

CpGassoc

July 17, 2024

`cpg.assoc` *Association Analysis Between Methylation Beta Values and Phenotype of Interest*

Usage

`cpg.assoc(beta.val, indep, covariates = NULL, data = NULL, logit.transform = FALSE, chip.id = NULL, subset = NULL, random = FALSE, fdr.cutoff = 0.05, large.data = TRUE, fdr.method = "BH", logitperm = FALSE)`

Arguments

- | | |
|------------------------------|---|
| <code>beta.val</code> | A vector, matrix, or data frame containing the beta values of interest (1 row per CpG site, 1 column per individual). |
| <code>indep</code> | A vector containing the variable to be tested for association. <code>cpg.assoc</code> will evaluate the association between the beta values (dependent variable) and <code>indep</code> (independent variable). |
| <code>covariates</code> | A data frame consisting of additional covariates to be included in the model. <code>covariates</code> can also be specified as a matrix if it takes the form of a model matrix with no intercept column, or can be specified as a vector if there is only one covariate of interest. Can also be a formula(e.g. <code>cov1+cov2</code>). |
| <code>data</code> | an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in <code>data</code> , the variables are taken from the environment from which <code>cpg.assoc</code> is called. |
| <code>logit.transform</code> | Logical. If <code>TRUE</code> , the logit transform of the beta values $\log(\text{beta.val}/(1-\text{beta.val}))$ will be used. Any values equal to zero or one will be set to the next smallest or next largest value respectively; values <0 or >1 will be set to <code>NA</code> . |
| <code>chip.id</code> | An optional vector containing chip or batch identifiers. If specified, <code>chip.id</code> will be included as a factor in the model. |

subset	An optional logical vector specifying a subset of observations to be used in the fitting process.
random	Logical. If <code>TRUE</code> , <code>chip.id</code> will be included in the model as a random effect, and a random intercept model will be fitted. If <code>FALSE</code> , <code>chip.id</code> will be included in the model as an ordinary categorical covariate, for a much faster analysis.
fdr.cutoff	The desired FDR threshold. The default setting is <code>.05</code> . The set of CpG sites with $FDR < \text{fdr.cutoff}$ will be labeled as significant.
large.data	Logical. Enables analyses of large datasets. When <code>large.data=TRUE</code> , <code>cpg.assoc</code> avoids memory problems by performing the analysis in chunks.
fdr.method	Character. Method used to calculate False Discovery Rate. Choices include any of the methods available in <code>p.adjust()</code> . The default method is "BH" for the Benjamini and Hochberg method.
logitperm	Logical. For internal use only.

Details

`cpg.assoc` is designed to test for association between an independent variable and methylation at a number of CpG sites, with the option to include additional covariates and factors. `cpg.assoc` assesses significance with the Holm (step-down Bonferroni) and FDR methods.

If `class(indep)='factor'`, `cpg.assoc` will perform an ANOVA test of the variable conditional on the covariates specified. Covariates, if entered, should be in the form of a data frame, matrix, or vector. For example, `covariates=data.frame(weight,age,factor(city))`. The data frame can also be specified prior to calling `cpg.assoc`. The covariates should either be vectors or columns of a matrix or data.frame.

`cpg.assoc` is also designed to deal with large data sets. Setting `large.data=TRUE` will make `cpg.assoc` split up the data to enable efficient analysis of large datasets.

Value

`cpg.assoc` will return an object of class *cpg*. The functions `summary` and `plot` can be called to get a summary of results and to create QQ plots.

results	A data frame consisting of the t or F statistics and P-values for each CpG site, as well as indicators of Holm and FDR significance. CpG sites will be in the same order as the original input, but the <code>sort()</code> function can be used directly on the <code>cpg.assoc</code> object to sort CpG sites by p-value.
results	A data frame consisting of the t or F statistics and P-values for each CpG site, as well as indicators of Holm and FDR significance. CpG sites will be in the same order as the original input, but the <code>sort()</code> function can be used directly on the <code>cpg.assoc</code> object to sort CpG sites by p-value.
Holm.sig	A list of sites that met criteria for Holm significance.
FDR.sig	A data.frame of the CpG sites that were significant by the FDR method specified.
info	A data frame consisting of the minimum P-value observed, the FDR method that was used, the phenotype of interest, the number of covariates in the model, the name of the matrix or data frame the methylation beta values were taken from, the FDR cutoff value and whether a mixed effects analysis was performed.
indep	The independent variable that was tested for association.
covariates	Data.frame or matrix of covariates, if specified (otherwise NULL).
chip	chip.id vector, if specified (otherwise NULL).

coefficients A data frame consisting of the degrees of freedom, and if object is continous the intercept effect adjusted for possible covariates in the model, the estimated effect size, and the standard error. The degrees of freedom is used in [plot.cpg](#) to compute the genomic inflation factors.

Authors

Barfield, R.; Conneely, K.; Kilaru, V.
Maintainer: R. Barfield: barfieldrichard8@gmail.com

See Also

[cpg.perm](#), [cpg.work](#), [plot.cpg.scatterplot](#), [cpg.combine](#), [manhattan](#), [plot.cpg.perm](#), [sort.cpg.perm](#), [sort.cpg](#), [cpg.qc](#), [cpg.GC](#)

Examples

```
> #Sample output from CpGassoc
> ###NOTE: If you are dealing with large data, do not specify large.data=FALSE. The
> ##This will involve partitioning up the data and performing more gc() to clear up
> library(CpGassoc)
> data(samplecpg, samplepheno, package="CpGassoc")
> results<-cpg.assoc(samplecpg, samplepheno$weight, large.data=FALSE)
> results
```

The top ten CpG sites were:

	CPG.Labels	T.statistic	P.value	Holm.sig	FDR	gc.p.value
694	CpG694	3.454271	0.0006456268	FALSE	0.4318310	0.0006456268
293	CpG293	3.412320	0.0007485123	FALSE	0.4318310	0.0007485123
560	CpG560	3.313353	0.0010549618	FALSE	0.4318310	0.0010549618
148	CpG148	3.133454	0.0019286973	FALSE	0.5645412	0.0019286973
998	CpG998	-3.079596	0.0022986204	FALSE	0.5645412	0.0022986204
1059	CpG1059	-2.883525	0.0042668430	FALSE	0.7693539	0.0042668430
1182	CpG1182	-2.819710	0.0051827097	FALSE	0.7693539	0.0051827097
100	CpG100	2.787987	0.0057015107	FALSE	0.7693539	0.0057015107
751	CpG751	-2.759379	0.0062093208	FALSE	0.7693539	0.0062093208
238	CpG238	2.756367	0.0062650966	FALSE	0.7693539	0.0062650966

To access results for all 1228 CpG sites use `object$results` or `sort(object)$results` to obtain results sorted by p-value.

General info:

```
  Min.P.Observed Num.Cov fdr.cutoff FDR.method Phenotype chipinfo num.Holm
1  0.0006456268      0      0.05          BH    weight    NULL      0
  num.fdr
1      0
```

0 sites were found significant by the Holm method

0 sites were found significant by BH method

The beta values were taken from: `samplecpg`

Effect sizes and standard error can be accessed using `$coefficients`

Other attributes are: `results`, `Holm.sig`, `FDR.sig`, `info`

They can be accessed using the `$`

```
> #Analysis with covariates. There are multiple ways to do this. One can define the
> #dataframe prior or do it in the function call or as a function such as ~Cov1+Cov2
> #We will do it in the function call
> test<-cpg.assoc(samplecpg,samplepheno$weight,data.frame(samplepheno$Distance,samp
> #Doing a mixed effects model. This does take more time, so we will do a subset of
> #the samplecpg
> randtest<-cpg.assoc(samplecpg[1:10,],samplepheno$weight,chip.id=samplepheno$chip,1
>
> #summary function will work on items of class cpg.
>
>
```

`cpg.combine`

Combine various objects of class `cpg`

Description

Takes a list containing objects of class `cpg` and combines them into one `cpg` item. Assumes that there are no repeated CpG sites between the various objects (i.e. analysis wasn't performed on the same sites twice).

usage

```
cpg.combine(allvalues, fdr.method="BH",fdr.cutoff=.05)
```

Arguments

allvalues	A list containing the <i>cpg</i> objects that are desired to be consolidated.
fdr.method	FDR method that user wants to use. For options see the <code>cpg.assoc</code> help page.
fdr.cutoff	The desired FDR threshold. The default setting is .05. The set of CpG sites with $FDR < fdr.cutoff$ will be labeled as significant.

Value

indo.data	An object of class <i>cpg</i> that is the consolidated version of the objects of class <i>cpg</i> that were passed in.
-----------	--

Authors

Barfield, R.;Conneely, K.; Kilaru,V.
 Maintainer: R. Barfield: barfieldrichard8@gmail.com

Note

This is designed to be used by `cpg.assoc` when it does analysis on large data sets or by the user if they split up the analysis by chromosome or some other such partition.

See Also

[cpg.perm](#), [cpg.work](#), [plot.cpg.scatterplot](#), [cpg.assoc](#), [manhattan](#), [plot.cpg.perm](#), [sort.cpg.perm](#), [sort.cpg](#)

Examples

```
> library(CpGassoc)
> data(samplecpg,samplepheno,package="CpGassoc")
> ###NOTE: If you are dealing with large data, do not specify large.data=FALSE. The
> ##This will involve partitioning up the data and performing more gc() to clear up
> test1<-cpg.assoc(samplecpg[1:100,],samplepheno$weight,large.data=FALSE)
> test2<-cpg.assoc(samplecpg[101:200,],samplepheno$weight,large.data=FALSE)
> overall<-cpg.combine(list(test1,test2))
> overall
```

The top ten CpG sites were:

	CpG.Labels	T.statistic	P.value	Holm.sig	FDR	gc.p.value
148	CpG148	3.133454	0.001928697	FALSE	0.3857395	0.008032723
100	CpG100	2.787987	0.005701511	FALSE	0.5701511	0.018186157
52	CpG52	-2.400358	0.017093566	FALSE	0.6753972	0.041721245
3	CpG3	-2.307436	0.021828750	FALSE	0.6753972	0.050222867
85	CpG85	2.289916	0.022840129	FALSE	0.6753972	0.051979447
72	CpG72	-2.093410	0.037296699	FALSE	0.6753972	0.075466953
153	CpG153	-2.080196	0.038502367	FALSE	0.6753972	0.077318076
178	CpG178	-2.055509	0.040844281	FALSE	0.6753972	0.080876123
70	CpG70	-2.023648	0.044045272	FALSE	0.6753972	0.085664559
35	CpG35	-2.000859	0.046463353	FALSE	0.6753972	0.089228937

To access results for all 200 CpG sites use `object$results`
or `sort(object)$results` to obtain results sorted by p-value.

General info:

	Min.P.Observed	Num.Cov	fdr.cutoff	FDR.method	Phenotype	chipinfo	num.Holm
1	0.001928697	0	0.05	BH	weight	NULL	0
	num.fdr						
1	0						

0 sites were found significant by the Holm method

0 sites were found significant by BH method

The beta values were taken from: `samplecpg`

Effect sizes and standard error can be accessed using `$coefficients`

Other attributes are: `results`, `Holm.sig`, `FDR.sig`, `info`, `coefficients`

They can be accessed using the `$`

<code>cpg.perm</code>	<i>Perform a Permutation Test of the Association Between Methylation and a Phenotype of Interest</i>
-----------------------	--

Description

Calls `cpg.assoc` to get the observed P-values from the study and then performs a user-specified number of permutations to calculate an empirical p-value. In addition to the same test statistics computed by `cpg.assoc`, `cpg.perm` will compute the permutation p-values for the observed p-value, the number of Holm significant sites, and the number of FDR significant sites.

Usage

```
cpg.perm(beta.values, indep, covariates = NULL, nperm, data = NULL, seed = NULL,  
logit.transform = FALSE, chip.id = NULL, subset = NULL, random = FALSE,  
fdr.cut.off = 0.05, fdr.method = "BH", large.data=TRUE)
```

Arguments

beta.values	A vector, matrix, or data frame containing the beta values of interest (1 row per CpG site, 1 column per individual).
indep	A vector containing the main variable of interest. <code>cpg.assoc</code> will evaluate the association between <code>indep</code> and the beta values.
covariates	A data frame consisting of the covariates of interest. <code>covariates</code> can also be a matrix if it is a model matrix minus the intercept column. It can also be a vector if there is only one covariate of interest. Can also be a formula(e.g. <code>cov1+cov2</code>).
nperm	The number of permutations to be performed.
data	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in data, the variables are taken from the environment from which <code>cpg.perm</code> is called.
seed	The required seed for random number generation. If not input, will use R's internal seed.
logit.transform	Logical. If <code>TRUE</code> , the logit transform of the beta values $\log(\text{beta.val}/(1-\text{beta.val}))$ will be used. Any values equal to zero or one will be set to the next smallest or next largest value respectively; values <0 or >1 will be set to NA.
chip.id	An optional vector containing the chip information. If specified, <code>chip.id</code> will be included as a factor in the model.
subset	An optional logical vector specifying a subset of observations to be used in the fitting process.
random	Logical. If <code>TRUE</code> , the <code>chip.id</code> will be processed as a random effect, and a random intercept model will be fitted.
fdr.cut.off	The threshold at which to compare the FDR values. The default setting is <code>.05</code> . Any FDR values less than <code>.05</code> will be considered significant.
fdr.method	Character. Method used to calculate False Discovery Rate. Can be any of the methods listed in <code>p.adjust</code> . The default method is "BH" for the Benjamini and Hochberg method.
large.data	Logical. Enables analyses of large datasets. When <code>large.data=TRUE</code> , <code>cpg.assoc</code> avoids memory problems by performing the analysis in chunks.

Value

The item returned will be of class *cpg.perm*. It will contain all of the values of class *cpg* *cpg.assoc* and a few more:

permutation.matrix	A matrix consisting of the minimum observed P-value, the number of Holm significant CpG sites, and the number of FDR significant sites for each permutation.
perm.p.values	A data frame consisting of the permutation P-values, and the number of permutations performed.
perm.tstat	If one hundred or more permutations were performed and indep is a continuous variable, consists of the quantile .025 and .975 of observed t-statistics for each permutation, ordered from smallest to largest. perm.tstat is used by <code>plot.cpg.perm</code> to compute the confidence intervals for the QQ plot of t-statistics. Otherwise NULL.
perm.pval	If one hundred or more permutations were performed, consists of the observed p-values for each permutation, ordered from smallest to largest. perm.pval is used by <code>plot.cpg.perm</code> to compute the confidence intervals for the QQ plot of the p-values. Otherwise NULL.
gc.permutation.matrix	Similar to the permutation.matrix only in relation to the genomic control adjusted p-values.

Authors

Barfield, R.; Conneely, K.; Kilaru, V.

Maintainer: R. Barfield: barfieldrichard8@gmail.com

See Also

[cpg.assoc](#), [cpg.work](#), [plot.cpg.scatterplot](#), [cpg.combine](#), [manhattan](#), [plot.cpg.perm](#), [sort.cpg.perm](#), [sort.cpg.qc](#), [cpg.GC](#)

Examples

```
> ##Loading the data
> library(CpGassoc)
> data(samplecpg, samplepheno, package="CpGassoc")
> ###NOTE: If you are dealing with large data, do not specify large.data=FALSE. The default option is
> ##This will involve partitioning up the data and performing more gc() to clear up space
> #Performing a permutation 10 times
> Testperm<-cpg.perm(samplecpg, samplepheno$weight, data.frame(samplepheno$Dose, samplepheno$Distance),
+                    seed=2314, nperm=10, large.data=FALSE)
> Testperm
```

The permutation P-values, number of permutations and seed:

```
  p.value.p p.value.holm p.value.FDR nperm seed
1         0.5           1           1    10 2314
```

Other information:

```

Min.P.Observed Num.Cov fdr.cutoff FDR.method num.real.Holm num.real.fdr
1 0.0006002142 2 0.05 BH 0 0

```

The top ten CpG sites were:

```

CPG.Labels T.statistic P.value Holm.sig FDR gc.p.value
694 CpG694 3.475160 0.0006002142 FALSE 0.3833341 0.0006002142
293 CpG293 3.464076 0.0006243226 FALSE 0.3833341 0.0006243226
560 CpG560 3.333678 0.0009848497 FALSE 0.4031318 0.0009848497
148 CpG148 3.187753 0.0016135434 FALSE 0.4953578 0.0016135434
238 CpG238 3.012760 0.0028504303 FALSE 0.5921086 0.0028504303
998 CpG998 -3.008091 0.0028930386 FALSE 0.5921086 0.0028930386
1059 CpG1059 -2.932014 0.0036749081 FALSE 0.6295151 0.0036749081
100 CpG100 2.889847 0.0041873059 FALSE 0.6295151 0.0041873059
1006 CpG1006 -2.831992 0.0049965867 FALSE 0.6295151 0.0049965867
1182 CpG1182 -2.823521 0.0051263442 FALSE 0.6295151 0.0051263442

```

To access results for all 1228 CpG sites use `object$results`
or `sort(object)$results` to obtain results sorted by p-value.

0 sites were found significant by the Holm method

0 sites were found significant by BH method

The beta values were taken from: `samplecpg`

Other attributes are: `permutation.matrix`, `perm.p.values`, `gc.permutation.matrix`, `results`,
`Holm.sig`, `FDR.sig`, `info`, `coefficients`.

They can be accessed using the `$`

> *#All the contents of CpGassoc are included in the output from Testperm*

> *#Using the output from CpGassoc in the example*

> `test<-cpg.assoc(samplecpg,samplepheno$weight,data.frame(samplepheno$Distance,samplepheno$Dose),lar`

> `all.equal(Testperm$results,test$results)`

[1] TRUE

> *#summary function works on objects of class cpg.perm*

> `summary(Testperm)`

The permutation P-values, number of permutations and seed:

```

p.value.p p.value.holm p.value.FDR nperm seed
1 0.5 1 1 10 2314

```

Other information:

```

Min.P.Observed Num.Cov fdr.cutoff FDR.method num.real.Holm num.real.fdr
1 0.0006002142 2 0.05 BH 0 0

```

The top ten CpG sites were:

```

CPG.Labels T.statistic P.value Holm.sig FDR gc.p.value
694 CpG694 3.475160 0.0006002142 FALSE 0.3833341 0.0006002142
293 CpG293 3.464076 0.0006243226 FALSE 0.3833341 0.0006243226
560 CpG560 3.333678 0.0009848497 FALSE 0.4031318 0.0009848497
148 CpG148 3.187753 0.0016135434 FALSE 0.4953578 0.0016135434
238 CpG238 3.012760 0.0028504303 FALSE 0.5921086 0.0028504303
998 CpG998 -3.008091 0.0028930386 FALSE 0.5921086 0.0028930386
1059 CpG1059 -2.932014 0.0036749081 FALSE 0.6295151 0.0036749081
100 CpG100 2.889847 0.0041873059 FALSE 0.6295151 0.0041873059
1006 CpG1006 -2.831992 0.0049965867 FALSE 0.6295151 0.0049965867

```

```
1182 CpG1182 -2.823521 0.0051263442 FALSE 0.6295151 0.0051263442
```

```
To access results for all 1228 CpG sites use object$results  
or sort(object)$results to obtain results sorted by p-value.
```

```
0 sites were found significant by the Holm method  
0 sites were found significant by BH method
```

```
The beta values were taken from: samplecpg  
Other attributes are: permutation.matrix, perm.p.values, gc.permutation.matrix, results ,  
Holm.sig, FDR.sig, info, coefficients.  
They can be accessed using the $  
>
```

<code>cpg.GC</code>	<i>For genomic control adjusted statistics.</i>
---------------------	---

Description

`cpg.GC` accepts an object of class `cpg.perm` or `cpg` and returns information regarding Holm and FDR-significance of the GC (genomic control) adjusted test statistics. For `cpg.perm` will return permutation p-values based on the GC-adjusted values from each permutation.

Usage

```
cpg.GC(x)
```

Arguments

`x` Object of class `cpg.perm` or `cpg`.

Details

`cpg.GC` will display the number of Holm and FDR-significant sites using the genomic control adjusted p-values test statistics. It will also display the estimated genomic control inflation factor.

Value

`cpg.GC` returns an object of class `cpg.gc` or `cpg.perm.gc`

<code>gc.results</code>	Matrix consisting of GC-adjusted test statistics for each CpG site. Similar to the results output of <code>cpg.assoc</code> .
<code>gc.info</code>	Data frame with information on the number of Holm and FDR significant sites. Will also have the genomic control inflation estimate. Objects from <code>cpg.perm</code> will also have information concerning the permutation p-values.

Authors

Barfield, R.; Conneely, K.; Kilaru, V.
Maintainer: R. Barfield: barfieldrichard8@gmail.com

See Also

[cpg.assoc](#), [cpg.work](#), [plot.cpg.scatterplot](#), [cpg.combine](#), [manhattan](#), [plot.cpg.perm](#),
[sort.cpg.perm](#), [sort.cpg](#), [cpg.qc](#)

Examples

```
> library(CpGassoc)
> data(samplecpg, samplepheno, package="CpGassoc")
> results<-cpg.assoc(samplecpg, samplepheno$weight, large.data=FALSE)
> cpg.GC(results)
```

Using genomic control adjustment the top sites are:

	CPG.Labels	GC.Adjusted	Adjust.P.value	Adj.Holm	Adj.FDR
694	CpG694	3.454271	0.0006456268	FALSE	0.4318310
293	CpG293	3.412320	0.0007485123	FALSE	0.4318310
560	CpG560	3.313353	0.0010549618	FALSE	0.4318310
148	CpG148	3.133454	0.0019286973	FALSE	0.5645412
998	CpG998	-3.079596	0.0022986204	FALSE	0.5645412
1059	CpG1059	-2.883525	0.0042668430	FALSE	0.7693539
1182	CpG1182	-2.819710	0.0051827097	FALSE	0.7693539
100	CpG100	2.787987	0.0057015107	FALSE	0.7693539
751	CpG751	-2.759379	0.0062093208	FALSE	0.7693539
238	CpG238	2.756367	0.0062650966	FALSE	0.7693539

General info:

	num.holm	FDR.method	num.fdr	gcvalue
1	0	BH	0	1

0 sites were found significant by the Holm method

0 sites were found significant by BH method

> ##If the genomic inflation factor is less than one there is no need for adjustment

`cpg.qc` *Performs quality control on Illumina data.*

Description

`cpg.qc` is designed to perform quality control on Illumina data prior to analysis. In addition to the matrix of beta values, this function requires as input matrices of Signal A, Signal B, and detection p-values. It will remove samples that have low intensity (mean signal intensity less than half of the overall median or 2000). It can also set to NA datapoints with detection p-values exceeding a user-specified cutoff, and can remove samples or sites that have a missing rate above a user-specified value. Finally, users can opt to compute beta values as $M/(U+M)$ or $M/(U+M+100)$.

Usage

```
cpg.qc(beta.orig,siga,sigb,pval,p.cutoff=.001,cpg.miss=NULL,sample.miss=NULL,constant100=FALSE)
```

Arguments

<code>beta.orig</code>	The original beta values matrix obtained from GenomeStudio.
<code>sigA</code>	The unmethylated signals matrix obtained from GenomeStudio.
<code>sigB</code>	The methylated signals matrix obtained from GenomeStudio.
<code>pval</code>	A matrix of detection p-values obtained from GenomeStudio. <code>pval</code> should have the same dimension as the beta values and signals: one row for each site and one column for each individual.
<code>p.cutoff</code>	The user-specified cutoff for detection p-values (default=.001).
<code>cpg.miss</code>	Optional cutoff value. If specified, <code>cpg.qc</code> will remove cpg sites where the proportion of missing values exceeds this cutoff.
<code>sample.miss</code>	Optional cutoff value. If specified, <code>cpg.qc</code> will remove samples where the proportion of missing values exceeds this cutoff.
<code>constant100</code>	Logical. If <code>TRUE</code> , the new beta values will be calculated as $M/(U+M+100)$; if <code>FALSE</code> (default) they will be calculated as $M/(U+M)$.

Details

It is important that all the matrices or data frames listed above (`pval`, `sigA`, `sigB`, `beta.orig`) are ordered similarly with respect to samples and CpG sites.

Value

returns a new matrix of beta values that has been subjected to the specified quality control filters. This matrix can be input directly into `cpg.assoc`.

Authors

Barfield, R.; Conneely, K.; Kilaru, V.

Maintainer: R. Barfield: barfieldrichard8@gmail.com

See Also

[cpg.perm](#), [cpg.assoc](#), [plot.cpg](#) scatterplot

Examples

```
> ##See the examples in the CpGassoc tutorial.
```

<code>cpg.work</code>	<i>Does the analysis between the CpG sites and phenotype of interest</i>
-----------------------	--

Description

Association Analysis Between Methylation Beta Values and Phenotype of Interest. This function contains the code that does the brunt of the work for `cpg.assoc` and `cpg.perm`.

Usage

```
cpg.work(beta.values, indep, covariates = NULL, data = NULL, logit.transform = FALSE, chip.id = NULL, subset = NULL, random = FALSE, fdr.cutoff = 0.05, callarge = FALSE, fdr.method = "BH", logitperm = FALSE, big.split=FALSE)
```

Arguments

<code>beta.values</code>	A vector, matrix, or data frame containing the beta values of interest (1 row per CpG site, 1 column per individual).
<code>indep</code>	A vector containing the main variable of interest. <code>cpg.work</code> will evaluate the association between <code>indep</code> and the beta values.
<code>covariates</code>	A data frame consisting of the covariates of interest. <code>covariates</code> can also be a matrix if it is a model matrix minus the intercept column. It can also be a vector if there is only one covariate of interest. Can also be a formula (e.g. <code>cov1+cov2</code>).
<code>data</code>	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in <code>data</code> , the variables are taken from the environment from which <code>cpg.work</code> is called.
<code>logit.transform</code>	Logical. If <code>TRUE</code> , the logit transform of the beta values $\log(\text{beta.val}/(1-\text{beta.val}))$ will be used. Any values equal to zero or one will be set to the next smallest or next largest value respectively; values <0 or >1 will be set to <code>NA</code> .
<code>chip.id</code>	An optional vector containing chip or batch identities. If specified, <code>chip.id</code> will be included as a factor in the model.
<code>subset</code>	an optional logical vector specifying a subset of observations to be used in the fitting process.
<code>random</code>	Logical. If <code>TRUE</code> , the <code>chip.id</code> will be included in the model as a random effect, and a random intercept model will be fitted. If <code>FALSE</code> , <code>chip.id</code> will be included in the model as an ordinary categorical covariate, for a much faster analysis.
<code>fdr.cutoff</code>	The threshold at which to compare the FDR values. The default setting is <code>.05</code> . Any FDR values less than <code>.05</code> will be considered significant.
<code>callarge</code>	Logical. Used by <code>cpg.assoc</code> when it calls <code>cpg.work</code> . If <code>TRUE</code> it means that <code>beta.values</code> is actually split up from a larger data set and that <code>memory.limit</code> may be a problem. This tells <code>cpg.work</code> to perform more <code>rm()</code> and <code>gc()</code> to clear up space.

<code>fdr.method</code>	Character. Method used to calculate False Discovery Rate. Can be any of the methods listed in <code>p.adjust</code> . The default method is "BH" for the Benjamini and Hochberg method.
<code>logitperm</code>	Passes from <code>cpg.perm</code> when permutation test is performed. Stops from future checks involving the logistic transformation.
<code>big.split</code>	Passes from <code>cpg.assoc</code> . Internal flag to inform <code>cpg.work</code> that the large data did not need to be split up.

Details

`cpg.work` does the analysis between the methylation and the phenotype of interest. It is called by `cpg.assoc` to do the brunt of the work. It can be called itself with the same input as `cpg.assoc`, it just cannot handle large data sets.

Value

`cpg.work` will return an object of class `cpg`. The functions `summary` and `plot` can be called to get a summary of results and to create QQ plots. The output is in the same order as the original input. To sort it by p-value, use the `sort` function.

<code>results</code>	A data frame consisting of the statistics and P-values for each CpG site. Also has the adjusted p-value based on the <code>fdr.method</code> and whether the site was Holm significant.
<code>Holm.sig</code>	A list of sites that met criteria for Holm significance.
<code>FDR.sig</code>	A data.frame of the sites that were FDR significant by the <code>fdr</code> method.
<code>info</code>	A data frame consisting of the minimum P-value observed, the <code>fdr</code> method used, what the phenotype of interest was, and the number of covariates in the model.
<code>indep</code>	The main phenotype of interest.
<code>covariates</code>	If <code>covariates</code> was non <code>NULL</code> , the covariates will be included. Otherwise will be <code>NULL</code> .
<code>chip</code>	If <code>chip.id</code> was non <code>NULL</code> , the chip will be included. Otherwise will be <code>NULL</code> .
<code>coefficients</code>	A data frame consisting of the degrees of freedom, and if object is continuous the intercept effect adjusted for possible covariates in the model, the estimated effect size, and the standard error. The degrees of freedom is used in plot.cpg to compute the genomic inflation factors.

Authors

Barfield, R.; Conneely, K.; Kilaru, V.

Maintainer: R. Barfield: barfieldrichard8@gmail.com

See Also

[cpg.perm](#), [cpg.assoc](#), [plot.cpg.scatterplot](#), [cpg.combine](#), [manhattan](#), [plot.cpg.perm](#), [sort.cpg.perm](#), [sort.cpg.qc](#)

Examples

```
> ##See the examples listed in cpg.assoc for ways in which to use cpg.work.  
> ##Just change the cpg.assoc to cpg.work.
```

design	Create full and reduced design matrices for the <i>cpg.assoc</i> function.
--------	--

Description

Designed to be used by `cpg.assoc` and `cpg.perm`. Creates a full and reduced design matrices.

Usage

```
design(covariates, indep, chip.id, random)
```

Arguments

covariates	A data frame consisting of the covariates of interest. <code>covariates</code> can also be a matrix if it is a model matrix minus the intercept column. It can also be a vector if there is only one covariate of interest. If no covariates must be specified as <code>NULL</code> .
indep	A vector containing the main variable of interest. <code>cpg.assoc</code> will evaluate the association between <code>indep</code> and the beta values.
chip.id	An optional vector containing chip or batch identities. If specified, <code>chip.id</code> will be included as a factor in the model.
random	Is the model going to be a mixed effects. If so, <code>chip.id</code> will not be included in the design matrices.

Value

Returns a list containing the full and reduced design matrices.

full	The full design matrix
reduced	The reduced design matrix

Author

Barfield, R.; Kilaru, V.; Conneely, K.

Maintainer: R. Barfield: barfieldrichard8@gmail.com

Note

The design function is designed to be used exclusively by the `cpg.assoc` and `cpg.perm` functions.

See Also

[cpg.assoc](#), [cpg.perm](#), [plot.cpg](#), [cpg.work](#), [scatterplot](#), [cpg.combine](#), [manhattan](#), [plot.cpg.perm](#), [sort.cpg.perm](#), [sort.cpg](#)

examples

```
> library(CpGassoc)
> data(samplecpg, samplepheno, package="CpGassoc")
> #Example where there are covariates:
> covar<-data.frame(samplepheno$weight, samplepheno$Distance)
> test<-design(covar, samplepheno$SBP, samplepheno$chip, FALSE)
> dim(test$full)

[1] 258 26

> dim(test$reduced)

[1] 258 25

> test$reduced[1:5, 1:5]

  (Intercept) samplepheno.weight samplepheno.Distance factor(chip.id)3
1             1             31.02998             28.49084             0
2             1             20.83885             13.10059             0
3             1             21.47078             14.76703             0
4             1             23.95091             25.54482             0
5             1             34.12922             29.45997             0
  factor(chip.id)4
1                 0
2                 0
3                 0
4                 0
5                 0

> test$full[1:5, 1:5]

  (Intercept) indep samplepheno.weight samplepheno.Distance factor(chip.id)3
1             1 16.98629             31.02998             28.49084             0
2             1 34.90645             20.83885             13.10059             0
3             1 21.55838             21.47078             14.76703             0
4             1 20.90882             23.95091             25.54482             0
5             1 27.01004             34.12922             29.45997             0

> #When no covariates or chip.id:
> test2<-design(NULL, samplepheno$SBP, NULL, FALSE)
> dim(test2$full)

[1] 258 2
```

```
> dim(test2$reduced)
[1] 258  1
```

manhattan	Create a manhattan plot
-----------	-------------------------

Description

This function will produce a manhattan plot for the observed P-values from a object of class *cpg* or *cpg.perm*.

Usage

```
manhattan(x, cpgname, chr, pos, save.plot = NULL, file.type="pdf",
popup.pdf = FALSE, eps.size = c(15, 5), main.title = NULL, cpg.labels
= NULL, chr.list = NULL, color.list = NULL, ...)
```

Arguments

<code>x</code>	Object of class <i>cpg</i> or <i>cpg.perm</i> .
<code>cpgname</code>	A vector consisting of the labels for each CpG site.
<code>chr</code>	A vector consisting of the chromosome number for each CpG site.
<code>pos</code>	The map position of each CpG site within its chromosome.
<code>save.plot</code>	Name of the file for the plot to be saved to. If not specified, plot will not be saved.
<code>file.type</code>	Type of file to be saved. Can either be "pdf" or "eps". Selecting <code>file.type="eps"</code> will result in publication quality editable postscript files that can be opened by Adobe Illustrator or Photoshop.
<code>popup.pdf</code>	TRUE or FALSE. If creating a pdf file, this indicates if the plot should appear in a popup window as well. If running in a cluster-like environment, best to leave FALSE.
<code>eps.size</code>	Vector indicating the size of .eps file (if creating one). Corresponds to horrizontal and height.

main.title	Main title to be put on the graph. If <code>NULL</code> one based on the analysis will be used.
cpg.labels	A character scalar of either "FDR" or "HOLM" which will label the significant sites on the manhattan plot.
chr.list	A vector listing the chromosomes to be plotted (all available chromosomes are plotted by default). The X and Y chromosomes can be denoted by 23 and 24
color.list	A vector of custom colors to be used for each chromosomes in the manhattan plot.
...	Arguments to be passed to methods, such as graphical parameters.

Authors

Barfield, R.;Conneely, K.; Kilaru,V.

Maintainer: R. Barfield: barfieldrichard8@gmail.com

Note

`cpgname`, `chr`, and `pos` must be sorted in the same order, so that the first `cpgname[1]` corresponds to `chr[1]` and `pos[1]`, and so on.

See Also

[cpg.assoc](#), [cpg.perm](#), [plot.cpg](#), [cpg.work](#), [scatterplot](#), [cpg.combine](#), [design](#), [plot.cpg.perm](#), [sort.cpg.perm](#), [sort.cpg](#)

Examples

Object of class `cpg`

Methods for object of class

Usage

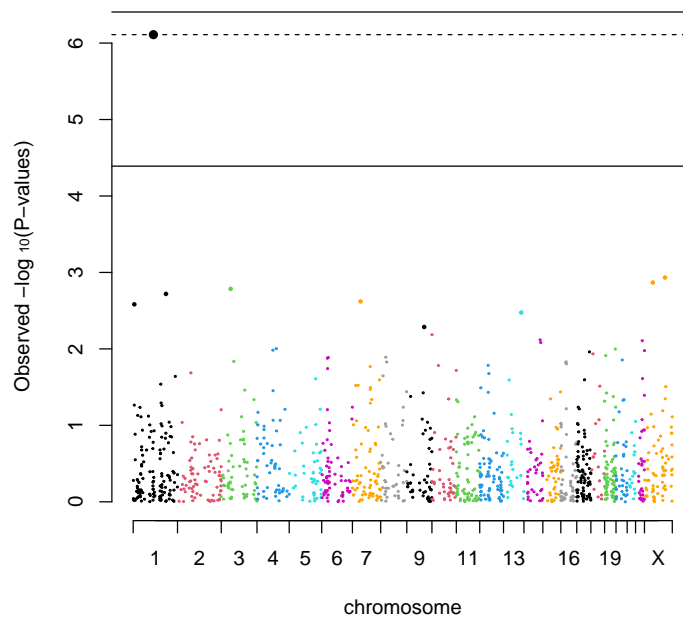
```
plot.cpg(x, save.plot = NULL, file.type="pdf", popup.pdf = FALSE,
tplot = FALSE, classic = TRUE, main.title = NULL, eps.size = c(5,
```

```

> #Doing a Manhattan plot. First load the data:
>
> #Doing a Manhattan plot. First load the data:
> library(CpGassoc)
> data(samplecpg,samplepheno,annotation,package="CpGassoc")
> ###NOTE: If you are dealing with large data, do not specify large.data=FALSE. The default
> ##This will involve partitioning up the data and performing more gc() to clear up space
> examplemanhat<-cpg.assoc(samplecpg,samplepheno$Disease,large.data=FALSE)
> manhattan(examplemanhat,annotation$TargetID,annotation$CHR,annotation$MAPINFO)
>

```

Manhattan Plot for association between methylation and Disease:



5), gc.p.val = FALSE, gcdisplay = FALSE, ...)

summary.cpg(object,...)

print.cpg(x,...)

sort.cpg(x,decreasing,...)

Arguments

x	Output of class <i>cpg</i> from <code>cpg.assoc</code> or <code>cpg.work</code> .
save.plot	Name of the file for the plot to be saved to. If not specified, plot will not be saved.
file.type	Type of file to be saved. Can either be "pdf" or "eps". Selecting <code>file.type="eps"</code> will result in publication quality editable postscript files that can be opened by Adobe Illustrator or Photoshop.
popup.pdf	TRUE or FALSE. If creating a pdf file, this indicates if the plot should appear in a popup window as well. If running in a cluster-like environment, best to leave FALSE.
tplot	Logical. If TRUE, ordered t-statistics will be plotted against their expected quantiles. If FALSE (default), $-\log(p)$ will be plotted. If <code>indep</code> is a class variable this option will be ignored.
classic	Logical. If TRUE, a classic qq-plot will be generated, with all p-values plotted against predicted values (including significant). If FALSE Holm-significant CpG sites will not be used to compute expected quantiles and will be plotted separately.
main.title	Main title to be put on the graph. If NULL one based on the analysis will be used.
eps.size	Vector indicating the size of .eps file (if creating one). Corresponds to the options <code>horizontal</code> and <code>height</code> in the <code>postscript</code> function.

<code>gc.p.val</code>	Logical. If <code>TRUE</code> , plot will use the genomic control adjusted p-values.
<code>gdisplay</code>	Logical. If <code>TRUE</code> , plot will display the genomic control value in the legend.
<code>object</code>	Output of class <code>cpg</code> from <code>cpg.assoc</code> or <code>cpg.work</code> .
<code>decreasing</code>	Logical. Should the sort be increasing or decreasing? Not available for partial sorting.
<code>...</code>	Arguments to be passed to methods, such as graphical parameters.

Description

Methods and extra functions for class `cpg`.

`plot.cpg` creates a QQ plot based on the association p-values or t-statistics from the function `cpg.assoc`.

Value

`sort.cpg` returns an item of class `cpg` that is sorted by p-value.

`summary.cpg` creates a qq-plot based on the data, and scatterplots or boxplots for the top sites.

Authurs

Barfield, R.; Kilaru, V.; Conneely, K.

Maintainer: R. Barfield: barfieldrichard8@gmail.com

Note

Plots with empirical confidence intervals based on permutation tests can be obtained from `cpg.perm`.

See [plot.cpg.perm](#) for more info

See Also

[cpg.perm](#), [cpg.work](#), [cpg.assoc.scatterplot](#), [cpg.combine](#), [manhattan](#), [plot.cpg.perm](#), [sort.cpg.perm](#), [cpg.qc](#)

Examples

Object of class `cpg.perm`

Methods for object of class `cpg.perm`

Usage

```
plot.cpg.perm(x, save.plot = NULL, file.type="pdf", popup.pdf = FALSE,
main.title = NULL, eps.size = c(5, 5), tplot = FALSE, perm.ci = TRUE,
classic = TRUE, gc.p.val = FALSE, gcdisplay = FALSE, ...)
summary.cpg.perm(object,...)
print.cpg.perm(x,...)
sort.cpg.perm(x,decreasing,...)
```

Description

Methods and extra functions for class `cpg.perm`. `plot.cpg.perm` creates a QQ plot based on the association p-values or t-statistics from the function `cpg.perm`.

Arguments

<code>x</code>	Output from <code>cpg.perm</code> . Of class <code>cpg.perm</code> .
<code>save.plot</code>	Name of the file for the plot to be saved to. If not specified, plot will not be saved.
<code>file.type</code>	Type of file to be saved. Can either be "pdf" or "eps". Selecting <code>file.type="eps"</code> will result in publication quality editable postscript files that can be opened by Adobe Illustrator or Photoshop.
<code>popup.pdf</code>	TRUE or FALSE. If creating a pdf file, this indicates if the plot should appear in a popup window as well. If running in a cluster-like environment, best to leave FALSE.

main.title	Main title to be put on the graph. If <code>NULL</code> one based on the analysis will be used
eps.size	Vector indicating the size of .eps file (if creating one). Corresponds to the options <code>horizontal</code> and <code>height</code> in the <code>postscript</code> function.
tplot	Logical. If <code>TRUE</code> , ordered t-statistics will be plotted against their expected quantiles. If <code>FALSE</code> (default), $-\log(p)$ will be plotted. If <code>indep</code> is a class variable this option will be ignored.
perm.ci	Logical. If <code>TRUE</code> , the confidence intervals computed will be from the permuted values, otherwise will be based on the theoretical values.
classic	Logical. If <code>TRUE</code> , a classic qq-plot will be generated, with all p-values plotted against predicted values (including significant). If <code>FALSE</code> Holm-significant CpG sites will not be used to compute expected quantiles and will be plotted separately.
gc.p.val	Logical. If <code>TRUE</code> , plot will use the genomic control adjusted p-values.
gdisplay	Logical. If <code>TRUE</code> , plot will display the genomic control value in the legend.
object	Output of class <code>cpg.perm</code> from <code>cpg.perm</code> .
decreasing	Logical. Should the sort be increasing or decreasing? Not available for partial sorting.
...	Arguments to be passed to methods, such as graphical parameters.

Authors

Barfield, R.; Kilaru, V.; Conneely, K.

Maintainer: R. Barfield: barfieldrichard8@gmail.com


```

> ##Using the results from the example given in cpg.assoc.
> ###NOTE: If you are dealing with large data, do not specify large.data=FALSE. The default
> ##This will involve partitioning up the data and performing more gc() to clear up space
> ##QQ Plot:
> library(CpGassoc)
> data(samplecpg,samplepheno,package="CpGassoc")
> test<-cpg.assoc(samplecpg,samplepheno$weight,data.frame(samplepheno$Distance,samplepheno
> plot(test)
> ##t-statistic plot:
> plot(test,tplot=TRUE)
> ##Now an example of sort
> head(sort(test)$results)

```

	CPG.Labels	T.statistic	P.value	Holm.sig	FDR	gc.p.value
694	CpG694	3.475160	0.0006002142	FALSE	0.3833341	0.0006002142
293	CpG293	3.464076	0.0006243226	FALSE	0.3833341	0.0006243226
560	CpG560	3.333678	0.0009848497	FALSE	0.4031318	0.0009848497
148	CpG148	3.187753	0.0016135434	FALSE	0.4953578	0.0016135434
238	CpG238	3.012760	0.0028504303	FALSE	0.5921086	0.0028504303
998	CpG998	-3.008091	0.0028930386	FALSE	0.5921086	0.0028930386

```

> ##Summary
> summary(test)

```

The top ten CpG sites were:

	CPG.Labels	T.statistic	P.value	Holm.sig	FDR	gc.p.value
694	CpG694	3.475160	0.0006002142	FALSE	0.3833341	0.0006002142
293	CpG293	3.464076	0.0006243226	FALSE	0.3833341	0.0006243226
560	CpG560	3.333678	0.0009848497	FALSE	0.4031318	0.0009848497
148	CpG148	3.187753	0.0016135434	FALSE	0.4953578	0.0016135434
238	CpG238	3.012760	0.0028504303	FALSE	0.5921086	0.0028504303
998	CpG998	-3.008091	0.0028930386	FALSE	0.5921086	0.0028930386
1059	CpG1059	-2.932014	0.0036749081	FALSE	0.6295151	0.0036749081
100	CpG100	2.889847	0.0041873059	FALSE	0.6295151	0.0041873059
1006	CpG1006	-2.831992	0.0049965867	FALSE	0.6295151	0.0049965867
1182	CpG1182	-2.823521	0.0051263442	FALSE	0.6295151	0.0051263442

To access results for all 1228 CpG sites use object\$results or sort(object)\$results to obtain results sorted by p-value.

General info:

	Min.P.Observed	Num.Cov	fdr.cutoff	FDR.method	Phenotype	chipinfo	num.Holm
1	0.0006002142	2	0.05	BH	weight	NULL	0
	num.fdr						
1	0						

0 sites were found significant by the Holm method
0 sites were found significant by BH method

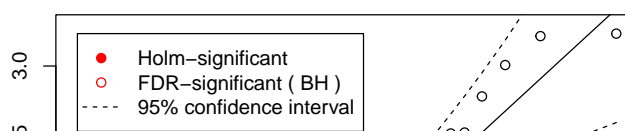
The beta values were taken from: samplecpg

Effect sizes and standard error can be accessed using \$coefficients

Other attributes are: results, Holm.sig, FDR.sig, info

They can be accessed using the \$

**QQ plot for association
between methylation and weight**



Note

Empirical confidence intervals will be computed only if there are a hundred or more permutations. Otherwise the theoretical confidence intervals will be plotted.

See Also

[cpg.assoc](#), [cpg.perm](#), [plot.cpg](#), [cpg.work](#), [scatterplot](#), [cpg.combine](#), [design](#), [manhattan](#), [sort.cpg](#)

Examples

<code>scatterplot</code>	<i>Plot beta values of individual CpG sites against the independent variable.</i>
--------------------------	---

Usage

```
scatterplot(x, cpg.rank = NULL, cpg.name = NULL, save.plot = NULL,  
file.type="pdf", eps.size = c(5, 5), popup.pdf = FALSE, beta.values =  
NULL,user.indep=NULL,main.title=NULL, ...)
```

Arguments

<code>x</code>	Object of class <i>cpg</i> or <i>cpg.perm</i> .
<code>cpg.rank</code>	A vector listing the rank of sites to be plotted. The rank is based on the ordered p-values.
<code>cpg.name</code>	A character vector containing the names of CpG sites to be plotted against the phenotype of interest. This option is ignored if <code>cpg.rank</code> is specified.
<code>save.plot</code>	Prefix of the filename for the plot(s) to be saved to. If specified, plot filenames will be created by appending this prefix to either <code>cpg.rank</code> or <code>cpg.name</code> . If not specified, plot will not be saved.
<code>file.type</code>	Type of file to be saved. Can either be "pdf" or "eps". Selecting <code>file.type="eps"</code> will result in publication quality editable postscript files that can be opened by Adobe Illustrator or Photoshop.

eps.size	Vector indicating the size of .eps file (if creating one). Corresponds to horizontal and height.
popup.pdf	TRUE or FALSE. If creating a pdf file, this indicates if the plot should appear in a popup window as well. If running in a cluster-like environment, best to leave FALSE.
beta.values	If the object has been renamed (i.e. <i>xinfo</i> beta <code>info</code> is no longer in <code>ls(.GlobalEnv)</code>) then specify the new object here.
user.indep	Default NULL. If return.data=F in run, scatterplot will not work. Pass in samplepheno here. Must be in same order as samplecpg.
main.title	Main title to be put on the graph. If NULL one based on the analysis will be used
...	Arguments to be passed to methods, such as graphical parameters.

Details

An unlimited number of CpG sites can be selected for plotting by specifying either `cpg.rank` or `cpg.name`, as shown in the Examples below. Note that only one of these options is needed; if both are entered, `cpg.rank` will be used.

Authors

Barfield, R.; Conneely, K.; Kilaru, V.

Maintainer: R. Barfield: barfieldrichard8@gmail.com

See Also

[cpg.assoc](#), [cpg.perm](#), [manhattan](#), [cpg.work](#), [plot.cpg.perm](#), [cpg.combine](#), [design](#), [plot.cpg](#), [sort.cpg.perm](#), [sort.cpg](#)

Examples

```

> library(CpGassoc)
> data(samplecpg,samplepheno,package="CpGassoc")
> ##We will do the analysis on a subset to save time
> ###NOTE: If you are dealing with large data, do not specify large.data=FALSE. The default
> ##This will involve partitioning up the data and performing more gc() to clear up space
> #The qq plot:
> Testperm<-cpg.perm(samplecpg,samplepheno$weight,data.frame(samplepheno$Dose,samplepheno$
+ seed=2314,nperm=10,large.data=FALSE)
> plot(Testperm)
> #The t-statistic plot from cpg.perm has confidence intervals since we were allowed to pe
> plot(Testperm,tplot=TRUE)
> #If there was 100 or more permutations, there would be emperical confidence intervals.
>
> ###Now for Sort
> head(sort(Testperm)$results)

```

	CPG.Labels	T.statistic	P.value	Holm.sig	FDR	gc.p.value
694	CpG694	3.475160	0.0006002142	FALSE	0.3833341	0.0006002142
293	CpG293	3.464076	0.0006243226	FALSE	0.3833341	0.0006243226
560	CpG560	3.333678	0.0009848497	FALSE	0.4031318	0.0009848497
148	CpG148	3.187753	0.0016135434	FALSE	0.4953578	0.0016135434
238	CpG238	3.012760	0.0028504303	FALSE	0.5921086	0.0028504303
998	CpG998	-3.008091	0.0028930386	FALSE	0.5921086	0.0028930386

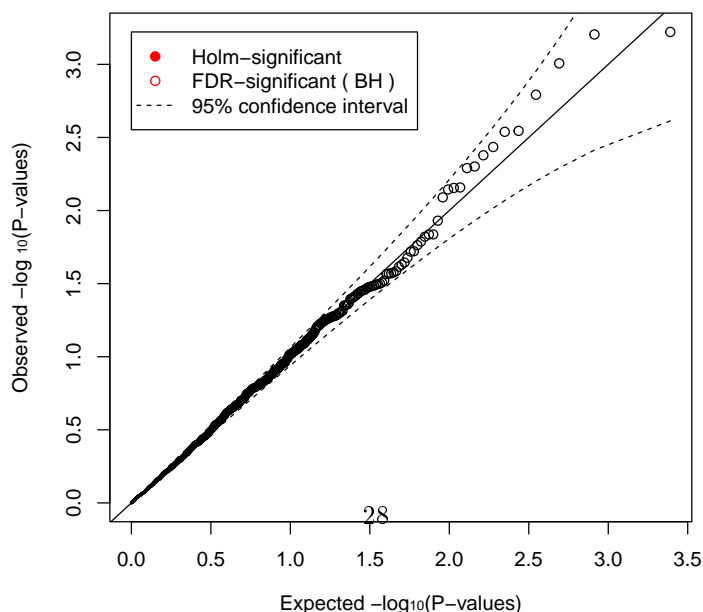
```

> head(Testperm$results)

```

	CPG.Labels	T.statistic	P.value	Holm.sig	FDR	gc.p.value
1	CpG1	-1.63736663	0.10279215	FALSE	0.9439499	0.10279215
2	CpG2	-0.09076561	0.92775038	FALSE	0.9927071	0.92775038
3	CpG3	-2.36081337	0.01899094	FALSE	0.9057057	0.01899094
4	CpG4	1.28326656	0.20056830	FALSE	0.9530109	0.20056830
5	CpG5	-1.29476076	0.19657851	FALSE	0.9530109	0.19657851
6	CpG6	-0.94975324	0.34314045	FALSE	0.9911946	0.34314045

**QQ plot for association
between methylation and weight**



```
> #Load the data:
> data(samplecpg,samplepheno,package="CpGassoc")
> library(CpGassoc)
> ###NOTE: If you are dealing with large data, do not specify large.data=FALSE. The default
> ##This will involve partitioning up the data and performing more gc() to clear up space
> test<-cpg.assoc(samplecpg,samplepheno$weight,large.data=FALSE)
> ##Using rank, will plot the top three sites in order of significance:
> scatterplot(test,cpg.rank=c(1:3),user.indep=samplepheno$weight)
```

Press enter to continue

Press enter to continue

Press enter to continue

All 3 sites plotted

```
> ##Using name, specify three sites:
> scatterplot(test,cpg.name=c("CpG1182","CpG1000","CpG42"),user.indep=samplepheno$weight)
```

Press enter to continue

Press enter to continue

Press enter to continue

All 3 sites plotted

```
> ##Plotting something that is categorical in nature:
> test2<-cpg.assoc(samplecpg,factor(samplepheno$Disease),large.data=FALSE)
> scatterplot(test2,c(2),user.indep=as.factor(samplepheno$Disease))
```

Press enter to continue

All 1 sites plotted