# Package 'methylclock'

May 26, 2026

**Type** Package

**Title** Methylclock - DNA methylation-based clocks

**Version** 1.18.0

**Description** This package allows to estimate chronological and gestational DNA methylation (DNAm) age as well as biological age using different methylation clocks.
Chronological DNAm age (in years) : Horvath's clock, Hannum's clock, BNN, Horvath's skin+blood clock, PedBE clock and Wu's clock.
Gestational DNAm age : Knight's clock, Bohlin's clock, Mayne's clock and Lee's clocks.
Biological DNAm clocks : Levine's clock and Telomere Length's clock.

**biocViews** DNAMethylation, BiologicalQuestion, Preprocessing, StatisticalMethod, Normalization

**License** MIT + file LICENSE

**Depends** R (>= 4.1.0), methylclockData, devtools, quadprog

**Imports** Rcpp (>= 1.0.6), ExperimentHub, dplyr, impute, PerformanceAnalytics, Biobase, ggpmisc, tidyverse, ggplot2, ggpubr, minfi, tibble, RPMM, stats, graphics, tidyr, gridExtra, preprocessCore, dynamicTreeCut, planet

**Suggests** BiocStyle, knitr, GEOquery, rmarkdown

**LinkingTo** Rcpp

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**URL** https://github.com/isglobal-brge/methylclock

**BugReports** https://github.com/isglobal-brge/methylclock/issues

**VignetteBuilder** knitr

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

**git_branch** RELEASE_3_23

**git_last_commit** a236f18

**git_last_commit_date** 2026-04-28

**Repository** Bioconductor 3.23

**Date/Publication** 2026-05-25

**Author** Dolors Pelegri-Siso [aut, cre] (ORCID:
<https://orcid.org/0000-0002-5993-3003>),
Juan R. Gonzalez [aut] (ORCID: <https://orcid.org/0000-0003-3267-2146>)

**Maintainer** Dolors Pelegri-Siso <dolors.pelegri@isglobal.org>

# Contents



---

`checkClocks` *Check wheter input data contains the required CpGs for the implemented clocks.*

---

## Description

Check wheter input data contains the required CpGs for the implemented clocks.

## Usage

```
checkClocks(x, ...)
```

## Arguments

| | |
|---|---|
| `x` | data.frame or tibble (Individual in columns, CpGs in rows, CpG names in first colum - i.e. Horvath's format), ExpressionSet or GenomicRatioSet. A matrix is also possible having the CpG names in the rownames. |
| `...` | other parameters |

## Details

To be supplied

## Value

a list with the different clocks when there are more than 80 the required CpGs

## Examples

```
TestDataset <- get_TestDataset()
checkClocks(TestDataset)
```

---

`checkClocksGA` *Check wheter input data contains the required CpGs for the implemented clocks for Gestational Age.*

---

## Description

Check wheter input data contains the required CpGs for the implemented clocks for Gestational Age.

## Usage

```
checkClocksGA(x, ...)
```

## Arguments

| | |
|---|---|
| x | data.frame or tibble (Individual in columns, CpGs in rows, CpG names in first colum - i.e. Horvath's format), ExpressionSet or GenomicRatioSet. A matrix is also possible having the CpG names in the rownames. |
| ... | other parameters |

## Details

To be supplied

## Value

a list with the different GA clocks when there are more than 80

## Examples

```
TestDataset <- get_TestDataset()
checkClocksGA(TestDataset)
```

## commonClockCpgs *Get common CpGs*

### Description

Show the required CpGs contained on input data for the implemented clocks

### Usage

```
commonClockCpgs(object, clock)
```

### Arguments

| | |
|---|---|
| object | resulting object from checkClocks functions |
| clock | string with the implemented clock, possible values are : "Knight", "Bohlin", "Mayne" and "Lee", "Horvath", "Hannum", "Levine", "skinHorvath", "PedBE", "Wu" and "TL" |

### Value

The common CpGs between input data and defined GA clock

### Examples

```
TestDataset <- get_TestDataset()
cpgs.missing.GA <- checkClocksGA(TestDataset)
cpgs.missing <- checkClocks(TestDataset)
commonClockCpgs(cpgs.missing.GA, "Bohlin")
commonClockCpgs(cpgs.missing, "Hannum")
```

## DNAmAge *DNAm age estimation using different DNA methylation clocks.*

### Description

DNAm age estimation using different DNA methylation clocks.

### Usage

```
DNAmAge(
  x,
  clocks = "all",
  toBetas = FALSE,
  fastImp = FALSE,
  normalize = FALSE,
  age,
```

```
  cell.count = TRUE,
  cell.count.reference = "blood gse35069 complete",
  min.perc = 0.8,
  ...
)
```

## Arguments

| | |
|---|---|
| x | data.frame (Individual in columns, CpGs in rows, CpG names in first colum - i.e. Horvath's format), matrix (individuals in columns and Cpgs in rows having CpG names in the rownames), ExpressionSet or GenomicRatioSet. |
| clocks | the methods used for estimating DNAmAge. Currrently "Horvath", "Hannum", "Levine", "BNN", "skinHorvath", "PedBE", "Wu", "TL", "BLUP", "EN" and "all" are available. Default is "all" and all clocks are estimated. |
| toBetas | Should data be transformed to beta values? Default is FALSE. If TRUE, it implies data are M values. |
| fastImp | Is fast imputation performed if necessary? (see details). Default is FALSE |
| normalize | Is Horvath's normalization performed? By default is FALSE |
| age | individual's chronological age. |
| cell.count | Are cell counts estimated? Default is TRUE. |
| cell.count.reference | Used when 'cell.count' is TRUE. Default is "blood gse35069 complete".See 'meffil::meffil.list.cell.count.references()' for possible values. |
| min.perc | Indicates the minimum conicidence percentage required between CpGs in or dataframee x and CpGs in clock coefficients to perform the calculation. If min.prec is too low, the estimated gestational DNAm age can be poor |
| ... | Other arguments to be passed through impute package |

## Details

Imputation is performed when having missing data. Fast imputation is performed by ... what about imputing only when CpGs for the clock are missing?

## Value

The estimated chronological and biological mDNA age

## Examples

```
MethylationData <- get_MethylationDataExample()
age.example55 <- DNAmAge(MethylationData)
```

DNAmGA *Gestational DNAm age estimation using different DNA methylation clocks.*

## Description

Gestational DNAm age estimation using different DNA methylation clocks.

## Usage

```
DNAmGA(
  x,
  toBetas = FALSE,
  fastImp = FALSE,
  normalize = FALSE,
  age,
  cell.count = TRUE,
  cell.count.reference = "andrews and bakulski cord blood",
  min.perc = 0.8,
  ...
)
```

## Arguments

| | |
|---|---|
| x | data.frame (Individual in columns, CpGs in rows, CpG names in first colum - i.e. Horvath's format), matrix (individuals in columns and Cpgs in rows having CpG names in the rownames), ExpressionSet or GenomicRatioSet. |
| toBetas | Should data be transformed to beta values? Default is FALSE. If TRUE, it implies data are M values. |
| fastImp | Is fast imputation performed if necessary? (see details). Default is FALSE |
| normalize | Is Horvath's normalization performed? By default is FALSE |
| age | individual's chronological age. Required to compute gestational age difference output |
| cell.count | Are cell counts estimated? Default is TRUE. |
| cell.count.reference | Used when 'cell.count' is TRUE. Default is "blood gse35069 complete". See 'meffil::meffil.list.cell.count.references()' for possible values. |
| min.perc | Indicates the minimum conicidence percentage required between CpGs in or dataframee x and CpGs in clock coefficients to perform the calculation. If min.prec is too low, the estimated gestational DNAm age can be poor |
| ... | Other arguments to be passed through impute package |

## Details

Imputation is performed when having missing data. Fast imputation is performed by ... what about imputing only when CpGs for the clock are missing?

## Value

the estimated gestational DNAm age

## Examples

```
TestDataset <- get_TestDataset()
TestDataset[1:5, ]
ga.test <- DNAmGA(TestDataset)
```

---

`getCellTypeReference` *Get cell type reference*

---

## Description

Get cell type reference

## Usage

```
getCellTypeReference(name)
```

## Arguments

| | |
|---|---|
| `name` | string with predefined datasets andrews and bakulski cord blood, blood gse35069, blood gse35069 chen, blood gse35069 complete, "combined cord blood", "cord blood gse68456", "gervin and lyle cord blood", "guintivano dlpfc" or "saliva gse48472" |

## Details

ORIGINAL AUTHOR: Matthew Suderman at githug : https://github.com/perishky/meffil The original meffilListCellTypeReferences and getCellTypeReference function from meffil v1.0.0

## Value

name and reference.globals

## Examples

```
name <- "andrews and bakulski cord blood"
getCellTypeReference(name)
```

load_DNAmGA_Clocks_data *Loads DNAmGA clock data from methylclockData*

## Description

Loads DNAmGA clock data from methylclockData

## Usage

```
load_DNAmGA_Clocks_data()
```

## Value

void

## Examples

```
load_DNAm_Clocks_data()
```

load_DNAm_Clocks_data *Loads DNAm clock data from methylclockData*

## Description

Loads DNAm clock data from methylclockData

## Usage

```
load_DNAm_Clocks_data()
```

## Value

void

## Examples

```
load_DNAm_Clocks_data()
```

`meffilEstimateCellCountsFromBetas` *Estimate cell counts for a beta matrix from a reference*

## Description

Estimate cell type ratios from methylation profiles of purified cell populations (Infinium HumanMethylation450 BeadChip).

## Usage

```
meffilEstimateCellCountsFromBetas(beta, cellTypeReference, verbose = FALSE)
```

## Arguments

| | |
|---|---|
| `beta` | Matrix of Illumina 450K methylation levels (rows = CpG sites, columns = subjects). |
| `cellTypeReference` | Character string name of the cell type reference to use for estimating cell counts. See `meffilListCellTypeReferences()` for a list of available references. New references can be created using |
| `verbose` | If `TRUE`, then status messages are printed during execution (Default: `FALSE`). |

## Details

ORIGINAL AUTHOR: Matthew Suderman The original meffil.list.cellTypeReferences and get.cellTypeReference function from meffil v1.0.0 downloaded from githug : https://github.com/perishky/meffil

## Value

A matrix of cell count estimates.

Results should be nearly identical to `minfi::estimateCellCounts()`

betas

## Examples

```
cell.count.reference <- "andrews and bakulski cord blood"
TestDataset <- get_TestDataset()
cpgs <- t(as.matrix(TestDataset[, -1]))
colnames(cpgs) <- TestDataset$CpGName
meffilEstimateCellCountsFromBetas(t(cpgs), cell.count.reference)
```

`meffilListCellTypeReferences` *List of available cell type references*

## Description

List of available cell type references

## Usage

```
meffilListCellTypeReferences()
```

## Details

ORIGINAL AUTHOR: Matthew Suderman The original meffilListCellTypeReferences and getCellTypeReference function from meffil v1.0.0 at githug : https://github.com/perishky/meffil

## Value

a list with reference globals

## Examples

```
meffilListCellTypeReferences()
```

`methylclock` *methylclock*

## Description

Package to estimate DNA methylation age (DNAmAge) using different methylation clocks.

## Author(s)

Juan R Gonzalez <juanr.gonzalez@isglobal.org>

plotCorClocks *Plot correlation among DNAm clockx*

## Description

Plot correlation among DNAm clockx

## Usage

```
plotCorClocks(x, ...)
```

## Arguments

| | |
|---|---|
| x | a tible or data.frame with the different DNAm clocks |
| ... | other arguments to be passs through function 'chart.Correlation' from 'PerformanceAnalytics' package |

## Details

To be supplied

## Value

Plot with Correlation Clocks

## Examples

```
library(Biobase)
library(GEOquery)

dd <- GEOquery::getGEO("GSE109446")
gse109446 <- dd[[1]]
controls <- Biobase::pData(gse109446)$`diagnosis:ch1` == "control"
gse <- gse109446[, controls]
age <- as.numeric(Biobase::pData(gse)$`age:ch1`)
age.gse <- DNAmAge(gse, age = age)
plotCorClocks(age.gse)
```

plotDNAmAge *Plot DNAm age estimation vs chronological age.*

## Description

Plot DNAm age estimation vs chronological age.

## Usage

```
plotDNAmAge(x, y, tit = "Horvath's method", clock = "chronological", ...)
```

## Arguments

| | |
|---|---|
| x | DNAm age estimation |
| y | Chronological age |
| tit | Plot title. Default is "Horvath's method". |
| clock | Type of clock 'chronological' or 'GA', default 'chronological' |
| ... | Other plot parameters for ggplot |

## Value

Plot with estimated DNAmAge

## Examples

```
library(tidyverse)

path <- system.file("extdata", package = "methylclock")
covariates <- read_csv(file.path(
path,
"SampleAnnotationExample55.csv"
))
age <- covariates$Age
MethylationData <- get_MethylationDataExample()

age.example55 <- DNAmAge(MethylationData)
plotDNAmAge(age.example55$Horvath, age)
```

## progress_data *PROGRESS cohor data*

### Description

The PROGRESS cohort data is available in the additional file 8 of : Knight, A.K., Craig, J.M., Theda, C. et al. An epigenetic clock for gestational age at birth based on blood methylation data. Genome Biol 17, 206 (2016). https://doi.org/10.1186/s13059-016-1068-z

### Usage

```
data(progress_data)
```

### Format

A data frame with 148 obs. and 151 variables

### Details

A dataset containing data from the PROGRESS (Programming Research in Obesity, Growth, Environment and Social Stressors) cohort

### Examples

```
data(progress_data)
```

## progress_vars *PROGRESS cohort variables*

### Description

The PROGRESS cohort data is available in the additional file 8 of : Knight, A.K., Craig, J.M., Theda, C. et al. An epigenetic clock for gestational age at birth based on blood methylation data. Genome Biol 17, 206 (2016). https://doi.org/10.1186/s13059-016-1068-z

### Usage

```
data(progress_vars)
```

### Format

A data frame with 150 obs. and 3 variables

### Details

A dataset containing data from the PROGRESS (Programming Research in Obesity, Growth, Environment and Social Stressors) cohort

## Examples

```
data(progress_vars)
```

# Index