Package 'ChAMPdata'

November 10, 2026

Type Package

Title Data Packages for ChAMP package

Version 2.44.0

Date 2021-11-22

Author Yuan Tian [ctb,aut], Tiffany Morris [cre,aut], Lee Stirling [ctb] and Andrew Teschendorff [ctb]

Maintainer Yuan Tian <champ450k@gmail.com>

Description
Provides datasets needed for ChAMP including a test dataset and blood controls for CNA analysis.

License GPL-3

Depends GenomicRanges (>= 1.22.4),BiocGenerics(>= 0.16.1),R (>= 3.3)

biocViews ExperimentData

git_url https://git.bioconductor.org/packages/ChAMPdata

git_branch RELEASE_3_23

git_last_commit d6bcdab

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-10

Contents

ChAMPdata-package 2
Anno450K 3
AnnoEPIC 4
bloodCtl 4
CellTypeMeans27K 6
CellTypeMeans450K 7
ControlProbes450K 7
ControlProbesEPIC 8
EPIC.manifest.hg19 9
EPIC.manifest.pop.hg19 10
EPICSimData 11

hm450.manifest.hg19 12
hm450.manifest.pop.hg19 13
hprdAsigH 14
illumina450Gr 15
illuminaEPICGr 16
MatchGeneName 17
multi.hit 17
PathwayList 18
probe.features 19
probe.features.epic 20
probeInfoALL.epic.lv 21
probeInfoALL.lv 21
snp.hit 22
testDataSet 23

Index **24**

`ChAMPdata-package` *Data Packages to use with the ChAMP Chip Analysis Methylation Pipeline*

Description

This includes four data packages. ProbeInfoALL.lv includes annotation for the 450k array as required by the BMIQ normalization. probe.features includes probe annotations for the 450k array as included in the saved results files for the MVP and DMR functions. champBloodCtls provides reference control data for the champ.CNA function. testDataSet includes loaded and filtered (for detection) p-value of 6 arrays for the 450k array along with an accompanying samples sheet. This can be used to test the package. In addition, the raw IDAT files for these 6 arrays are available and can be accessed using system.file().

Details

Package:	ChAMPdata
Type:	Package
Version:	1.9.3
Date:	2016-03-22
License:	GPL-3

Three of the four packages are used internally by the ChAMP package. The testDataSet can be used to test the package.

Author(s)

Tiffany Morris, Cambridge Epigenetix; Yuan Tian, UCL Cancer Institute, Medical Genomics; Lee Stirling, UCL Cancer Institute, Medical Genomics; Andrew Teschendorff, UCL Cancer Institute, Statistical Genomics;

Maintainer: Yuan Tian and Tiffany Morris <champ450k@gmail.com>

Examples

```
data(probeInfoALL.lv)
data(probe.features)
data(testDataSet)
data(champBloodCtls)
data(EPICSimData)
data(illumina450Gr)
data(illuminaEPICGr)
data(probeInfoALL.epic.lv)
data(probe.features.epic)
data(CellTypeMeans27K)
data(CellTypeMeans450K)
```

Anno450K *Anno450K*

Description

HumanMethylation450K Loading Information, generated from HumanMethylation450_15017482_v1-2.csv.

Usage

```
data(Anno450K)
```

Format

Loading Information, probe and their address on chip.

`Annotation` Mapping between CpG ID and Address on Chip

`ControlProbe` Control Probe information, extracted from very origin Manifest.

Examples

```
data(Anno450K)
```

AnnoEPIC *AnnoEPIC*

Description

HumanMethylationEPIC Loading Information, generated from Infinium MethylationEPIC v1.0 B4 Manifest File (CSV Format) (https://support.illumina.com/array/array_kits/infinium-methylationepic-beadchip-kit/downloads.html).

Usage

```
data(AnnoEPIC)
```

Format

Loading Information, probe and their address on chip.

`Annotation` Mapping between CpG ID and Address on Chip

`ControlProbe` Control Probe information, extracted from very origin Manifest.

Examples

```
data(AnnoEPIC)
```

bloodCtl *Blood Control data*

Description

Blood control data for CNA analysis

Usage

```
data(bloodCtl)
```

Format

The format is: List of 6 $ mset :Formal class 'MethylSet' [package "minfi"] with 8 slots .. ..@ preprocessMethod : Named chr [1:3] "Raw (no normalization or bg correction)" "1.8.9" "0.4.0" .. .. ..- attr(*, "names")= chr [1:3] "rg.norm" "minfi" "manifest" .. ..@ assayData :<environment: 0x105193308> .. ..@ phenoData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. ..@ varMetadata :'data.frame': 10 obs. of 1 variable: .. ..$ labelDescription: chr [1:10] NA NA NA NA ... .@ data :'data.frame': 2 obs. of 10 variables: .. ..$ Sample_Name : chr [1:2] "blood_1" "blood_2" .. ..$ Sample_Plate: chr [1:2] "c" "c" .. ..$ Sample_Group: chr [1:2] "B" "B" .. ..$ Pool_ID : chr [1:2] "blood" "blood" .. ..$ Project : chr [1:2] "blood_pilot" "blood_pilot" .. ..$ Sample_Well : chr

[1:2] "F01" "H01" .. ..$ Array : chr [1:2] "R06C01" "R02C02" .. ..$ Slide : num [1:2] 9.31e+09 9.31e+09 .. ..$ Basename : chr [1:2] "/Users/regmtmo/Desktop/Sync/ACTIVE work/Lotte_450k_twinStudy_9May2013/bloodPilot_Sept2013/MORRIS Meth450K 280813/champBlood/930"| __truncated__ "/Users/regmtmo/Desktop/Sync/ACTIVE work/Lotte_450k_twinStudy_9May2013/bloodPilot_Sept2013/MORRIS Meth450K 280813/champBlood/930"| __truncated__ .. ..$ filenames : chr [1:2] "/Users/regmtmo/Desktop/Sync/ACTIVE work/Lotte_450k_twinStudy_9May2013/bloodPilot_Sept2013/MORRIS Meth450K 280813/champBlood/930"| __truncated__ "/Users/regmtmo/Desktop/Sync/ACTIVE work/Lotte_450k_twinStudy_9May2013/bloodPilot_Sept2013/MORRIS Meth450K 280813/champBlood/930"| __truncated__ .. ..@ dimLabels : chr [1:2] "sampleNames" "sampleColumns" .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 .. ..@ featureData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable: .. ..$ labelDescription: chr(0) .. ..@ data :'data.frame': 485512 obs. of 0 variables .. ..@ dimLabels : chr [1:2] "featureNames" "featureColumns" .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 .. ..@ experimentData :Formal class 'MIAME' [package "Biobase"] with 13 slots .. ..@ name : chr "" .. ..@ lab : chr "" .. ..@ contact : chr "" .. ..@ title : chr "" .. ..@ abstract : chr "" .. ..@ url : chr "" .. ..@ pubMedIds : chr "" .. ..@ samples : list() .. ..@ hybridizations : list() .. ..@ normControls : list() .. ..@ preprocessing : list() .. ..@ other : list() .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 2 .. ..$: int [1:3] 1 0 0 .. ..$: int [1:3] 1 1 0 .. ..@ annotation : Named chr [1:2] "IlluminaHumanMethylation450k" "ilmn12.hg19" .. .. ..- attr(*, "names")= chr [1:2] "array" "annotation" .. ..@ protocolData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable: .. ..$ labelDescription: chr(0) .. ..@ data :'data.frame': 2 obs. of 0 variables .. ..@ dimLabels : chr [1:2] "sampleNames" "sampleColumns" .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 4 .. ..$: int [1:3] 3 0 2 .. ..$: int [1:3] 2 22 0 .. ..$: int [1:3] 1 3 0 .. ..$: int [1:3] 1 0 0 $ rgSet :Formal class 'RGChannelSetExtended' [package "minfi"] with 7 slots .. ..@ assayData :<environment: 0x105310db8> .. ..@ phenoData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. ..@ varMetadata :'data.frame': 10 obs. of 1 variable: .. ..$ labelDescription: chr [1:10] NA NA NA NA ... .@ data :'data.frame': 2 obs. of 10 variables: .. ..$ Sample_Name : chr [1:2] "blood_1" "blood_2" .. ..$ Sample_Plate: chr [1:2] "c" "c" .. ..$ Sample_Group: chr [1:2] "B" "B" .. ..$ Pool_ID : chr [1:2] "blood" "blood" .. ..$ Project : chr [1:2] "blood_pilot" "blood_pilot" .. ..$ Sample_Well : chr [1:2] "F01" "H01" .. ..$ Array : chr [1:2] "R06C01" "R02C02" .. ..$ Slide : num [1:2] 9.31e+09 9.31e+09 .. ..$ Basename : chr [1:2] "/Users/regmtmo/Desktop/Sync/ACTIVE work/Lotte_450k_twinStudy_9May2013/bloodPilot_Sept2013/MORRIS Meth450K 280813/champBlood/930"| __truncated__ "/Users/regmtmo/Desktop/Sync/ACTIVE work/Lotte_450k_twinStudy_9May2013/bloodPilot_Sept2013/MORRIS Meth450K 280813/champBlood/930"| __truncated__ .. ..$ filenames : chr [1:2] "/Users/regmtmo/Desktop/Sync/ACTIVE work/Lotte_450k_twinStudy_9May2013/bloodPilot_Sept2013/MORRIS Meth450K 280813/champBlood/930"| __truncated__ "/Users/regmtmo/Desktop/Sync/ACTIVE work/Lotte_450k_twinStudy_9May2013/bloodPilot_Sept2013/MORRIS Meth450K 280813/champBlood/930"| __truncated__ .. ..@ dimLabels : chr [1:2] "sampleNames" "sampleColumns" .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 .. ..@ featureData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable: .. ..$ labelDescription: chr(0) .. ..@

data :'data.frame': 622399 obs. of 0 variables .. ..@ dimLabels : chr [1:2] "featureNames" "featureColumns" .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 .. ..@ experimentData :Formal class 'MIAME' [package "Biobase"] with 13 slots .. ..@ name : chr "" .. ..@ lab : chr "" .. ..@ contact : chr "" .. ..@ title : chr "" .. ..@ abstract : chr "" .. ..@ url : chr "" .. ..@ pubMedIds : chr "" .. ..@ samples : list() .. ..@ hybridizations : list() .. ..@ normControls : list() .. ..@ preprocessing : list() .. ..@ other : list() .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 2 .. ..$: int [1:3] 1 0 0 .. ..$: int [1:3] 1 1 0 .. ..@ annotation : Named chr [1:2] "IlluminaHumanMethylation450k" "ilmn12.hg19" .. .. ..- attr(*, "names")= chr [1:2] "array" "annotation" .. ..@ protocolData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots .. ..@ varMetadata :'data.frame': 0 obs. of 1 variable: .. ..$ labelDescription: chr(0) .. ..@ data :'data.frame': 2 obs. of 0 variables .. ..@ dimLabels : chr [1:2] "sampleNames" "sampleColumns" .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 1 .. ..$: int [1:3] 1 1 0 .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots .. ..@ .Data:List of 4 .. ..$: int [1:3] 3 0 2 .. ..$: int [1:3] 2 22 0 .. ..$: int [1:3] 1 3 0 .. ..$: int [1:3] 1 0 0 $ pd :'data.frame': 2 obs. of 10 variables: ..$ Sample_Name : chr [1:2] "blood_1" "blood_2" ..$ Sample_Plate: chr [1:2] "" "" ..$ Sample_Group: chr [1:2] "champCtls" "champCtls" ..$ Pool_ID : chr [1:2] "" "" ..$ Project : chr [1:2] "" "" ..$ Sample_Well : chr [1:2] "" "" ..$ Array : chr [1:2] "R06C01" "R02C02" ..$ Slide : num [1:2] 9.31e+09 9.31e+09 ..$ Basename : chr [1:2] "" "" ..$ filenames : chr [1:2] "" "" $ intensity: num [1:485512, 1:2] 12820 2714 1381 4083 3863 ... ..- attr(*, "dimnames")=List of 2 .. ..$: chr [1:485512] "cg00050873" "cg00212031" "cg00213748" "cg00214611" ... .$: chr [1:2] "blood_1" "blood_2" $ beta : num [1:485512, 1:2] 0.8648 0.0924 0.7846 0.0323 0.7118 ... ..- attr(*, "dimnames")=List of 2 .. ..$: chr [1:485512] "cg00050873" "cg00212031" "cg00213748" "cg00214611" ... .$: chr [1:2] "blood_1" "blood_2" $ detP : num [1:485512, 1:2] 0 0 0 0 0 0 0 0 0 0 ... ..- attr(*, "dimnames")=List of 2 .. ..$: chr [1:485512] "cg00050873" "cg00212031" "cg00213748" "cg00214611" ... .$: chr [1:2] "blood_1" "blood_2"

Examples

```
data(bloodCtl)
## maybe str(bloodCtl) ; plot(bloodCtl) ...
```

`CellTypeMeans27K` *Cell-type purified whole blood methylation CpG Sites for use with the RefBaseEWAS cell-type correction function.*

Description

This dataset contains 500 CpGs sites from cell type purified whole blood samples.

Usage

```
data(CellTypeMeans450K)
```

Format

The format is: num [1:500, 1:6] 0.901 0.9188 0.8503 0.8913 0.0436 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:500] "cg00226923" "cg16698623" "cg14102807" "cg14127336" ... ..$: chr [1:6] "CD8T" "CD4T" "NK" "Bcell" ...

Examples

```
data(CellTypeMeans27K)
```

CellTypeMeans450K *Cell-type purified whole blood methylation CpG Sites for use with the RefBaseEWAS cell-type correction function.*

Description

This dataset contains 600 CpGs sites from cell type purified whole blood samples.

Usage

```
data(CellTypeMeans450K)
```

Format

The format is: num [1:600, 1:6] 0.125 0.2 0.178 0.284 0.226 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:600] "cg25939861" "cg00219921" "cg08777095" "cg04329870" ... ..$: chr [1:6] "CD8T" "CD4T" "NK" "Bcell" ...

Examples

```
data(CellTypeMeans450K)
```

ControlProbes450K *ControlProbes450K*

Description

HumanMethylation450K Control Probes, generated from HumanMethylation450_15017482_v1-2.csv.

Usage

```
data(ControlProbes450K)
```

Format

A data frame with Control Probes used in champ.SVD function and their probe IDs in manifest HumanMethylation450_15017482_v1-2.csv.

`Name` The control name.

`Index` Probe ID in HumanMethylation450_15017482_v1-2.csv.

`RG` Color we extracted.

Examples

```
data(ControlProbes450K)
```

ControlProbesEPIC *ControlProbesEPIC*

Description

HumanMethylationEPIC Control Probes, generated from MethylationEPIC_v-1-0_B2.csv.

Usage

```
data(ControlProbesEPIC)
```

Format

A data frame with Control Probes used in champ.SVD function and their probe IDs in manifest MethylationEPIC_v-1-0_B2.csv.

`Name` The control name.

`Index` Probe ID in MethylationEPIC_v-1-0_B2.csv.

`RG` Color we extracted.

Examples

```
data(ControlProbesEPIC)
```

`EPIC.manifest.hg19` *SNP annotation for EPIC array.*

Description

This dataset is added to do filtering on SNP for EPIC array. Data is based on hg19.

Usage

```
data(EPIC.manifest.hg19)
```

Format

'data.frame': 866895 obs. of 41 variables: $ seqnames : Factor w/ 26 levels "chr1","chr2",..: 1 1 1 1 1 1 1 1 1 ... $ start : int 10525 10848 10850 15865 18827 29407 29425 36603 68849 68889 ... $ end : int 10526 10849 10851 15866 18828 29408 29426 36604 68850 68890 ... $ width : int 2 2 2 2 2 2 2 2 2 ... $ strand : Factor w/ 3 levels "+","-","*": 3 3 3 3 3 3 3 3 3 ... $ addressA : num 21611527 91693541 82663207 2665852 84794291 ... $ addressB : num NA 47784201 3701821 39757192 NA ... $ channel : chr "Both" "Grn" "Grn" "Red" ... $ designType : Factor w/ 2 levels "I","II": 2 1 1 1 2 1 1 2 2 2 ... $ nextBase : chr "G/A" "C" "C" "A" ... $ nextBaseRef : chr "C" "G" "G" "C" ... $ probeType : chr "cg" "cg" "cg" "cg" ... $ orientation : Factor w/ 2 levels "down","up": 1 2 2 1 1 1 1 2 1 1 ... $ probeCpGcnt : int 3 7 8 2 2 8 7 1 1 0 ... $ context35 : int 4 12 12 4 3 11 12 1 1 1 ... $ probeStart : int 10526 10800 10802 15865 18828 29407 29425 36554 68850 68890 ... $ probeEnd : num 10575 10849 10851 15914 18877 ... $ ProbeSeqA : chr "AAACRAAACTACRTTATCCTCTACACAAATTTCRATAATACTCTAAAAAC" "ACACATACTAACACATCAAAATAAAAACATAACACAAACACAAAAAAACA" "ACATACTAACACATCAAAATAAAAACATAACACAAACACAAAAAAACACA" "CCAATAACTAACCACTCTACTAAAATCCATCCACCAAACTAAAAACATCA" ... $ ProbeSeqB : chr "" "ACACATACTAACGCGTCGAAATAAAAACGTAACGCAAACGCAAAAAAACG" "ACATACTAACGCGTCGAAATAAAAACGTAACGCAAACGCAAAAAAACGCG" "CCGATAACTAACCACTCTACTAAAATCCATCCGCCAAACTAAAAACATCG" ... $ chrmA : chr "chr1" "chr1" "chr1" "chr1" ... $ begA : int 10526 10800 10802 15865 18828 29407 29425 36554 68850 68890 ... $ flag.A : int 16 0 0 16 16 16 16 0 16 16 ... $ mapQ.A : int 0 22 22 17 1 3 3 0 6 6 ... $ cigarA : chr "50M" "50M" "50M" "50M" ... $ chrmB : chr "." "chr1" "chr1" "chr1" ... $ begB : int NA 10800 10802 15865 NA 29407 29425 NA NA NA ... $ flag.B : int NA 0 0 16 NA 16 16 NA NA NA ... $ mapQ.B : int NA 22 22 17 NA 3 3 NA NA NA ... $ cigarB : chr "." "50M" "50M" "50M" ... $ posMatch : logi NA NA NA NA NA NA ... $ MASK.mapping : logi TRUE FALSE FALSE FALSE TRUE TRUE ... $ MASK.typeINextBaseSwitch: logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.rmsk15 : logi TRUE TRUE TRUE FALSE FALSE FALSE ... $ MASK.sub35.copy : logi TRUE FALSE TRUE TRUE FALSE TRUE ... $ MASK.sub30.copy : logi TRUE TRUE TRUE TRUE FALSE TRUE ... $ MASK.sub25.copy : logi TRUE TRUE TRUE TRUE FALSE TRUE ... $ MASK.sub40.copy : logi TRUE FALSE FALSE TRUE FALSE FALSE ... $ MASK.snp5.common : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.snp5.GMAF1p : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.extBase : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general : logi TRUE TRUE TRUE TRUE TRUE TRUE ...

Examples

```
data(EPIC.manifest.hg19)
```

`EPIC.manifest.pop.hg19` *SNP annotation for EPIC array for different population.*

Description

This dataset is added to do filtering on SNP for EPIC array. Since different population have different SNP probes, this file is included to provide a comprehensive filtering annotation.

Usage

```
data(EPIC.manifest.pop.hg19)
```

Format

'data.frame': 866895 obs. of 67 variables: $ seqnames : Factor w/ 26 levels "chr1","chr2",..: 1 1 1 1 1 1 1 1 1 ... $ start : int 10525 10848 10850 15865 18827 29407 29425 68849 68889 69591 ... $ end : int 10526 10849 10851 15866 18828 29408 29426 68850 68890 69592 ... $ width : int 2 2 2 2 2 2 2 2 2 ... $ strand : Factor w/ 3 levels "+","-","*": 3 3 3 3 3 3 3 3 3 ... $ MASK.general.AFR: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.AFR : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.EAS: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.EAS : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.EUR: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.EUR : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.SAS: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.SAS : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.AMR: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.AMR : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.GWD: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.GWD : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.YRI: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.YRI : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.TSI: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.TSI : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.IBS: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.IBS : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.CHS: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.CHS : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.PUR: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.PUR : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.JPT: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.JPT : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.GIH: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.GIH : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.CHB: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.CHB : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.STU: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.STU : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.ITU: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.ITU : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.LWK: logi TRUE TRUE TRUE TRUE TRUE TRUE ...

$ MASK.snp5.LWK : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.KHV: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.KHV : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.FIN: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.FIN : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.ESN: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.ESN : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.CEU: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.CEU : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.PJL: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.PJL : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.ACB: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.ACB : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.CLM: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.CLM : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.CDX: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.CDX : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.GBR: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.GBR : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.BEB: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.BEB : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.PEL: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.PEL : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.MSL: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.MSL : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.MXL: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.MXL : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.ASW: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.ASW : logi FALSE FALSE FALSE FALSE FALSE FALSE ...

Examples

```
data(EPIC.manifest.pop.hg19)
```

`EPICSimData` *Simulation EPIC beadarray Dataset.*

Description

This dataset is available to test ChAMP functions on EPIC array.

Usage

```
data(EPICSimData)
```

Format

List of 3 $ beta: num [1:780385, 1:16] 0.7382 0.8572 0.8624 0.4492 0.0751 ... ..- attr(*, "dimnames")=List of 2 .. ..$: chr [1:780385] "cg14817997" "cg16269199" "cg13869341" "cg02404219" ... .$: chr [1:16] "HELA_Sim_1" "HELA_Sim_2" "HELA_Sim_3" "HELA_Sim_4" ... $ pd :'data.frame': 16 obs. of 10 variables: ..$ Sample_Name : chr [1:16] "HELA_Sim_1" "HELA_Sim_2" "HELA_Sim_3" "HELA_Sim_4" ... ..$ Sample_Plate: chr [1:16] "EPIC2" "EPIC2" "EPIC2" "EPIC2" ... ..$ Sample_Group: chr [1:16] "control" "control" "control" "control" ... ..$ Pool_ID : logi [1:16] NA NA NA NA NA NA ... ..$ Project : chr [1:16] "X" "X" "X" "X" ... ..$ Sample_Well : chr [1:16] "A01" "B01" "C01" "D01" ... ..$ Array : chr [1:16] "R01C01" "R02C01"

"R03C01" "R04C01" ... ..$ Slide : chr [1:16] "200079600019" "200079600019" "200079600019" "200079600019" ... ..$ Basename : chr [1:16] "character(0)" "character(0)" "character(0)" "character(0)" ... ..$ filenames : chr [1:16] "./Demo/200079600019_R01C01" "./Demo/200079600019_R02C01" "./Demo/200079600019_R03C01" "./Demo/200079600019_R04C01" ... $ detP: num [1:780385, 1:16] 0 0 0 0 0 0 0 0 0 0 ... ..- attr(*, "dimnames")=List of 2 .. ..$: chr [1:780385] "cg18478105" "cg14361672" "cg01763666" "cg02115394" ... .$: NULL

Examples

```
data(EPICSimData)
```

`hm450.manifest.hg19` *SNP annotation for 450K array.*

Description

This dataset is added to do filtering on SNP for 450K array. Data is based on hg19

Usage

```
data(hm450.manifest.hg19)
```

Format

'data.frame': 485577 obs. of 41 variables: $ seqnames : Factor w/ 26 levels "chr1","chr2",..: 1 1 1 1 1 1 1 1 1 ... $ start : int 15865 18827 29407 29425 29435 68849 69591 91550 135252 370260 ... $ end : int 15866 18828 29408 29426 29436 68850 69592 91551 135253 370261 ... $ width : int 2 2 2 2 2 2 2 2 2 ... $ strand : Factor w/ 3 levels "+","-","*": 3 3 3 3 3 3 3 3 3 3 ... $ addressA : num 62703328 27651330 25703424 61731400 26752380 ... $ addressB : num 16661461 NA 34666387 14693326 50693408 ... $ channel : chr "Red" "Both" "Red" "Red" ... $ designType : Factor w/ 2 levels "I","II": 1 2 1 1 1 2 2 2 2 2 ... $ nextBase : chr "A" "G/A" "A" "T" ... $ nextBaseRef : chr "C" "C" "C" "A" ... $ probeType : chr "cg" "cg" "cg" "cg" ... $ orientation : Factor w/ 2 levels "down","up": 1 1 1 1 1 1 2 2 2 2 ... $ probeCpGcnt : int 2 2 8 7 7 1 3 1 4 3 ... $ context35 : int 4 3 11 12 12 1 2 2 4 5 ... $ probeStart : int 15865 18828 29407 29425 29435 68850 69542 91501 135203 370211 ... $ probeEnd : num 15914 18877 29456 29474 29484 ... $ ProbeSeqA : chr "CCAATAACTAACCACTCTACTAAAATCCATCCACCAAACTAAAAACATCA" "ACTCRAAATTTACTCAATAAACCRTTCAATATATACAAAAACAATTCCCC" "AAAAAAAACACAATAAAAAACAAACAACAACATTAAAACCCAAAAACACA" "AATCCTAAAACCACACTCAAAAAAAACACAATAAAAAACAAACAACAACA" ... $ ProbeSeqB : chr "CCGATAACTAACCACTCTACTAAAATCCATCCGCCAAACTAAAAACATCG" "" "GAAAAAAACGCAATAAAAAACGAACGACGACGTTAAAACCCGAAAACGCG" "AATCCTAAAACCGCGCTCGAAAAAAACGCAATAAAAAACGAACGACGACG" ... $ chrmA : chr "chr1" "chr1" "chr1" "chr1" ... $ begA : int 15865 18828 29407 29425 29435 68850 69542 91501 135203 370211 ... $ flag.A : int 16 16 16 16 16 16 0 0 0 0 ... $ mapQ.A : int 17 1 3 3 3 6 60 6 0 3 ... $ cigarA : chr "50M" "50M" "50M" "50M" ... $ chrmB : chr "chr1" "." "chr1" "chr1" ... $ begB : int 15865 NA 29407 29425 29435 NA NA NA NA NA ... $ flag.B : int 16 NA 16 16 16 NA NA NA NA NA ... $ mapQ.B : int 17 NA 3 3 3 NA NA NA NA NA ... $ cigarB : chr "50M" "." "50M" "50M" ...

$ posMatch : logi NA NA NA NA NA NA ... $ MASK.mapping : logi FALSE TRUE TRUE TRUE TRUE TRUE ... $ MASK.typeINextBaseSwitch: logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.rmsk15 : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.sub35.copy : logi TRUE FALSE TRUE FALSE FALSE TRUE ... $ MASK.sub30.copy : logi TRUE FALSE TRUE FALSE FALSE TRUE ... $ MASK.sub25.copy : logi TRUE FALSE TRUE FALSE TRUE TRUE ... $ MASK.sub40.copy : logi TRUE FALSE FALSE FALSE FALSE TRUE ... $ MASK.snp5.common : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.snp5.GMAF1p : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.extBase : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general : logi TRUE TRUE TRUE TRUE TRUE TRUE ...

Examples

```
data(hm450.manifest.hg19)
```

`hm450.manifest.pop.hg19`

SNP annotation for 450K array for different population. Data is based on hg19.

Description

This dataset is added to do filtering on SNP for 450K array. Since different population have different SNP probes, this file is included to provide a comprehensive filtering annotation.

Usage

```
data(hm450.manifest.pop.hg19)
```

Format

'data.frame': 866895 obs. of 67 variables: $ seqnames : Factor w/ 26 levels "chr1","chr2",..: 1 1 1 1 1 1 1 1 1 ... $ start : int 10525 10848 10850 15865 18827 29407 29425 68849 68889 69591 ... $ end : int 10526 10849 10851 15866 18828 29408 29426 68850 68890 69592 ... $ width : int 2 2 2 2 2 2 2 2 2 ... $ strand : Factor w/ 3 levels "+","-","*": 3 3 3 3 3 3 3 3 3 ... $ MASK.general.AFR: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.AFR : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.EAS: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.EAS : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.EUR: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.EUR : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.SAS: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.SAS : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.AMR: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.AMR : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.GWD: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.GWD : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.YRI: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.YRI : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.TSI: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.TSI : logi FALSE FALSE FALSE

FALSE FALSE FALSE ... $ MASK.general.IBS: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.IBS : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.CHS: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.CHS : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.PUR: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.PUR : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.JPT: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.JPT : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.GIH: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.GIH : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.CHB: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.CHB : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.STU: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.STU : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.ITU: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.ITU : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.LWK: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.LWK : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.KHV: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.KHV : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.FIN: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.FIN : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.ESN: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.ESN : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.CEU: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.CEU : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.PJL: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.PJL : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.ACB: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.ACB : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.CLM: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.CLM : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.CDX: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.CDX : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.GBR: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.GBR : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.BEB: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.BEB : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.PEL: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.PEL : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.MSL: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.MSL : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.MXL: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.MXL : logi FALSE FALSE FALSE FALSE FALSE FALSE ... $ MASK.general.ASW: logi TRUE TRUE TRUE TRUE TRUE TRUE ... $ MASK.snp5.ASW : logi FALSE FALSE FALSE FALSE FALSE FALSE ...

Examples

```
data(hm450.manifest.pop.hg19)
```

`hprdAsigH` *hprdAsigH PPI network dataset for champ.FEM function.*

Description

Dataset would be used in champ.FEM, the adjacent matrix.

Usage

```
data(hprdAsigH)
```

Examples

```
data(hprdAsigH)
```

`illumina450Gr` *GRange Struct of illumina 450K array.*

Description

This dataset will be used in champ.lasso functions.

Usage

```
data(illumina450Gr)
```

Format

..@ seqnames :Formal class 'Rle' [package "S4Vectors"] with 4 slots .. .. ..@ values : Factor w/ 24 levels "chr16","chr3",..: 1 2 3 4 5 1 4 3 6 7 ... .@ lengths : int [1:446018] 1 2 1 1 1 1 1 1 1 1 ... .@ elementMetadata: NULL .. .. ..@ metadata : list() ..@ ranges :Formal class 'IRanges' [package "IRanges"] with 6 slots .. .. ..@ start : int [1:485512] 53468112 37459206 171916037 91194674 42263294 69341139 28890100 41167802 230560793 23034447 ... .@ width : int [1:485512] 1 1 1 1 1 1 1 1 1 1 ... .@ NAMES : chr [1:485512] "cg00000029" "cg00000108" "cg00000109" "cg00000165" ... .@ elementType : chr "integer" .. .. ..@ elementMetadata: NULL .. .. ..@ metadata : list() ..@ strand :Formal class 'Rle' [package "S4Vectors"] with 4 slots .. .. ..@ values : Factor w/ 3 levels "+","-","*": 1 2 1 2 1 2 1 2 1 2 ... .@ lengths : int [1:242276] 3 2 2 1 2 1 1 1 3 2 ... .@ elementMetadata: NULL .. .. ..@ metadata : list() ..@ elementMetadata:Formal class 'DataFrame' [package "S4Vectors"] with 6 slots .. .. ..@ rownames : NULL .. .. ..@ nrows : int 485512 .. .. ..@ listData :List of 27 .. ..$ chrom_hg18 : Factor w/ 24 levels "chr1","chr10",..: 8 16 16 1 21 6 8 21 1 7 ... .$ MAPINFO_hg18 : int [1:485512] 52025613 37434210 173398731 90967262 42382451 68410892 28797601 41286959 228627416 20585888 ... .$ chrom_hg38 : Factor w/ 37 levels "chr1","chr1_KI270706v1_random",..: 13 26 26 1 32 8 13 32 1 11 ... .$ MAPINFO_hg38 : int [1:485512] 53434200 37417715 172198247 90729117 42405776 68874422 28878779 41310283 230425047 22838619 ... .$ arm : Factor w/ 2 levels "p","q": 2 1 2 1 1 2 1 1 2 2 ... .$ probeStrand : chr [1:485512] "+" "+" "+" "-" ... .$ ensemblID : chr [1:485512] "ENSG00000103479" "ENSG00000198590" "ENSG00000075420" "ENSG00000143032" ... .$ geneSymbol : chr [1:485512] "RBL2" "C3orf35" "FNDC3B" "BARHL2" ... .$ geneStrand : Factor w/ 48 levels "-","-;-","-;-;-",..: 24 24 24 1 24 1 24 1 1 1 ... .$ relationtoGene : Factor w/ 21 levels "downstream","inside",..: 3 2 2 3 2 2 2 3 2 3 ... .$ distancetoTSS : chr [1:485512] "-250" "31445" "158618" "-11880" ... .$ nearestGeneBoundary: chr [1:485512] "250" "17782" "158618" "11880" ... .$ feature : Factor w/ 7 levels "1stExon","3'UTR",..: 6 4 4 5 2 2 1 6 5 7 ... .$ cgi : Factor w/ 4 levels "island","open sea",..: 4 2 2 4 2 4 4 4 4 1 ... .$ cgiName : chr [1:485512] NA NA NA NA ... .$ featureCgi : Factor w/ 28 levels

"1stExon - island",..: 24 14 14 20 6 8 4 24 20 25 ... .$ conservedTfsb : Factor w/ 815 levels "V$AHR_01","V$AHR_01;V$EGR1_01;V$EGR2_01;V$EGR3_01;V$NGFIC_01",..: NA NA NA 578 NA NA NA NA NA NA ... .$ bwaMultiMap : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 ... .$ promoterAssociated : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 2 2 1 2 ... .$ asnCN : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ... .$ afrCN : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ... .$ amrCN : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ... .$ eurCN : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ... .$ asnLast4 : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ... .$ afrLast4 : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ... .$ amrLast4 : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ... .$ eurLast4 : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ... .@ elementType : chr "ANY" .. .. ..@ elementMetadata: NULL .. .. ..@ metadata : list() ..@ seqinfo :Formal class 'Seqinfo' [package "GenomeInfoDb"] with 4 slots .. .. ..@ seqnames : chr [1:24] "chr16" "chr3" "chr1" "chr8" ... .@ seqlengths : int [1:24] NA NA NA NA NA NA NA NA NA NA ... .@ is_circular: logi [1:24] NA NA NA NA NA NA ... .@ genome : chr [1:24] NA NA NA NA ... ..@ metadata : list()

Examples

```
data(illumina450Gr)
```

`illuminaEPICGr` *GRange S4 Object of illumina EPIC array.*

Description

This dataset will be used in champ.lasso functions.

Usage

```
data(illuminaEPICGr)
```

Format

..@ seqnames :Formal class 'Rle' [package "S4Vectors"] with 4 slots .. .. ..@ values : Factor w/ 24 levels "chr19","chr20",..: 1 2 3 4 5 6 7 8 3 9 ... .@ lengths : int [1:822229] 1 1 1 1 1 1 1 1 1 1 ... .@ elementMetadata: NULL .. .. ..@ metadata : list() ..@ ranges :Formal class 'IRanges' [package "IRanges"] with 6 slots .. .. ..@ start : int [1:866836] 5236016 61847650 6841125 198303466 24072640 93581139 57865112 15248173 144921929 131463936 ... .@ width : int [1:866836] 2 2 2 2 2 2 2 2 2 2 ... .@ NAMES : chr [1:866836] "cg07881041" "cg18478105" "cg23229610" "cg03513874" ... .@ elementType : chr "integer" .. .. ..@ elementMetadata: NULL .. .. ..@ metadata : list() ..@ strand :Formal class 'Rle' [package "S4Vectors"] with 4 slots .. .. ..@ values : Factor w/ 3 levels "+","-","*": 2 1 2 1 2 1 2 1 2 1 ... .@ lengths : int [1:433990] 3 1 1 2 2 3 1 5 1 1 ... .@ elementMetadata: NULL .. .. ..@ metadata : list() ..@ elementMetadata:Formal class 'DataFrame' [package "S4Vectors"] with 6 slots .. .. ..@ rownames : NULL .. .. ..@ nrows : int 866836 .. .. ..@ listData : Named list() .. .. ..@ elementType : chr "ANY" .. .. ..@ elementMetadata: NULL .. .. ..@ metadata : list() ..@ seqinfo :Formal class 'Seqinfo' [package "GenomeInfoDb"] with 4 slots .. .. ..@ seqnames : chr [1:24] "chr19" "chr20" "chr1" "chr2" ... .@ seqlengths : int [1:24] NA NA NA NA NA NA NA NA NA NA ... .@ is_circular: logi [1:24] NA NA NA NA NA NA ... .@ genome : chr [1:24] NA NA NA NA ... ..@ metadata : list()

Examples

```
data(illuminaEPICGr)
```

`MatchGeneName` *MatchGeneName*

Description

Two name list used in MAP.GUI functioni in ChAMP.

Usage

```
data(MatchGeneName)
```

Examples

```
data(MatchGeneName)
```

`multi.hit` *multi.hit information.*

Description

This dataset contains 9341 multi hit CpGs.

Usage

```
data(multi.hit)
```

Format

The format is: 'data.frame': 9341 obs. of 38 variables: $ TargetID : Factor w/ 485577 levels "cg00000029","cg00000108",..: 30 92 234 246 294 325 358 374 402 418 ... $ INFINIUM_DESIGN_TYPE : Factor w/ 2 levels "I","II": 1 2 1 1 2 1 2 2 2 1 ... $ GENOME_BUILD : int 37 37 37 37 37 37 37 37 37 37 ... $ CHR : Factor w/ 25 levels "","1","10","11",..: 12 8 9 7 2 21 20 5 20 2 ... $ MAPINFO : int 54746945 82925333 33318910 81580765 149370974 74508806 58777304 54500832 7339455 149719536 ... $ CHROMOSOME_36 : Factor w/ 26 levels "","1","10","11",..: 12 8 9 7 2 21 20 5 20 2 ... $ COORDINATE_36 : Factor w/ 484739 levels "","100000057",..: 367785 440537 259149 440391 109936 419166 365930 343920 414240 110329 ... $ STRAND : Factor w/ 3 levels "","F","R": 3 2 3 2 3 3 3 2 2 3 ... $ UCSC_REFGENE_NAME : Factor w/ 42456 levels "","A1BG","A1CF;A1CF;A1CF",..: 19768 1 1 38813 13130 15618 1 13453 6037 1 ... $ UCSC_REFGENE_ACCESSION : Factor w/ 52670 levels "","NM_000014",..: 35153 1 1 598 51448 2129 1 50622 15082 1 ... $ UCSC_REFGENE_GROUP : Factor w/ 4165 levels "","1stExon","1stExon;1stExon",..: 2970 1 1 2035 2035 943 1 2035 2447 1 ... $ UCSC_CPG_ISLANDS_NAME : Factor w/ 27177 levels "","chr1:10003165-10003585",..: 1 8132 8895 1 1 23921 1 5852 1 400 ... $ RELATION_TO_UCSC_CPG_ISLAND: Factor w/ 6 levels "","Island","N_Shelf",..: 1 6 2 1 1

2 1 2 1 2 ... $ SNP : Factor w/ 139064 levels "","rs10000296",..: 88450 1 93755 1 94630 80342 79504 1 1 89545 ... $ SNP_DISTANCE : Factor w/ 11304 levels "","-1","-1;0",..: 10743 1 2655 1 3352 1938 10243 1 1 5664 ... $ snp.hit : int 0 0 0 0 1 0 0 0 0 0 ... $ bwa.multi.hit : int 1 1 1 1 1 1 1 1 1 1 ... $ analyzed : int 0 0 0 0 0 0 0 0 0 0 ... $ DNase : int 0 0 0 0 0 1 0 0 0 0 ... $ h3k4me3 : int 0 0 0 0 0 1 0 0 0 1 ... $ h3k36me3 : int 0 0 0 0 0 0 0 0 0 0 ... $ h3k27ac : int 0 0 0 0 0 1 1 0 0 0 ... $ h3k4me1 : int 0 0 0 0 0 0 0 0 0 0 ... $ bivalent : int 0 0 0 0 0 0 0 0 0 1 ... $ h3k27me3 : int 0 0 0 0 0 0 0 1 0 1 ... $ h3k9me3 : int 0 0 0 0 0 0 0 0 0 0 ... $ dmc.constitutive : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.relapse : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.T : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.MLL : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.dic920 : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.HeH : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.t119 : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.t1221 : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.t922 : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.iamp : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.undefined : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.nonrecurrent : int NA NA NA NA NA NA NA NA NA NA ...

Examples

```
data(multi.hit)
```

`PathwayList` *PathwayList, used in champ.GSEA()*

Description

PathwayList from MsigDB,version 4.28. Contains all pathways would be checked in GSEA process.

Usage

```
data(PathwayList)
```

Format

A list contains 8567 gene list, each represent on Pathway or Gene Module. List of 8567 $ chr5q23 : chr [1:84] "PTMAP2" "FTMT" "PRR16" "MGC32805" ... $ chr16q24 : chr [1:123] "LOC642452" "SNORD68" "FLJ12547" "LOC729942" ... $ chr8q24 : chr [1:214] "KIFC2" "LYPD2" "EIF2C2" "ADCY8" ... $ chr13q11 : chr [1:22] "LOC645626" "SNX19P2" "SKA3" "ZMYM2" ... $ chr7p21 : chr [1:57] "CRS" "RPL36AP26" "RAD17P1" "TAS2R2" ... $ chr10q23 : chr [1:134] "SNCG" "LOC389997" "FAM190B" "HPSE2" ... ...

Examples

```
data(PathwayList)
```

probe.features *probe.features*

Description

HumanMethylation450 probe annotations

Usage

```
data(probe.features)
```

Format

A data frame with 485577 observations on the following 9 variables.

`CHR` a factor with levels

`MAPINFO` a numeric vector

`Strand` Strand of CpG sites.

`Type` type I&II probes.

`gene` Probe related gene.

`feature` a factor with levels `1stExon` `3'UTR}` \code{`5'UTR` `Body` `IGR` `TSS1500` `TSS200`

`cgi` a factor with levels `island` `open sea` `shelf` `shore`

`feat.cgi` a factor with levels `1stExon - island` `1stExon - open sea` `1stExon - shelf` `1stExon - shore` `3'UTR - island}` \code{`3'UTR - open sea` `3'UTR - shelf}` \code{`3'UTR - shore` `5'UTR - island}` \code{`5'UTR - open sea` `5'UTR - shelf}` \code{`5'UTR - shore` `Body - island` `Body - open sea` `Body - shelf` `Body - shore` `IGR - island` `IGR - open sea` `IGR - shelf` `IGR - shore` `TSS1500 - island` `TSS1500 - open sea` `TSS1500 - shelf` `TSS1500 - shore` `TSS200 - island` `TSS200 - open sea` `TSS200 - shelf` `TSS200 - shore`

`UCSC_CpG_Islands_Name` Island information.

`DHS` DHS information.

`Enhancer` Enhancer information.

`Phantom` Phantom information.

`Probe_SNPs` If SNP exist next to CpG Sites.

`Probe_SNPs_10` If SNP exist 10 bp to CpG sites.

Examples

```
data(probe.features)
```

probe.features.epic *probe.features.epic*

Description

HumanMethylationEPIC probe annotations

Usage

```
data(probe.features.epic)
```

Format

A data frame with 868565 observations on the following 14 variables.

`CHR` a factor with levels

`MAPINFO` a numeric vector

`Strand` Strand of CpG sites.

`Type` type I&II probes.

`gene` Probe related gene.

`feature` a factor with levels `1stExon` `3'UTR}` \code{`5'UTR` `Body` `IGR` `TSS1500` `TSS200`

`cgi` a factor with levels `island` `open sea` `shelf` `shore`

`feat.cgi` a factor with levels `1stExon - island` `1stExon - open sea` `1stExon - shelf` `1stExon - shore` `3'UTR - island}` \code{`3'UTR - open sea` `3'UTR - shelf}` \code{`3'UTR - shore` `5'UTR - island}` \code{`5'UTR - open sea` `5'UTR - shelf}` \code{`5'UTR - shore` `Body - island` `Body - open sea` `Body - shelf` `Body - shore` `IGR - island` `IGR - open sea` `IGR - shelf` `IGR - shore` `TSS1500 - island` `TSS1500 - open sea` `TSS1500 - shelf` `TSS1500 - shore` `TSS200 - island` `TSS200 - open sea` `TSS200 - shelf` `TSS200 - shore`

`UCSC_CpG_Islands_Name` Island information.

`SNP_ID` ID of SNP related to the probe.

`SNP_DISTANCE` The distance between probe and the SNP site.

Examples

```
data(probe.features.epic)
```

probeInfoALL.epic.lv *Probe Info Data for use with the BMIQ normalization.*

Description

The probe details are formatted here for the BMIQ function.

Usage

```
data(probeInfoALL.epic.lv)
```

Format

The format is: List of 3 $ Design : num [1:867531] 2 1 2 2 1 2 2 2 2 1 ... $ CGI : num [1:867531] 0 1 0 0 1 1 0 0 0 0 ... $ probeID: chr [1:867531] "cg07881041" "cg18478105" "cg23229610" "cg03513874" ...

Examples

```
data(probeInfoALL.epic.lv)
```

probeInfoALL.lv *Probe Info Data for use with the BMIQ normalization.*

Description

The probe details are formatted here for the BMIQ function.

Usage

```
data(probeInfoALL.lv)
```

Format

The format is: List of 5 $ typeC : num [1:485577] 1 1 1 1 1 1 1 1 1 1 ... $ Design : num [1:485577] 2 2 2 2 2 2 2 2 2 1 ... $ GeneGroup: int [1:485577] 1 NA 5 NA 6 6 4 1 NA 2 ... $ CGI : num [1:485577] 1 0 0 1 0 1 1 1 1 1 ... $ probeID : chr [1:485577] "cg00000029" "cg00000108" "cg00000109" "cg00000165" ...

Examples

```
data(probeInfoALL.lv)
```

snp.hit *CpG site related to SNP sites information.*

Description

This dataset contains snp sites located into SNP related sites.

Usage

```
data(snp.hit)
```

Format

The format is: 'data.frame': 29481 obs. of 38 variables: $ TargetID : Factor w/ 485577 levels "cg00000029","cg00000108",..: 16 21 34 65 76 98 119 120 147 160 ... $ INFINIUM_DESIGN_TYPE : Factor w/ 2 levels "I","II": 2 1 2 1 2 2 2 2 2 2 ... $ GENOME_BUILD : int 37 37 37 37 37 37 37 37 37 37 ... $ CHR : Factor w/ 25 levels "","1","10","11",..: 13 2 2 10 2 13 18 17 3 8 ... $ MAPINFO : int 23913414 5937253 170490434 57839538 8739981 237027592 178364146 110612894 112068221 90066806 ... $ CHROMOSOME_36 : Factor w/ 26 levels "","1","10","11",..: 13 2 2 10 2 13 18 17 3 8 ... $ COORDINATE_36 : Factor w/ 484739 levels "","100000057",..: 206129 365300 143074 353467 452543 205188 156567 31100 30980 456465 ... $ STRAND : Factor w/ 3 levels "","F","R": 2 2 3 3 2 3 2 3 2 2 ... $ UCSC_REFGENE_NAME : Factor w/ 42456 levels "","A1BG","A1CF;A1CF;A1CF",..: 19016 25371 1 37783 30924 812 811 1 1 20318 ... $ UCSC_REFGENE_ACCESSION : Factor w/ 52670 levels "","NM_000014",..: 38959 27879 1 36493 25264 27559 26 1 1 51089 ... $ UCSC_REFGENE_GROUP : Factor w/ 4165 levels "","1stExon","1stExon;1stExon",..: 2035 2035 1 2035 1216 2348 2970 1 1 2496 ... $ UCSC_CPG_ISLANDS_NAME : Factor w/ 27177 levels "","chr1:10003165-10003585",..: 15329 2125 1 1 1 15292 19510 1 2598 1 ... $ RELATION_TO_UCSC_CPG_ISLAND: Factor w/ 6 levels "","Island","N_Shelf",..: 2 2 1 1 1 4 6 1 5 1 ... $ SNP : Factor w/ 139064 levels "","rs10000296",..: 17352 122999 51875 43171 103744 112856 35391 40027 21722 58062 ... $ SNP_DISTANCE : Factor w/ 11304 levels "","-1","-1;0",..: 1351 639 852 852 1410 5828 852 3352 14 508 ... $ snp.hit : int 1 1 1 1 1 1 1 1 1 1 ... $ bwa.multi.hit : int 0 0 0 0 0 0 0 0 0 0 ... $ analyzed : int 0 0 0 0 0 0 0 0 0 0 ... $ DNase : int 0 0 0 0 0 0 1 0 0 0 ... $ h3k4me3 : int 0 0 0 0 0 0 1 0 0 0 ... $ h3k36me3 : int 0 0 0 0 0 0 0 0 0 0 ... $ h3k27ac : int 0 0 0 0 0 0 1 0 0 0 ... $ h3k4me1 : int 0 0 0 0 0 0 1 0 0 0 ... $ bivalent : int 0 0 0 0 0 0 0 0 0 0 ... $ h3k27me3 : int 0 0 0 0 0 1 0 0 0 0 ... $ h3k9me3 : int 0 0 0 0 0 0 0 0 0 0 ... $ dmc.constitutive : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.relapse : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.T : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.MLL : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.dic920 : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.HeH : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.t119 : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.t1221 : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.t922 : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.iamp : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.undefined : int NA NA NA NA NA NA NA NA NA NA ... $ dmc.nonrecurrent : int NA NA NA NA NA NA NA NA NA NA ...

Examples

```
data(snp.hit)
```

testDataSet *Test dataset.*

Description

This dataset is available to test ChAMP functions.

Usage

```
data(testDataSet)
```

Format

The format is: List of 6 $ mset :Formal class 'MethylSet' [package "minfi"] with 8 slots $ rgSet :Formal class 'RGChannelSet' [package "minfi"] with 7 slots $ pd :'data.frame': 6 obs. of 9 variables: ..$ Sample_Name ..$ Sample_Well ..$ Sample_Plate ..$ Sample_Group ..$ Pool_ID ..$ Array ..$ Slide ..$ Basename $ intensity $ beta $ detP

Examples

```
data(testDataSet)
```

Index

∗ **datasets**
  Anno450K, 3
  AnnoEPIC, 4
  bloodCtl, 4
  CellTypeMeans27K, 6
  CellTypeMeans450K, 7
  ControlProbes450K, 7
  ControlProbesEPIC, 8
  EPIC.manifest.hg19, 9
  EPIC.manifest.pop.hg19, 10
  EPICSimData, 11
  hm450.manifest.hg19, 12
  hm450.manifest.pop.hg19, 13
  hprdAsigH, 14
  illumina450Gr, 15
  illuminaEPICGr, 16
  MatchGeneName, 17
  multi.hit, 17
  PathwayList, 18
  probe.features, 19
  probe.features.epic, 20
  probeInfoALL.epic.lv, 21
  probeInfoALL.lv, 21
  snp.hit, 22
  testDataSet, 23
∗ **package**
  ChAMPdata-package, 2

Anno450K, 3
AnnoEPIC, 4

bloodCtl, 4

CellTypeMeans27K, 6
CellTypeMeans450K, 7
ChAMPdata *(*ChAMPdata-package*)*, 2
ChAMPdata-package, 2
ControlProbes450K, 7
ControlProbesEPIC, 8

EPIC.manifest.hg19, 9
EPIC.manifest.pop.hg19, 10
EPICSimData, 11

hm450.manifest.hg19, 12
hm450.manifest.pop.hg19, 13
hprdAsigH, 14

illumina450Gr, 15
illuminaEPICGr, 16

MatchGeneName, 17
multi.hit, 17

PathwayList, 18
probe.features, 19
probe.features.epic, 20
probeInfoALL.epic.lv, 21
probeInfoALL.lv, 21

snp.hit, 22

testDataSet, 23