 before any further analysis proceeds. Can be either (1) a matrix with n columns, (2) a matrix with n rows, (3) a dataframe with n rows, (4) a vector or factor of length n, or (5) simply 1, for an intercept term. |
| include.intercept | Applies to both Z and eta. When Z or eta (or both) is specified (not NULL) but does not already include an intercept term, this will automatically include one. If only one of Z or eta should include an intercept, this variable should be set to FALSE, and the intercept term should be included manually where desired. |
| fullW0 | Can be included to speed up execution. Is returned by previous calls of RUV4, RUVinv, or RUVrinv (see below). |
| invsvd | Can be included to speed up execution. Generally used when calling RUV(r)inv many times with different values of lambda. Is returned by previous calls of RUV(r)inv (see below). |
| lambda | Ridge parameter. If specified, the ridged inverse method will be used. |
| randomization | Whether the inverse-method variances should be computed using randomly generated factors of interest (as opposed to a numerical integral). |

| | |
|---|---|
| iterN | The number of random "factors of interest" to generate (used only when randomization=TRUE). |
| inputcheck | Perform a basic sanity check on the inputs, and issue a warning if there is a problem. |

## Details

Implements the RUV-inv algorithm as described in Gagnon-Bartsch, Jacob, and Speed (2013).

## Value

A list containing

| | |
|---|---|
| betahat | The estimated coefficients of the factor(s) of interest. A p by n matrix. |
| sigma2 | Estimates of the features' variances. A vector of length n. |
| t | t statistics for the factor(s) of interest. A p by n matrix. |
| p | P-values for the factor(s) of interest. A p by n matrix. |
| Fstats | F statistics for testing all of the factors in X simultaneously. |
| Fpvals | P-values for testing all of the factors in X simultaneously. |
| multiplier | The constant by which sigma2 must be multiplied in order get an estimate of the variance of betahat |
| df | The number of residual degrees of freedom. |
| W | The estimated unwanted factors. |
| alpha | The estimated coefficients of W. |
| byx | The coefficients in a regression of Y on X (after both Y and X have been "adjusted" for Z). Useful for projection plots. |
| bwx | The coefficients in a regression of W on X (after X has been "adjusted" for Z). Useful for projection plots. |
| X | X. Included for reference. |
| k | k. Included for reference. |
| ctl | ctl. Included for reference. |
| Z | Z. Included for reference. |
| eta | eta. Included for reference. |
| fullW0 | Can be used to speed up future calls of RUV4. |
| lambda | lambda. Included for reference. |
| invsvd | Can be used to speed up future calls of RUV(r)inv. |
| include.intercept | include.intercept. Included for reference. |
| method | Character variable with value "RUVinv". Included for reference. |

## Note

Additional resources can be found at http://www-personal.umich.edu/~johanngb/ruv/.

## Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

## References

Using control genes to correct for unwanted variation in microarray data. Gagnon-Bartsch and Speed, 2012. Available at: http://biostatistics.oxfordjournals.org/content/13/3/539.full.

Removing Unwanted Variation from High Dimensional Data with Negative Controls. Gagnon-Bartsch, Jacob, and Speed, 2013. Available at: http://statistics.berkeley.edu/tech-reports/820.

## See Also

RUV2, RUV4, RUVrinv, variance_adjust, invvar

## Examples

```
## Create some simulated data
m = 50
n = 10000
nc = 1000
p = 1
k = 20
ctl = rep(FALSE, n)
ctl[1:nc] = TRUE
X = matrix(c(rep(0,floor(m/2)), rep(1,ceiling(m/2))), m, p)
beta = matrix(rnorm(p*n), p, n)
beta[,ctl] = 0
W = matrix(rnorm(m*k),m,k)
alpha = matrix(rnorm(k*n),k,n)
epsilon = matrix(rnorm(m*n),m,n)
Y = X%*%beta + W%*%alpha + epsilon

## Run RUV-inv
fit = RUVinv(Y, X, ctl)

## Get adjusted variances and p-values
fit = variance_adjust(fit)
```

---

RUVrinv *Remove Unwanted Variation, ridged inverse method*

---

## Description

The RUV-rinv algorithm. Estimates and adjusts for unwanted variation using negative controls.

## Usage

```
RUVrinv(Y, X, ctl, Z=1, eta=NULL, include.intercept=TRUE,
        fullW0=NULL, invsvd=NULL, lambda=NULL, k=NULL, l=NULL,
        randomization=FALSE, iterN=100000, inputcheck=TRUE)
```

## Arguments

| | |
|---|---|
| Y | The data. A m by n matrix, where m is the number of samples and n is the number of features. |
| X | The factor(s) of interest. A m by p matrix, where m is the number of samples and p is the number of factors of interest. Very often p = 1. Factors and dataframes are also permissible, and converted to a matrix by design.matrix. |
| ctl | An index vector to specify the negative controls. Either a logical vector of length n or a vector of integers. |
| Z | Any additional covariates to include in the model, typically a m by q matrix. Factors and dataframes are also permissible, and converted to a matrix by design.matrix. Alternatively, may simply be 1 (the default) for an intercept term. May also be NULL. |
| eta | Gene-wise (as opposed to sample-wise) covariates. These covariates are adjusted for by RUV-1 before any further analysis proceeds. Can be either (1) a matrix with n columns, (2) a matrix with n rows, (3) a dataframe with n rows, (4) a vector or factor of length n, or (5) simply 1, for an intercept term. |
| include.intercept | Applies to both Z and eta. When Z or eta (or both) is specified (not NULL) but does not already include an intercept term, this will automatically include one. If only one of Z or eta should include an intercept, this variable should be set to FALSE, and the intercept term should be included manually where desired. |
| fullW0 | Can be included to speed up execution. Is returned by previous calls of RUV4, RUVinv, or RUVrinv (see below). |
| invsvd | Can be included to speed up execution. Generally used when calling RUV(r)inv many times with different values of lambda. Is returned by previous calls of RUV(r)inv (see below). |
| lambda | Ridge parameter. If unspecified, an appropriate default will be used. |
| k | When calculating the default value of lambda, a call to RUV4 is made. This parameter specifies the value of k to use. Otherwise, an appropriate default k will be used. |
| l | If lambda and k are both NULL, then k must be estimated using the getK routine. The getK routine only accepts a single-column X. If p > 1, l specifies which column of X should be used in the getK routine. |
| randomization | Whether the inverse-method variances should be computed using randomly generated factors of interest (as opposed to a numerical integral). |
| iterN | The number of random "factors of interest" to generate (used only when randomization=TRUE). |
| inputcheck | Perform a basic sanity check on the inputs, and issue a warning if there is a problem. |

## Details

Implements the RUV-rinv algorithm as described in Gagnon-Bartsch, Jacob, and Speed (2013). This function is essentially just a wrapper to RUVinv, but with a little extra code to calculate the default value of lambda.

## Value

A list containing

| | |
|---|---|
| betahat | The estimated coefficients of the factor(s) of interest. A p by n matrix. |
| sigma2 | Estimates of the features' variances. A vector of length n. |
| t | t statistics for the factor(s) of interest. A p by n matrix. |
| p | P-values for the factor(s) of interest. A p by n matrix. |
| Fstats | F statistics for testing all of the factors in X simultaneously. |
| Fpvals | P-values for testing all of the factors in X simultaneously. |
| multiplier | The constant by which sigma2 must be multiplied in order get an estimate of the variance of betahat |
| df | The number of residual degrees of freedom. |
| W | The estimated unwanted factors. |
| alpha | The estimated coefficients of W. |
| byx | The coefficients in a regression of Y on X (after both Y and X have been "adjusted" for Z). Useful for projection plots. |
| bwx | The coefficients in a regression of W on X (after X has been "adjusted" for Z). Useful for projection plots. |
| X | X. Included for reference. |
| k | k. Included for reference. |
| ctl | ctl. Included for reference. |
| Z | Z. Included for reference. |
| eta | eta. Included for reference. |
| fullW0 | Can be used to speed up future calls of RUV4. |
| lambda | lambda. Included for reference. |
| invsvd | Can be used to speed up future calls of RUV(r)inv. |
| include.intercept | include.intercept. Included for reference. |
| method | Character variable with value "RUVinv". Included for reference. (Note that RUVrinv is simply a wrapper to RUVinv, hence both return "RUVinv" as the method.) |

## Note

Additional resources can be found at http://www-personal.umich.edu/~johanngb/ruv/.

## Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

## References

Using control genes to correct for unwanted variation in microarray data. Gagnon-Bartsch and Speed, 2012. Available at: http://biostatistics.oxfordjournals.org/content/13/3/539.full.

Removing Unwanted Variation from High Dimensional Data with Negative Controls. Gagnon-Bartsch, Jacob, and Speed, 2013. Available at: http://statistics.berkeley.edu/tech-reports/820.

## See Also

RUV2, RUV4, RUVinv, variance_adjust, invvar, getK

## Examples

```
## Create some simulated data
m = 50
n = 10000
nc = 1000
p = 1
k = 20
ctl = rep(FALSE, n)
ctl[1:nc] = TRUE
X = matrix(c(rep(0,floor(m/2)), rep(1,ceiling(m/2))), m, p)
beta = matrix(rnorm(p*n), p, n)
beta[,ctl] = 0
W = matrix(rnorm(m*k),m,k)
alpha = matrix(rnorm(k*n),k,n)
epsilon = matrix(rnorm(m*n),m,n)
Y = X%*%beta + W%*%alpha + epsilon

## Run RUV-rinv
fit = RUVrinv(Y, X, ctl)

## Get adjusted variances and p-values
fit = variance_adjust(fit)
```

---

# ruv_cancorplot *RUV Canonical Correlation Plot*

---

## Description

Canonical correlation plot

## Usage

```
ruv_cancorplot(Y, X, ctl, W1 = NULL, W2 = NULL)
```

### Arguments

| | |
|---|---|
| Y | The data matrix. Rows are observations and columns are features (e.g. genes). |
| X | Factor(s) of interest. Can be a vector, factor, matrix, or dataframe. Must have the same length (or number of rows) as the number of row of Y. |
| ctl | Index of negative controls. |
| W1 | Optional. The left singular vectors of Y. Can be included to speed up execution. |
| W2 | Optional. The left singular vectors of Y[,ctl]. Can be included to speed up execution. |

### Details

Plots, as a function of k, the square of of the first canonical correlation of X and the first k left singular vectors of Y (and also, similarly, Y[,ctl]).

### Value

A ggplot.

### Author(s)

Johann Gagnon-Bartsch

---

## ruv_ecdf *RUV P-value Empirical CDF Plot*

---

### Description

Plots an ECDF of p-values returned by a call to ruv_summary

### Usage

```
ruv_ecdf(fit, X.col = "all", power = 1, uniform.lines = 0)
```

### Arguments

| | |
|---|---|
| fit | The results of a call to ruv_summary. |
| X.col | Which column of the X matrix to make the plot for, i.e. which factor's p-values to plot. Can be either an integer or a character string. Or, if "all" (the default), use the F-test p-values. |
| power | A power transformation of the x and y axes. For example, set to 1/2 for a square-root transformation. This can help to see the behavior of the ECDF near 0. |
| uniform.lines | A vector of values between 0 and 1, or NULL. If specified, light gray lines will be drawn, showing (locally) uniform distributions. |

## Value

A ggplot.

## Author(s)

Johann Gagnon-Bartsch

---

`ruv_hist` *RUV P-value Histogram Plot*

---

## Description

Plots a histogram of p-values returned by a call to `ruv_summary`

## Usage

```
ruv_hist(fit, X.col = "all", breaks = c(0, 0.001, 0.01, 0.05, seq(0.1, 1, by = 0.1)))
```

## Arguments

| | |
|---|---|
| `fit` | The results of a call to `ruv_summary`. |
| `X.col` | Which column of the X matrix to make the plot for, i.e. which factor's p-values to plot. Can be either an integer or a character string. Or, if "all" (the default), use the F-test p-values. |
| `breaks` | Breakpoints of the histogram. |

## Value

A ggplot.

## Author(s)

Johann Gagnon-Bartsch

## ruv_projectionplot *RUV Projection Plot*

### Description

Projection plot of an RUV regression fit.

### Usage

```
ruv_projectionplot(fit, X.col = 1, factor = "gradient", adjusted = TRUE)
```

### Arguments

| | |
|---|---|
| fit | The results of a call to ruv_summary. |
| X.col | Which column of the X matrix to make the plot for. Can be either an integer or a character string. |
| factor | Which unwanted factor to use (horizontal axis). Must be either an integer or the character string "gradient". |
| adjusted | Whether the plot should be adjusted for unwanted factors other than the one being plotted. Not relevant when factor = "gradient". |

### Value

A ggplot.

### Author(s)

Johann Gagnon-Bartsch

## ruv_rankplot *RUV Rank Plot*

### Description

A plot showing the number of positive controls to be found within the N top-ranked features, as a function of N. The ranking of the features is by p-value.

### Usage

```
ruv_rankplot(fit, pctl, X.col = "all", uniform.lines = 0)
```

### Arguments

| | |
|---|---|
| fit | The results of a call to ruv_summary. |
| pctl | Either an integer or character string specifying which column of fit$C to be used as positive controls. (Must be a logical vector). Alternatively, may some other index vector specifying the positive controls; importantly, in this case, the index vector must index the features as they are sorted in fit$C. |
| X.col | Which column of the X matrix to make the plot for. Can be either an integer or a character string. Or, if "all" (the default), use the F-test p-values. |
| uniform.lines | A vector of values between 0 and 1, or NULL. If specified, light gray lines will be drawn, showing (locally) uniform distributions. |

### Value

A ggplot.

### Author(s)

Johann Gagnon-Bartsch

---

## ruv_residuals *RUV Residuals*

---

### Description

Calculate the residuals or adjusted data matrix of an RUV2 or RUV4 fit.

### Usage

```
ruv_residuals(fit, type=c("residuals", "adjusted.Y"), subset_and_sort=TRUE)
```

### Arguments

| | |
|---|---|
| fit | The results of a call to ruv_summary. |
| type | Whether to compute residuals or an adjusted data matrix. Caution; see details below. |
| subset_and_sort | Whether to subset and sort the features, as in ruv_summary. |

### Details

This function will return either the residuals or an adjusted data matrix. The residuals are the result of removing all factors (wanted and unwanted), whereas the adjusted data matrix is the result of removing only the unwanted factors.

The residuals can be useful for diagnostics, e.g. in producing a residual RLE plot. The adjusted data matrix may also be useful for diagnostics, but typically should *not* be used for any additional downstream analyses. The adjusted data matrix can suffer from overfitting, which can be severe, especially when k is large, and this can produce artificially "good" results in downstream analyses.

If an adjusted data matrix for use in downstream analyses is desired, see RUVIII.

## Value

Either a matrix of residuals, or an adjusted data matrix.

## Author(s)

Johann Gagnon-Bartsch

## See Also

RUV2, RUV4, ruv_summary, RUVIII

---

`ruv_rle` *RUV RLE Plot*

---

## Description

An RLE (Relative Log Expression) Plot

## Usage

```
ruv_rle(Y, rowinfo = NULL, probs = c(0.05, 0.25, 0.5, 0.75, 0.95), ylim = c(-0.5, 0.5))
```

## Arguments

| | |
|---|---|
| `Y` | The data matrix. Rows are observations and columns are features (e.g. genes). |
| `rowinfo` | A dataframe of information about the observations. Should have the same number of rows as Y. This information will be included in the ggplot, and can be used for setting aesthetics such as color. |
| `probs` | The percentiles used to construct the boxplots. By default, whiskers are drawn to the 5th and 95th percentiles. Note that this is non-standard for boxplots. |
| `ylim` | Limits of the y axis. Defaults to (-0.5, 0.5) so that the plots are always on the same scale and can be easily compared. |

## Value

A ggplot.

## Author(s)

Johann Gagnon-Bartsch

## References

Irizarry, R. A., Bolstad, B. M., Collin, F., Cope, L. M., Hobbs, B., and Speed, T. P. (2003). Summaries of Affymetrix GeneChip probe level data. Nucleic acids research, 31(4):e15.

## ruv_scree *RUV Scree Plot*

### Description

A scree plot (on the log scale)

### Usage

```
ruv_scree(Y = NULL, Z = 1, Y.svd = NULL)
```

### Arguments

| | |
|---|---|
| Y | The data matrix. Rows are observations and columns are features (e.g. genes). If not specified, Y.svd must be specified instead (which is faster). |
| Z | Any variables to regress out of Y as a preprocessing step. May simply be 1 (the default) for an intercept term, i.e. the columns of Y are mean centered. May also be NULL. |
| Y.svd | The SVD of Y, as returned by the svd function. |

### Details

Because 0 cannot be plotted on a log scale, if any singular values are equal to 0, they will be changed to the minimum non-zero singular value and plotted in red. Exception: singular values that are 0 as a result of regressing out Z are simply not plotted.

### Value

A ggplot.

### Author(s)

Johann Gagnon-Bartsch

## ruv_shiny *RUV Shiny App*

### Description

A Shiny App that allows quick exploration of a dataset using RUV methods.

### Usage

```
ruv_shiny(Y, rowinfo, colinfo, options = list(port = 3840))
```

### Arguments

| | |
|---|---|
| Y | The data matrix. Rows are observations and columns are features (e.g. genes). |
| rowinfo | A dataframe of information about the observations. Should have the same number of rows as Y. Should contain at least one column that can be used as either a factor of interest or to define replicates. |
| colinfo | A dataframe of information about the observations. Should have the same number of rows as Y. Should contain at least one column that is a logical vector that can be used to define negative controls. |
| options | A list of options to pass to the shinyApp function. |

### Value

None. Calls shinyApp.

### Author(s)

Johann Gagnon-Bartsch

---

## ruv_summary *RUV Summary*

---

### Description

Post-process and summarize the results of call to RUV2, RUV4, RUVinv, or RUVrinv.

### Usage

```
ruv_summary(Y, fit, rowinfo=NULL, colinfo=NULL, colsubset=NULL, sort.by="F.p",
            var.type=c("ebayes", "standard", "pooled"),
            p.type=c("standard", "rsvar", "evar"), min.p.cutoff=10e-25)
```

### Arguments

| | |
|---|---|
| Y | The original data matrix used in the call to RUV2/4/inv/rinv |
| fit | A RUV model fit (a list), as returned by RUV2 / RUV4 / RUVinv / RUVrinv |
| rowinfo | A matrix or dataframe containing information about the rows (samples). This information is included in the summary that is returned. |
| colinfo | A matrix or dataframe containing information about the columns (features, e.g. genes). This information is included in the summary that is returned. |
| colsubset | A vector indexing the features of interest. Only only data on these features will be returned. |
| sort.by | An index variable; which column of C (see below) should be used to sort the features. The default is "F.p", meaning that features will be sorted by the F-test p-value. If NULL, features will note be sorted. |

| | |
|---|---|
| var.type | Which type of estimate for sigma2 should be used from the call to variance_adjust? The options are "ebayes", "standard", or "pooled." See variance_adjust for details. |
| p.type | Which type of p-values should be used from the call to variance_adjust? The options are "standard", "rsvar", or "evar". |
| min.p.cutoff | p-values below this value will be changed and set equal to this value. Useful for plotting p-values on a log scale. |

## Details

This function post-processes the results of a call to RUV2/4/inv/rinv and then nicely summarizes the output. The post-processing step primarily consists of a call to variance_adjust, which computes various adjustments to variances, t-statistics, and and p-values. See variance_adjust for details. The var.type and p.type options determine which of these adjustments are used. An additional post-processing step is that the column means of the Y matrix are computed, both before and after the call to RUV1 (if eta was specified).

After post-processing, the results are summarized into a list containing 4 objects: 1) the data matrix Y; 2) a dataframe R containing information about the rows (samples); 3) a dataframe C containing information about the columns (features, e.g. genes), and 4) a list misc of other information returned by RUV2/4/inv/rinv.

Finally, if colsubset is specified, then C is subset to include only the features of interest (as are the relevant entries of misc that are used to compute projection plots). If sort.by is specified, the features will also be sorted.

## Value

A list containing:

| | |
|---|---|
| Y | The original data matrix. |
| R | A dataframe of row-wise information, including X, Z, and any other data passed in with rowinfo |
| C | A dataframe of column-wise information, including p-values, estimated regression coefficients, estimated variances, column means, an index of the negative controls, and any other data passed in with colinfo. |
| misc | A list of additional information returned by RUV2/4/inv/rinv |

## Author(s)

Johann Gagnon-Bartsch

## See Also

RUV2, RUV4, RUVinv, RUVrinv, variance_adjust

ruv_svdgridplot *RUV SVD Grid Plot*

## Description

A plot composed of a grid of several subplots created by ruv_svdplot

## Usage

```
ruv_svdgridplot(Y.data, Y.space = NULL, rowinfo = NULL, colinfo = NULL, k = 1:3, Z = 1,
                left.additions = NULL, right.additions = NULL,
                factor.labels = paste("S.V.", k))
```

## Arguments

| | |
|---|---|
| Y.data | The data matrix. Rows are observations and columns are features (e.g. genes). |
| Y.space | Either a data matrix of the same dimension as Y.data, or the SVD of such a matrix, as returned by the svd function. The singular vectors of this matrix define the space in which Y.data will be plotted. If NULL, Y.data itself is used. |
| rowinfo | A dataframe of information about the observations. Should have the same number of rows as Y. This information will be included in the ggplots, and can be used for setting aesthetics such as color. |
| colinfo | A dataframe of information about the observations. Should have a number of rows equal to the number of columns of Y. This information will be included in the ggplots, and can be used for setting aesthetics such as color. |
| k | A numeric vector of the singular vectors to be plotted. Typically integers, but fractional values can also be specified. For example, a value of 2.5 corresponds to the linear combination (singular vector 2) + (singular vector 3), rescaled to have unit length. Similarly, a value of 2.2 corresponds to the (rescaled) linear combination 8*(singular vector 2) + 2*(singular vector 3), and -2.2 corresponds to the (rescaled) linear combination 8*(singular vector 2) - 2*(singular vector 3). Note that the vectors defined by 2.2 and -2.8 are orthogonal to each other, as are those defined by 2.3 and -2.7, etc. |
| Z | Any variables to regress out of Y.data as a preprocessing step. May simply be 1 (the default) for an intercept term, i.e. the columns of Y are mean centered. May also be NULL. Similarly for Y.space, unless Y.space is already an SVD. |
| left.additions | A list of additions to the ggplots of the left singular vectors. Can be used to set aesthetics such as color, etc. |
| right.additions | A list of additions to the ggplots of the right singular vectors. Can be used to set aesthetics such as color, etc. |
| factor.labels | The factor labels. |

## Details

Plots of the left singular vectors are shown on the left, and plots of the right singular vectors are shown on the right. The diagonal shows squares with side lengths proportional to the singular values.

## Value

A ggplot.

## Author(s)

Johann Gagnon-Bartsch

---

`ruv_svdplot` *RUV SVD Plot*

---

## Description

A generalization of a PC (principal component) plot.

## Usage

```
ruv_svdplot(Y.data, Y.space = NULL, info = NULL, k = c(1, 2), Z = 1, left = TRUE)
```

## Arguments

| | |
|---|---|
| `Y.data` | The data matrix. Rows are observations and columns are features (e.g. genes). |
| `Y.space` | Either a data matrix of the same dimension as `Y.data`, or the SVD of such a matrix, as returned by the `svd` function. The singular vectors of this matrix define the space in which `Y.data` will be plotted. If NULL, `Y.data` itself is used. |
| `info` | Additional data to be included in the ggplot, which can be used for setting aesthetics such as color. Converted to a dataframe, which should have a number of rows equal to the number of rows of `Y.data` (if `left=TRUE`) or the number of columns of `Y.data` (if `left=FALSE`). |
| `k` | A numeric vector of length 2. The singular vectors to be plotted. Typically integers, but fractional values can also be specified. For example, a value of 2.5 corresponds to the linear combination (singular vector 2) + (singular vector 3), rescaled to have unit length. Similarly, a value of 2.2 corresponds to the (rescaled) linear combination 8*(singular vector 2) + 2*(singular vector 3), and -2.2 corresponds to the (rescaled) linear combination 8*(singular vector 2) - 2*(singular vector 3). Note that the vectors defined by 2.2 and -2.8 are orthogonal to each other, as are those defined by 2.3 and -2.7, etc. |
| `Z` | Any variables to regress out of `Y.data` as a preprocessing step. May simply be 1 (the default) for an intercept term, i.e. the columns of Y are mean centered. May also be `NULL`. Similarly for `Y.space`, unless `Y.space` is already an SVD. |
| `left` | Plot the left singular vectors (if TRUE) or the right singular vectors (if FALSE). |

### Details

When `Y.space = NULL` and `Z = 1` and the values of `k` are integers, this is a standard PC plot.

### Value

A ggplot.

### Author(s)

Johann Gagnon-Bartsch

---

`ruv_varianceplot` *RUV Variance Plot*

---

### Description

A scatter plot of (squared) coefficient estimates against variance estimates.

### Usage

```
ruv_varianceplot(fit, X.col = 1, power = 1/4)
```

### Arguments

| | |
|---|---|
| `fit` | The results of a call to `ruv_summary`. |
| `X.col` | Which column of the X matrix to make the plot for. Can be either an integer or a character string. |
| `power` | Power transformation of the x and y axes. Default is fourth root. |

### Details

A black curve is also plotted, showing the estimated variances of the coefficient estimates.

### Value

A ggplot.

### Author(s)

Johann Gagnon-Bartsch

## ruv_volcano *RUV Volcano Plot*

### Description

A scatter plot of negative log p-values against coefficient estimates, commonly known as a volcano plot

### Usage

```
ruv_volcano(fit, X.col = 1)
```

### Arguments

| | |
|---|---|
| fit | The results of a call to ruv_summary. |
| X.col | Which column of the X matrix to make the plot for. Can be either an integer or a character string. |

### Value

A ggplot.

### Author(s)

Johann Gagnon-Bartsch

## sigmashrink *Empirical Bayes shrinkage estimate of sigma^2*

### Description

This function (re)implements the empirical bayes shrinkage estimate of Smyth (2004), which is also implemented in the Limma package. This function is normally called from the function variance_adjust, and is not normally intended for stand-alone use.

### Usage

```
sigmashrink(s2, d)
```

### Arguments

| | |
|---|---|
| s2 | "Standard" estimates of sigma^2 |
| d | "Standard" degrees of freedom of the residuals |

## Value

A list containing

| | |
|---|---|
| sigma2 | Estimates of sigma^2 using the empirical bayes shrinkage method of Smyth (2004) |
| df | Estimate of degrees of freedom using the empirical bayes shrinkage method of Smyth (2004) |

## Author(s)

Johann Gagnon-Bartsch <johanngb@umich.edu>

## References

Linear models and empirical bayes methods for assessing differential expression in microarray experiments. Smyth, 2004.

Using control genes to correct for unwanted variation in microarray data. Gagnon-Bartsch and Speed, 2012. Available at: http://biostatistics.oxfordjournals.org/content/13/3/539.full.

Removing Unwanted Variation from High Dimensional Data with Negative Controls. Gagnon-Bartsch, Jacob, and Speed, 2013. Available at: http://statistics.berkeley.edu/tech-reports/820.

## See Also

variance_adjust

---

# variance_adjust *Adjust Estimated Variances*

---

## Description

Calculate rescaled variances, empirical variances, etc. For use with RUV model fits.

## Usage

```
variance_adjust(fit, ctl.idx = NULL, ebayes = TRUE, pooled=TRUE, evar = TRUE,
                rsvar = TRUE, bin = 10, rescaleconst = NULL)
```

## Arguments

| | |
|---|---|
| fit | A RUV model fit (a list), as returned by RUV2 / RUV4 / RUVinv / RUVrinv |
| ctl.idx | An index vector to specify the negative controls for use with the rescaled variances method. If unspecified, by default fit$ctl is used. |
| ebayes | A logical variable. Should empirical bayes variance estimates be calculated? |
| pooled | A logical variable. Should pooled variance estimates be calculated? |
| evar | A logical variable. Should empirical variance estimates be calculated? |

| | |
|---|---|
| rsvar | A logical variable. Should rescaled variance estimates be calculated? |
| bin | The bin size to use when calculating empirical variances. |
| rescaleconst | Can be used to speed up execution. See get_empirical_variances. |

## Value

An RUV model fit (a list). In addition to the elements of the list returned by RUV2 / RUV4 / RUVinv / RUVrinv, the list will now contain:

| | |
|---|---|
| sigma2.ebayes | Estimates of sigma^2 using the empirical bayes shrinkage method of Smyth (2004) |
| df.ebayes | Estimate of degrees of freedom using the empirical bayes shrinkage method of Smyth (2004) |
| sigma2.pooled | Estimate of sigma^2 pooled (averaged) over all genes |
| df.pooled | Degrees of freedom for pooled estimate |
| varbetahat | "Standard" estimate of the variance of betahat |
| varbetahat.rsvar | "Rescaled Variances" estimate of the variance of betahat |
| varbetahat.evar | "Empirical Variances" estimate of the variance of betahat |
| varbetahat.ebayes | "Empirical Bayes" estimate of the variance of betahat |
| varbetahat.rsvar.ebayes | "Rescaled Empirical Bayes" estimate of the variance of betahat |
| varbetahat.pooled | "Pooled" estimate of the variance of betahat |
| varbetahat.rsvar.pooled | "Rescaled pooled" estimate of the variance of betahat |
| varbetahat.evar.pooled | Similar to the above, but all genes used to determine the rescaling, not just control genes |
| p.rsvar | P-values, after applying the method of rescaled variances |
| p.evar | P-values, after applying the method of empirical variances |
| p.ebayes | P-values, after applying the empirical bayes method of Smyth (2004) |
| p.pooled | P-values, after pooling variances |
| p.rsvar.ebayes | P-values, after applying the empirical bayes method of Smyth (2004) and the method of rescaled variances |
| p.rsvar.pooled | P-values, after pooling variances and the method of rescaled variances |
| p.evar.pooled | Similar to the above, but all genes used to determine the rescaling, not just control genes |
| Fpvals.ebayes | F test p-values, after applying the empirical bayes method of Smyth (2004) |
| Fpvals.pooled | F test p-values, after pooling variances |
| p.BH | FDR-adjusted p-values |

`Fpvals.BH` FDR-adjusted p-values (from F test)

`p.rsvar.BH` FDR-adjusted p-values (from p.rsvar)

`p.evar.BH` FDR-adjusted p-values (from p.evar)

`p.ebayes.BH` FDR-adjusted p-values (from p.ebayes)

`p.rsvar.ebayes.BH` FDR-adjusted p-values (from p.rsvar.ebayes)

`Fpvals.ebayes.BH` FDR-adjusted F test p-values (from Fpvals.ebayes)

`p.pooled.BH` FDR-adjusted p-values (from p.pooled)

`p.rsvar.pooled.BH` FDR-adjusted p-values (from p.rsvar.pooled)

`p.evar.pooled.BH` FDR-adjusted p-values (from p.evar.pooled)

`Fpvals.pooled.BH` FDR-adjusted F test p-values (from Fpvals.pooled)

## Author(s)

Johann Gagnon-Bartsch

## References

Using control genes to correct for unwanted variation in microarray data. Gagnon-Bartsch and Speed, 2012. Available at: http://biostatistics.oxfordjournals.org/content/13/3/539.full.

Removing Unwanted Variation from High Dimensional Data with Negative Controls. Gagnon-Bartsch, Jacob, and Speed, 2013. Available at: http://statistics.berkeley.edu/tech-reports/820.

Linear models and empirical bayes methods for assessing differential expression in microarray experiments. Smyth, 2004.

## See Also

`RUV2`, `RUV4`, `RUVinv`, `RUVrinv`, `get_empirical_variances`, `sigmashrink`

## Examples

```
## Create some simulated data
m = 50
n = 10000
nc = 1000
p = 1
k = 20
ctl = rep(FALSE, n)
ctl[1:nc] = TRUE
X = matrix(c(rep(0,floor(m/2)), rep(1,ceiling(m/2))), m, p)
beta = matrix(rnorm(p*n), p, n)
beta[,ctl] = 0
W = matrix(rnorm(m*k),m,k)
```

```
alpha = matrix(rnorm(k*n),k,n)
epsilon = matrix(rnorm(m*n),m,n)
Y = X%*%beta + W%*%alpha + epsilon

## Run RUV-inv
fit = RUVinv(Y, X, ctl)

## Get adjusted variances and p-values
fit = variance_adjust(fit)
```

# Index