# IlluminaHumanMethylation450kprobe

April 30, 2025

---

`IlluminaHumanMethylation450kprobe`
*Probe sequences for microarrays of type IlluminaHumanMethylation450*

---

## Description

Reannotation resource for Illumina HumanMethylation450 chips

## Usage

```
data(IlluminaHumanMethylation450kprobe)
```

## Format

A data frame with 485577 rows and 10 columns, as follows.

| | | |
|---|---|---|
| Probe_ID | character | Illumina probe ID |
| chr | factor | Chromosome of probe target in hg19 |
| strand | factor | Best strand match to hg19/GRCh37 |
| start | integer | Start coordinate of target in hg19 |
| end | integer | End coordinate of target in hg19 |
| site | character | Interrogated cytosine in hg19 |
| probe.sequence | character | Probe (allele A) sequence |
| source.sequence | character | Designed target sequence |
| forward.genomic.sequence | character | Closest match in hg19 |
| CpGs | integer | CpG sites (CpH/rs probes have 0) |

Interrogated di/trinucleotides span (site, site+(SNP=0,CpG=1,CpH=2)). CpH probe coordinates were liftOver()ed from hg18 to hg19 then aligned.

## Source

The probe sequence data was obtained from ftp://ftp.illumina.com. Data was extracted from HumanMethylation450_15017482_v.1.2.csv.

## Examples

```
library(IlluminaHumanMethylation450kprobe)
data(IlluminaHumanMethylation450kprobe)
head(IlluminaHumanMethylation450kprobe, 3)
summary(IlluminaHumanMethylation450kprobe)

# Let's use this data...
library(GenomicRanges)
chs = levels(IlluminaHumanMethylation450kprobe$chr)
names(chs) = paste('chr',chs,sep='')
CpGs.450k = with(IlluminaHumanMethylation450kprobe,
                 GRanges(paste('chr',chr,sep=''),
                         IRanges(start=site, width=2, names=Probe_ID),
                         strand=strand))

# verify the number of CpG sites in each probe:
library(Biostrings)
hm450 = with(IlluminaHumanMethylation450kprobe,
             DNAStringSet(forward.genomic.sequence))
head(dinucleotideFrequency(hm450)[,'CG'])
# [1] 3 2 1 1 3 1
tail(dinucleotideFrequency(hm450)[,'CG'])
# [1] 0 0 0 0 0 0 ...CpH probes (add rsID probes here!)

# find all the SNPs at CpG sites using GenomicRanges
library(parallel)
library(SNPlocs.Hsapiens.dbSNP.20110815)
CpG.snps.by.chr = mclapply(chs, function(ch) { # {{{ uses GenomicRanges
  snps = getSNPlocs(paste('ch', ch, sep=''), as.GRanges=TRUE)
  seqlevels(snps) <- gsub('ch','chr',seqlevels(snps))
  names(snps) = paste('rs',elementMetadata(snps)$RefSNP_id,sep='')
  message(paste('Scanning for CpG SNPs in probes on chromosome', ch))
  overlapping = findOverlaps(CpGs.450k, snps)@matchMatrix
  results = data.frame(
    Probe_ID=as.character(names(CpGs.450k)[overlapping[,1]]),
    rsID=as.character(names(snps)[overlapping[,2]])
  )
  return(results)
}) # }}}
SNPs = do.call(rbind, CpG.snps.by.chr)

# Obnoxious side effect of do.call(rbind)
SNPs$rsID = levels(SNPs$rsID)[SNPs$rsID]
SNPs$Probe_ID = levels(SNPs$Probe_ID)[SNPs$Probe_ID]

# For 27k array comparisons you could do...
# SNPs$hm27 = unlist(mget(SNPs$Probe_ID, IlluminaHumanMethylation450kMETHYL27))

# how many probes have SNPs at CpGs?
# message(nrow(SNPs))
# IlluminaHumanMethylation450kprobe$CpG.SNP = FALSE
# probe.SNPs = which(is.element(IlluminaHumanMethylation450kprobe$Probe_ID,
```

```
#                               SNPs$Probe_ID))
# IlluminaHumanMethylation450kprobe$CpG.SNP[probe.SNPs] = TRUE
#
# find repeats crossing CpG sites using IRanges
# library(BSgenome.Hsapiens.UCSC.hg19)
# CpG.rpts.by.chr = mclapply(chs, function(ch) { # {{{ uses IRanges
#   chr = Hsapiens[[paste('chr',ch,sep='')]]
#  rpts = union( masks(chr)$RM, masks(chr)$TRF )
#  probes = which(seqnames(CpGs.450k)==paste('chr',ch,sep=''))
#  # note how we have to use RangedData instead!!
#  CpGs.chr = ranges(CpGs.450k[probes])
#  message(paste('Scanning for repeats in CpG sites on chromosome', ch))
#  overlapping = findOverlaps(CpGs.chr, rpts)@matchMatrix
#  results = data.frame(
#    Probe_ID=as.character(names(CpGs.chr)[overlapping@matchMatrix[,1]]),
#    repeatID='RM/TRF'
#  )
#  return(results)
# }) # }}}
# RPTs = do.call(rbind, CpG.rpts.by.chr)

# how many probes have repeats at CpGs?
# message(nrow(RPTs))
# IlluminaHumanMethylation450kprobe$CpG.repeat = FALSE
# IlluminaHumanMethylation450kprobe$CpG.repeat[RPTs$Probe_ID] = TRUE

# how many have both?
# with(IlluminaHumanMethylation450kprobe,
#      sum(CpG.repeat & CpG.SNP))

# how many have either?
# with(IlluminaHumanMethylation450kprobe,
#     sum(CpG.repeat | CpG.SNP))

# We could change the above to find all SNPs and RPTs within probe targets:
# probes.450k = with(IlluminaHumanMethylation450kprobe,
#                    GRanges(paste('chr',chr,sep=''),
#                            IRanges(start=start, width=50, names=Probe_ID),
#                            strand=strand))
# Swap 'probes.450k'for 'CpGs.450k' in the previous lapply() loops to run.
# nb. If we want to look e.g. 10bp from the CpG site, then stranding matters.

# find the nearest TSS and its corresponding EntrezGene ID
library(GenomicFeatures)
CpGs.unstranded = CpGs.450k
strand(CpGs.unstranded) = '*'
refgene.TxDb = makeTranscriptDbFromUCSC('refGene', genome='hg19')

# nearest forward TSS
TSS.forward = transcripts(refgene.TxDb,
                                  vals=list(tx_strand='+'),
                                  columns='gene_id')
nearest.fwd = precede(CpGs.unstranded, TSS.forward)
```

```
nearest.fwd.eg = nearest.fwd # to keep dimensions right
notfound = which(is.na(nearest.fwd)) # track for later
nearest.fwd.eg[-notfound] = as.character(elementMetadata(TSS.forward)$gene_id[nearest.fwd[-notfound]])
TSSs.fwd = start(TSS.forward[nearest.fwd[-notfound]])
distToTSS.fwd = nearest.fwd # to keep dimensions right
distToTSS.fwd[-notfound] = start(CpGs.unstranded)[-notfound] - TSSs.fwd
# note that these are NEGATIVE -- which is correct!

# nearest reverse TSS
TSS.reverse = transcripts(refgene.TxDb,
                          vals=list(tx_strand='-'),
                          columns='gene_id')
nearest.rev = precede(CpGs.unstranded, TSS.reverse)
nearest.rev.eg = nearest.rev # to keep dimensions right
notfound = which(is.na(nearest.rev)) # track for later
nearest.rev.eg[-notfound] = as.character(elementMetadata(TSS.reverse)$gene_id[nearest.rev[-notfound]])
TSSs.rev = start(TSS.reverse[nearest.rev[-notfound]])
distToTSS.rev = nearest.rev # to keep dimensions right
distToTSS.rev[-notfound] = start(CpGs.unstranded)[-notfound] - TSSs.rev
# now these are POSITIVE: we are walking up the opposite strand.

# tabulate and link these together for the annotation package:
IlluminaHumanMethylation450kprobe$fwd.dist <- distToTSS.fwd
IlluminaHumanMethylation450kprobe$fwd.gene_id <- nearest.fwd.eg
IlluminaHumanMethylation450kprobe$rev.dist <- distToTSS.rev
IlluminaHumanMethylation450kprobe$rev.gene_id <- nearest.rev.eg

FWD.CLOSER = with(IlluminaHumanMethylation450kprobe,
                  union( which( abs(fwd.dist) < abs(rev.dist) ),
                         which( is.na(rev.dist) ) ) )
REV.CLOSER = with(IlluminaHumanMethylation450kprobe,
                  union( which( abs(fwd.dist) > abs(rev.dist) ),
                         which( is.na(fwd.dist) ) ) )
IlluminaHumanMethylation450kprobe$DISTTOTSS = pmin(abs(IlluminaHumanMethylation450kprobe$fwd.dist), abs(Illumin
IlluminaHumanMethylation450kprobe$ENTREZ = NA
IlluminaHumanMethylation450kprobe$ENTREZ[FWD.CLOSER] = IlluminaHumanMethylation450kprobe$fwd.gene_id
IlluminaHumanMethylation450kprobe$ENTREZ[REV.CLOSER] = IlluminaHumanMethylation450kprobe$rev.gene_id


# find the observed/expected CpG density around each site:
#
library(BSgenome.Hsapiens.UCSC.hg19)
window.width = 500 # could use larger or smaller
oecg.by.chr = mclapply(chs, function(ch) {
  probes = which(IlluminaHumanMethylation450kprobe$chr == ch)
  probecpgs = with(IlluminaHumanMethylation450kprobe[probes,],
                   IRanges(start=site, width=2, names=Probe_ID))
  cpgwindows = resize(probecpgs, fix="center", width=window.width)
  chr = Hsapiens[[paste('chr',ch,sep='')]]
  active(masks(chr)) = FALSE
  chr.seqs = Views(chr, cpgwindows)
  ocg = dinucleotideFrequency(chr.seqs, as.prob=T)[,'CG']
  c.g = alphabetFrequency(chr.seqs, as.prob=T,baseOnly=T)
```

```
    ecg = c.g[,'C'] * c.g[,'G']
  return(ocg/ecg)
})
```

# Index

∗ **datasets**
IlluminaHumanMethylation450kprobe, 1

IlluminaHumanMethylation450kprobe, 1