/Group « /S /Transparency /I true /CS /DeviceRGB»

ioctitlebempty ldtitle

[1]ldtitle1

oifpackagelaterparnotes2016/07/26 arnote@real

parnote@real

ooldtitleChronological and gestational DNAm age estimation using different methylation-based clocks

itle

uthor

ate

Package

methylclock 0.5.0

Contents

1 Description of implemented clocks

This manual describes how to estimate chronological and gestational DNA methylation (DNAm) age as well as biological age using different methylation clocks. The package includes the following estimators:

1.1 Chronological DNAm age (in years)

- **Horvaths clock**: It uses 353 CpGs described in Horvath (2013). It was trained using 27K and 450K arrays in samples from different tissues. Other three different age-related biomarkers are also computed:
  - **AgeAcDiff** (DNAmAge acceleration difference): Difference between DNAmAge and chronological age.
  - **IEAA** (Intrinsic Epigenetic Age Acceleration): Residuals obtained after regressing DNAmAge and chronological age adjusted by cell counts.
  - **EEAA** (Extrinsic Epigenetic Age Acceleration): Residuals obtained after regressing DNAmAge and chronological age. This measure was also known as DNAmAge acceleration residual in the first Horvath's paper.
- **Hannum's clock**: It uses 71 CpGs described in Hannum et al. (2013). It was trained using 450K array in blood samples. Another are-related biomarer is also computed:
  - **AMAR** (Apparent Methylomic Aging Rate): Measure proposed in Hannum et al. (2013) computed as the ratio between DNAm age and the chronological age.
- **BNN**: It uses Horvath's CpGs to train a Bayesian Neural Network (BNN) to predict DNAm age as described in Alfonso and Gonzalez (2018).
- **Horvath's skin+blood clock (Horvath2)**: Epigenetic clock for skin and blood cells. It uses 391 CpGs described in Horvath et al. (2018). It was trained using 450K EPIC arrays in skin and blood sampels.
- **PedBE clock**: Epigenetic clock from buccal epithelial swabs. It's intended purpose is buccal samples from individuals aged 0-20 years old. It uses 84 CpGs described in McEwen et al. (2019). The authors gathered 1,721 genome-wide DNAm profiles from 11 different cohorts with individuals aged 0 to 20 years old.

1.2 Gestational DNAm age (in weeks)

- **Knight's clock**: It uses 148 CpGs described in Knight et al. (2016). It was trained using 27K and 450K arrays in coord blood samples.
- **Bohlin's clock**: It uses 96 CpGs described in Bohlin et al. (2016). It was trained using 450K array in coord blood samples.
- **Mayne's clock**: It uses 62 CpGs described in Mayne et al. (2017). It was trained using 27K and 450K.
- **Lee's clocks**: Three different biological clocks described in Lee et al. (2019) are implemented. It was trained for 450K and EPIC arrays in placenta samples.
  - **RPC clock**: Robust placental clock (RPC). It uses 558 CpG sites.
  - **CPC clock**: Control placental clock (CPC). It usses 546 CpG sites.
  - **Refined RPC clock**: Useful for uncomplicated term pregnancies (e.g. gestational age >36 weeks). It uses 396 CpG sites.

The biological DNAm clocks implemented in our package are:

- **Levine's clock** (also know as PhenoAge): It uses 513 CpGs described in Levine et al. (2018). It was trained using 27K, 450K and EPIC arrays in blood samples.

The main aim of this package is to facilitate the interconnection with R and Bioconductor's infrastructure and, hence, avoiding submitting data to online calculators. Additionally, `methylclock` also provides an unified way of computing DNAm age to help downstream analyses.

2 Getting started

The package depends on some R packages that can be previously installed into your computer by:

otalleftmargin@ etminipage

```
library(BiocManager)
install(c("tidyverse", "impute", "Rcpp", "GAprediction"))

library(devtools)
install_github("perishky/meffil")
```

otalleftmargin otalleftmargin inipagefalse

Then `methylclock` package is installed into your computer by executing:

otalleftmargin@ etminipage

```
install_github("isglobal-brge/methylclock")
```

otalleftmargin otalleftmargin inipagefalse

The package is loaded into R as usual:

otalleftmargin@ etminipage

```
library(methylclock)
```

otalleftmargin otalleftmargin inipagefalse

These libraries are required to reproduce this document:

otalleftmargin@ etminipage

```
library(Biobase)
library(tibble)
library(ggplot2)
library(ggpmisc)
library(GEOquery)
```

otalleftmargin otalleftmargin inipagefalse

3 DNA Methylation clocks

The main function to estimate chronological and biological mDNA age is called `DNAmAge` while the gestational DNAm age is estimated using `DNAmGA` function. Both functions have similar input arguments. Next subsections detail some of the important issues to be consider before computind DNAm clocks.

3.1 Data format

The methylation data is given in the argument `x`. They can be either beta or M values. The argument `toBetas` should be set to TRUE when M values are provided. The `x` object can be:

- A **matrix** with CpGs in rows and individuals in columns having the name of the CpGs in the rownames.
- A **data frame** or a **tibble** with CpGs in rows and individuals in columns having the name of the CpGs in the first column (e.g. cg00000292, cg00002426, cg00003994, . . .) as required in the Horvath's DNA Methylation Age Calculator website (https://dnamage.genetics.ucla.edu/home).
- A **GenomicRatioSet** object, the default method to encapsulate methylation data in `minfi` Bioconductor package.
- An **ExpressionSet** object as obtained, for instance, when downloading methylation data from GEO (https://www.ncbi.nlm.nih.gov/geo/).

3.2 Data nomalization

In principle, data can be normalized by using any of the existing standard methods such as QN, ASMN, PBC, SWAN, SQN, BMIQ (see a revision of those methods in Wang et al. (2015)). `DNAmAge` function includes the BMIQ method proposed by Teschendorff et al. (2012) using Horvath's robust implementation that basically consists of an optimal R code implementation and optimization procedures. This normalization is recommended by Horvath since it improves the predictions for his clock. This normalization procedure is very time-consuming. In order to overcome these difficulties, we have parallelize this process using `BiocParallel` library. This step is not mandatory, so that, you can use your normalized data and set the argument `normalize` equal to FALSE (default).

3.3 Missing individual's data

All the implemented methods require complete cases. `DNAmAge` function has an imputation method based on KNN implemented in the function `knn.impute` from `impute` Bioconductor package. This is performed when missing data is present in the CpGs used in any of the computed clocks. There is also another option based on a fast imputation method that imputes missing values by the median of required CpGs as recommended in Bohlin et al. (2016). This is recommended when analyzing 450K arrays since `knn.impute` for large datasets may be very time consuming. Fast imputation can be performed by setting `fastImp=TRUE` which is not the default value.

3.4 Missing CpGs of DNAm clocks

By default the package computes the different clocks when there are more than 80% of the required CpGs of each method. Nothing is required when having missing CpGs since the main functions will return NA for those estimators when this criteria is not meet. Let us use a test dataset (`TestDataset`) which is available within the package to illustrate the type of information we are obtaining:

otalleftmargin otalleftmargin

```
otalleftmargin@ etminipage
```

```
cpgs.missing <- checkClocks(TestDataset)
          clock Cpgs_in_clock missing_CpGs percentage
  1     Horvath           354            2        0.6
  2      Hannum            71           64       90.1
  3      Levine           514            3        0.6
  4 SkinHorvath           392          283       72.2
  5       PedBE            95           91       95.8
  There are some clocks that cannot be computed since your data do not contain the required CpGs
          These are the total number of missing CpGs for each clock :
```

otalleftmargin otalleftmargin inipagefalse

otalleftmargin@ etminipage

```
cpgs.missing.GA <- checkClocksGA(TestDataset)
     clock Cpgs_in_clock missing_CpGs percentage
  1 Knight           149            0        0.0
  2 Bohlin            96           87       90.6
  3  Mayne            63            0        0.0
  4    Lee          1126         1072       95.2
  There are some clocks that cannot be computed since your data do not contain the required CpGs
          These are the total number of missing CpGs for each clock :

     clock Cpgs_in_clock missing_CpGs percentage
  1 Knight           149            0        0.0
  2 Bohlin            96           87       90.6
  3  Mayne            63            0        0.0
  4    Lee          1126         1072       95.2
```

otalleftmargin otalleftmargin inipagefalse

The objects `cpgs.missing` and `cpgs.missing.GA` are lists havint the missing CpGs of each clock

otalleftmargin@ etminipage

```
names(cpgs.missing)
  [1] "Horvath"  "Hannum"   "Levine"   "Horvath2" "PedBE"
cpgs.missing$Hannum
   [1] "cg20822990"      "cg22512670"      "cg25410668"      "cg04400972"
   [5] "cg16054275"      "cg10501210"      "ch.2.30415474F"  "cg22158769"
   [9] "cg02085953"      "cg06639320"      "cg22454769"      "cg24079702"
  [13] "cg23606718"      "cg22016779"      "cg03607117"      "cg07553761"
  [17] "cg00481951"      "cg25478614"      "cg25428494"      "cg02650266"
  [21] "cg08234504"      "cg23500537"      "cg20052760"      "cg16867657"
  [25] "cg06685111"      "cg00486113"      "cg13001142"      "cg20426994"
  [29] "cg14361627"      "cg08097417"      "cg07955995"      "cg22285878"
  [33] "cg03473532"      "cg08540945"      "cg07927379"      "cg16419235"
  [37] "cg07583137"      "cg22796704"      "cg19935065"      "cg23091758"
  [41] "cg23744638"      "cg04940570"      "cg11067179"      "cg22213242"
  [45] "cg06419846"      "cg02046143"      "cg00748589"      "cg18473521"
  [49] "cg01528542"      "ch.13.39564907R" "cg03032497"      "cg04875128"
```

otalleftmargin otalleftmargin

```
[53] "cg09651136"     "cg03399905"     "cg04416734"     "cg07082267"
[57] "cg14692377"     "cg06874016"     "cg21139312"     "cg02867102"
[61] "cg19283806"     "cg14556683"     "cg07547549"     "cg08415592"
```

otalleftotalleftmargin inipagefalse

3.5 Cell counts

The EEAA method requires to estimate cell counts. We use the package `meffil` (Min et al. (2018)) that provides some functions to estimate cell counts using predefined datasets. This is performed by setting `cell.count=TRUE` (default value). The reference panel is passed through the argument `cell.count.reference`. So far, the following options are available:

- **"blood gse35069 complete"**: methylation profiles from Reinius et al. (2012) for purified blood cell types. It includes CD4T, CD8T, Mono, Bcell, NK, Neu and Eos.
- **"blood gse35069"**: methylation profiles from Reinius et al. (2012) for purified blood cell types. It includes CD4T, CD8T, Mono, Bcell, NK and Gran.
- **"blood gse35069 chen"**: methylation profiles from Chen et al. (2017) blood cell types. It includes CD4T, CD8T, Mono, Bcell, NK, Neu and Eos.
- **"andrews and bakulski cord blood"**. Cord blood reference from Bakulski et al. (2016). It includes Bcell, CD4T, CD8T, Gran, Mono, NK and nRBC.
- **"cord blood gse68456"** Cord blood methylation profiles from Goede et al. (2015). It includes CD4T, CD8T, Mono, Bcell, NK, Neu, Eos and RBC.
- **"gervin and lyle cord blood"** Cord blood reference generated by Kristina Gervin and Robert Lyle, available at `miffil` package. It includes CD14, Bcell, CD4T, CD8T, NK, Gran.
- **"saliva gse48472"**: Reference generated from the multi-tissue pannel from Slieker et al. (2013). It includes Buccal, CD4T, CD8T, Mono, Bcell, NK, Gran.

4 Chronological and biological DNAm age estimation

Next we illustrate how to estimate the chronological DNAm age using several datasets which aim to cover different data input formats.

4.1 Data in Horvath's format (e.g. `csv` with CpGs in rows)

Let us start by reproducing the results proposed in Horvath (2013). It uses the format available in the file 'MethylationDataExample55.csv" from his tutorial (available here). These data are available at `methylclock` package. Although these data can be loaded into R by using standard functions such as `read.csv` we hihgly recommend to use functions from `tidiverse`, in particular `read_csv` from `readr` package. The main reason is that currently researchers are analyzing Illumina 450K or EPIC arrays that contains a huge number of CpGs that can take a long time to be loaded when using basic importing R function. These functions import `csv` data as tibble which is one of the possible formats of `DNAmAge` function

otalleftmargin@ etminipage

```
library(tidyverse)
path <- system.file("extdata", package = "methylclock")
MethylationData <- read_csv(file.path(path, "MethylationDataExample55.csv"))
```

otalleftotalleftmargin

```
MethylationData
  # A tibble: 27,578 x 17
     ProbeID GSM946048 GSM946049 GSM946052 GSM946054 GSM946055 GSM946056
     <chr>       <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
   1 cg0000~    0.706     0.730     0.705     0.751     0.715     0.634
   2 cg0000~    0.272     0.274     0.311     0.279     0.178     0.269
   3 cg0000~    0.0370    0.0147    0.0171    0.0290    0.0163    0.0243
   4 cg0000~    0.133     0.120     0.121     0.107     0.110     0.129
   5 cg0000~    0.0309    0.0192    0.0217    0.0132    0.0181    0.0243
   6 cg0000~    0.0700    0.0715    0.0655    0.0719    0.0914    0.0508
   7 cg0000~    0.993     0.993     0.993     0.994     0.991     0.994
   8 cg0000~    0.0215    0.0202    0.0187    0.0169    0.0162    0.0143
   9 cg0000~    0.0105    0.00518   0.00410   0.00671   0.00758   0.00518
  10 cg0001~    0.634     0.635     0.621     0.639     0.599     0.591
  # ... with 27,568 more rows, and 10 more variables: GSM946059 <dbl>,
  #   GSM946062 <dbl>, GSM946064 <dbl>, GSM946065 <dbl>, GSM946066 <dbl>,
  #   GSM946067 <dbl>, GSM946073 <dbl>, GSM946074 <dbl>, GSM946075 <dbl>,
  #   GSM946076 <dbl>
```

otalleftmargin inipagefalse

IMPORTANT NOTE: Be sure that the first column contains the CpG names. Sometimes, your imported data look like this one (it can happen, for instance, if the `csv` file was created in R without indicating `row.names=FALSE`)

otalleftmargin@ etminipage

```
> mydata

# A tibble: 473,999 x 6
      X1 Row.names BIB_15586_1X BIB_33043_1X EDP_5245_1X KAN_584_1X
   <int> <chr>            <dbl>        <dbl>       <dbl>      <dbl>
 1     1 cg000000~        0.635        0.575       0.614      0.631
 2     2 cg000001~        0.954        0.948       0.933      0.950
 3     3 cg000001~        0.889        0.899       0.901      0.892
 4     4 cg000001~        0.115        0.124       0.107      0.123
 5     5 cg000002~        0.850        0.753       0.806      0.815
 6     6 cg000002~        0.676        0.771       0.729      0.665
 7     7 cg000002~        0.871        0.850       0.852      0.863
 8     8 cg000003~        0.238        0.174       0.316      0.206
```

otalleftmargin inipagefalse

If so, the first column must be removed before being used as the input object in `DNAmAge` funcion. It can be done using `dplyr` function

otalleftmargin@ etminipage

```
> mydata2 <- select(mydata, -1)

# A tibble: 473,999 x 5
      Row.names BIB_15586_1X BIB_33043_1X EDP_5245_1X KAN_584_1X
    <chr>              <dbl>        <dbl>       <dbl>      <dbl>
```

otalleftmargin

```
        1    cg000000~       0.635        0.575        0.614       0.631
        2    cg000001~       0.954        0.948        0.933       0.950
        3    cg000001~       0.889        0.899        0.901       0.892
        4    cg000001~       0.115        0.124        0.107       0.123
        5    cg000002~       0.850        0.753        0.806       0.815
        6    cg000002~       0.676        0.771        0.729       0.665
        7    cg000002~       0.871        0.850        0.852       0.863
        8    cg000003~       0.238        0.174        0.316       0.206
```

otalleftmargin inipagefalse

In any case, if you use the object `mydata` that contains the CpGs in the second column, you will see this error message:

otalleftmargin@ etminipage

```
> DNAmAge(mydata)
Error in DNAmAge(mydata) : First column should contain CpG names
```

otalleftmargin inipagefalse

DNAmAge can be estimated by simply:

otalleftmargin@ etminipage

```
age.example55 <- DNAmAge(MethylationData)
age.example55
  # A tibble: 16 x 7
     id         Horvath Hannum Levine   BNN skinHorvath PedBE
     <fct>        <dbl> <lgl>   <dbl> <dbl> <lgl>       <lgl>
   1 GSM946048    51.8  NA      -30.3 56.4  NA          NA
   2 GSM946049    39.8  NA      -29.6 42.1  NA          NA
   3 GSM946052    26.4  NA      -33.3 25.6  NA          NA
   4 GSM946054    34.0  NA      -36.0 28.0  NA          NA
   5 GSM946055    10.1  NA      -52.8 13.4  NA          NA
   6 GSM946056    20.4  NA      -42.2 16.7  NA          NA
   7 GSM946059     6.00 NA      -44.8  7.54 NA          NA
   8 GSM946062    34.6  NA      -23.2 34.6  NA          NA
   9 GSM946064     7.91 NA      -49.8 12.0  NA          NA
  10 GSM946065     4.72 NA      -48.2  6.43 NA          NA
  11 GSM946066    29.6  NA      -39.9 28.5  NA          NA
  12 GSM946067     1.38 NA      -48.3  3.48 NA          NA
  13 GSM946073    56.0  NA      -26.7 47.3  NA          NA
  14 GSM946074    24.0  NA      -39.7 23.3  NA          NA
  15 GSM946075     9.38 NA      -45.4 11.9  NA          NA
  16 GSM946076    38.8  NA      -27.5 41.4  NA          NA
```

otalleftmargin inipagefalse

By default all available clocks (Hovarth, Hannum, Levine, BNN, Hovart2 and PedBE) are estimated. One may select a set of clocks by using the argument `clocks` as following:

otalleftmargin otalleftmargin@ etminipage

```
age.example55.sel <- DNAmAge(MethylationData,
                             clocks=c("Horvath", "BNN"))
age.example55.sel
  # A tibble: 16 x 3
     id        Horvath   BNN
     <fct>       <dbl> <dbl>
   1 GSM946048   51.8  56.4
   2 GSM946049   39.8  42.1
   3 GSM946052   26.4  25.6
   4 GSM946054   34.0  28.0
   5 GSM946055   10.1  13.4
   6 GSM946056   20.4  16.7
   7 GSM946059    6.00  7.54
   8 GSM946062   34.6  34.6
   9 GSM946064    7.91 12.0
  10 GSM946065    4.72  6.43
  11 GSM946066   29.6  28.5
  12 GSM946067    1.38  3.48
  13 GSM946073   56.0  47.3
  14 GSM946074   24.0  23.3
  15 GSM946075    9.38 11.9
  16 GSM946076   38.8  41.4
```

otalleotalleftmargin

inipagefalse

4.2 Age acceleration

However, in epidemiological studies one is intereste in assessing whether age acceleration is associated with a given trait or condition. Three different measures can be computed:

- **ageAcc**: Difference between DNAmAge and chronological age.
- **ageAcc2**: Residuals obtained after regressing chronological age and DNAmAge (similar to IEAA).
- **ageAcc3**: Residuals obtained after regressing chronological age and DNAmAge adjusted for cell counts (similar to EEAA).

All this estimates can be obtained for each clock when providing chronological age through `age` argument. This information is normally provided in a different file including different covariates (metadata or sample annotation data). In this example data are available at ‘SampleAnnotationExample55.csv’ file that is also available at `methylclock` package:

otalleftmargin@ etminipage

```
covariates <- read_csv(file.path(path,
                                 "SampleAnnotationExample55.csv"))
covariates
  # A tibble: 16 x 14
     OriginalOrder id    title geo_accession TissueDetailed Tissue
             <dbl> <chr> <chr> <chr>         <chr>          <chr>
   1             3 GSM9~ Auti~ GSM946048     Fresh frozen ~ occip~
   2             4 GSM9~ Cont~ GSM946049     Fresh frozen ~ occip~
   3             7 GSM9~ Auti~ GSM946052     Fresh frozen ~ occip~
```

otalleotalleftmargin

```
   4             9 GSM9~ Auti~ GSM946054     Fresh frozen ~ occip~
   5            10 GSM9~ Auti~ GSM946055     Fresh frozen ~ occip~
   6            11 GSM9~ Auti~ GSM946056     Fresh frozen ~ occip~
   7            14 GSM9~ Cont~ GSM946059     Fresh frozen ~ occip~
   8            17 GSM9~ Cont~ GSM946062     Fresh frozen ~ occip~
   9            19 GSM9~ Auti~ GSM946064     Fresh frozen ~ occip~
  10            20 GSM9~ Auti~ GSM946065     Fresh frozen ~ occip~
  11            21 GSM9~ Auti~ GSM946066     Fresh frozen ~ occip~
  12            22 GSM9~ Cont~ GSM946067     Fresh frozen ~ occip~
  13            28 GSM9~ Cont~ GSM946073     Fresh frozen ~ occip~
  14            29 GSM9~ Cont~ GSM946074     Fresh frozen ~ occip~
  15            30 GSM9~ Cont~ GSM946075     Fresh frozen ~ occip~
  16            31 GSM9~ Cont~ GSM946076     Fresh frozen ~ occip~
  # ... with 8 more variables: diseaseStatus <dbl>, Age <dbl>,
  #   PostMortemInterval <dbl>, CauseofDeath <chr>, individual <dbl>,
  #   Female <dbl>, Caucasian <lgl>, FemaleOriginal <lgl>
```

otalleftmargin

inipagefalse

In this case, chronological age is available at `Age` column:

otalleftmargin@ etminipage

```
age <- covariates$Age
head(age)
  [1] 60 39 28 39  8 22
```

otalleftmargin

inipagefalse

The different methylation clocks along with their age accelerated estimates can be simply computed by:

otalleftmargin@ etminipage

```
age.example55 <- DNAmAge(MethylationData, age=age,
                         cell.count=TRUE)
age.example55
  # A tibble: 16 x 17
     id    Horvath ageAcc.Horvath ageAcc2.Horvath ageAcc3.Horvath Hannum Levine
     <fct>   <dbl>          <dbl>           <dbl>           <dbl> <lgl>   <dbl>
   1 GSM9~   51.8          -8.22          -4.45           -4.91   NA      -30.3
   2 GSM9~   39.8           0.754          2.00            1.59   NA      -29.6
   3 GSM9~   26.4          -1.59          -1.67           -1.86   NA      -33.3
   4 GSM9~   34.0          -5.00          -3.76           -0.463 NA      -36.0
   5 GSM9~   10.1           2.06          -0.428           2.82   NA      -52.8
   6 GSM9~   20.4          -1.61          -2.42           -2.88   NA      -42.2
   7 GSM9~    6.00          2.00          -0.971          -0.827 NA      -44.8
   8 GSM9~   34.6           6.65           6.57            5.32   NA      -23.2
   9 GSM9~    7.91          2.91           0.0589         -2.61   NA      -49.8
  10 GSM9~    4.72          2.72          -0.489           1.46   NA      -48.2
  11 GSM9~   29.6          -0.427         -0.268          -1.37   NA      -39.9
  12 GSM9~    1.38          0.375         -2.95           -2.19   NA      -48.3
  13 GSM9~   56.0          -4.01          -0.242           1.62   NA      -26.7
```

otalleftmargin

```
14 GSM9~   24.0            2.03            1.23             -0.669 NA      -39.7
15 GSM9~    9.38           1.38           -1.11             -0.885 NA      -45.4
16 GSM9~   38.8            8.76            8.92              5.85  NA      -27.5
# ... with 10 more variables: ageAcc.Levine <dbl>, ageAcc2.Levine <dbl>,
#   ageAcc3.Levine <dbl>, BNN <dbl>, ageAcc.BNN <dbl>, ageAcc2.BNN <dbl>,
#   ageAcc3.BNN <dbl>, skinHorvath <lgl>, PedBE <lgl>, age <dbl>
```

otalleftmargin inipagefalse

By default, the argument `cell.count` is set equal to TRUE and, hence, can be omitted. This implies that `ageAcc3` will be computed for all clocks. In some occassions this can be very time consuming. In such cases one can simply estimate DNAmAge, accAge and accAge2 by setting `cell.count=FALSE`. NOTE: see section 3.5 to see the reference panels available to estimate cell counts.

Then, we can investigate, for instance, whether the accelerated age is associated with Autism. In that example we will use a non-parametric test (NOTE: use t-test or linear regression for large sample sizes)

otalleftmargin@ etminipage

```
autism <- covariates$diseaseStatus
kruskal.test(age.example55$ageAcc.Horvath ~ autism)

    Kruskal-Wallis rank sum test

  data:  age.example55$ageAcc.Horvath by autism
  Kruskal-Wallis chi-squared = 1.3346, df = 1, p-value = 0.248
kruskal.test(age.example55$ageAcc2.Horvath ~ autism)

    Kruskal-Wallis rank sum test

  data:  age.example55$ageAcc2.Horvath by autism
  Kruskal-Wallis chi-squared = 3.1875, df = 1, p-value = 0.0742
kruskal.test(age.example55$ageAcc3.Horvath ~ autism)

    Kruskal-Wallis rank sum test

  data:  age.example55$ageAcc3.Horvath by autism
  Kruskal-Wallis chi-squared = 2.8235, df = 1, p-value = 0.09289
```

otalleftmargin inipagefalse

4.3 Chronological age prediction using `ExpressionSet` data

One may be interested in assessing association between chronologial age and DNA methylation age or evaluating how well chronological age is predicted by DNAmAge. In order to illustrate this analysis we downloaded data from GEO corresponding to a set of healthy individuals (GEO accession number GSE58045). Data can be retrieved into R by using `GEOquery` package as an `ExpressionSet` object that can be the input of our main function.

otalleftmargin@ etminipage

```
dd <- GEOquery::getGEO("GSE58045")
gse58045 <- dd[[1]]
```

otalleftotalleftmargin inipagefalse

otalleftmargin@ etminipage

```
gse58045
  ExpressionSet (storageMode: lockedEnvironment)
  assayData: 27578 features, 172 samples
    element names: exprs
  protocolData: none
  phenoData
    sampleNames: GSM1399890 GSM1399891 ... GSM1400061 (172 total)
    varLabels: title geo_accession ... twin:ch1 (43 total)
    varMetadata: labelDescription
  featureData
    featureNames: cg00000292 cg00002426 ... cg27665659 (27578 total)
    fvarLabels: ID Name ... ORF (38 total)
    fvarMetadata: Column Description labelDescription
  experimentData: use 'experimentData(object)'
    pubMedIds: 22532803
  Annotation: GPL8490
```

otalleftotalleftmargin inipagefalse

The chronological age is obtained by using `pData` function from `Biobase` package that is able to deal with `ExpressionSet` objects:

otalleftmargin@ etminipage

```
library(Biobase)
pheno <- pData(gse58045)
age <- as.numeric(pheno$`age:ch1`)
```

otalleftotalleftmargin inipagefalse

And the different DNA methylation age estimates are obtained by using `DNAmAge` function (NOTE: as there are missing values, the program automatically runs `impute.knn` function to get complete cases):

otalleftmargin@ etminipage

```
age.gse58045 <- DNAmAge(gse58045, age=age)
  Imputing missing data of the entire matrix ....
  Data imputed. Starting DNAm clock estimation ...
age.gse58045
  # A tibble: 172 x 17
     id    Horvath ageAcc.Horvath ageAcc2.Horvath ageAcc3.Horvath Hannum Levine
     <fct>   <dbl>          <dbl>           <dbl>           <dbl> <lgl>   <dbl>
   1 GSM1~    65.6          1.07             4.58            5.46 NA       50.7
   2 GSM1~    66.3          0.197            4.06            5.06 NA       51.3
```

otalleftotalleftmargin

```
 3 GSM1~    53.9          -5.31           -2.98           -2.42  NA       40.5
 4 GSM1~    40.6          -5.23           -5.89           -6.14  NA       31.3
 5 GSM1~    50.1           0.982           1.06            1.28  NA       41.1
 6 GSM1~    63.7          -0.895           2.64            2.92  NA       48.1
 7 GSM1~    44.7          -0.875          -1.59           -1.76  NA       29.2
 8 GSM1~    59.7          -8.55           -4.20           -3.48  NA       41.0
 9 GSM1~    48.4          -5.84           -4.63           -2.50  NA       43.8
10 GSM1~    59.3          -3.93           -0.719          -0.609 NA       46.1
# ... with 162 more rows, and 10 more variables: ageAcc.Levine <dbl>,
#   ageAcc2.Levine <dbl>, ageAcc3.Levine <dbl>, BNN <dbl>, ageAcc.BNN <dbl>,
#   ageAcc2.BNN <dbl>, ageAcc3.BNN <dbl>, skinHorvath <lgl>, PedBE <lgl>,
#   age <dbl>
```

otalleftotalleftmargin inipagefalse

Figure **??** shows the correlation between DNAmAge obtained from Horvath's method and the chronological age, while Figure **??** depicts the correlation of a new method based on fitting a Bayesian Neural Network to predict DNAmAge based on Horvath's CpGs.

otalleftmargin@ etminipage

```
plotDNAmAge(age.gse58045$Horvath, age)
```

otalleftotalleftmargin inipagefalse

otalleftmargin@ etminipage

```
plotDNAmAge(age.gse58045$BNN, age, tit="Bayesian Neural Network")
```

otalleftotalleftmargin inipagefalse

4.4 Use of DNAmAge in association studies

Let us illustrate how to use DNAmAge information in association studies (e.g case/control, smokers/non-smokers, responders/non-responders, . . .). GEO number GSE58045 contains transcriptomic and epigenomic data of a study in lung cancer. Data can be retrieved into R by

otalleftmargin@ etminipage

```
dd <- GEOquery::getGEO("GSE19711")
gse19711 <- dd[[1]]
```

otalleftotalleftmargin inipagefalse

The object `gse19711`is an `ExpressionSet` that can contains CpGs and phenotypic (e.g clinical) information

otalleftmargin@ etminipage

otalleftotalleftmargin

```
gse19711
```

```
  ExpressionSet (storageMode: lockedEnvironment)
  assayData: 27578 features, 540 samples
    element names: exprs
  protocolData: none
  phenoData
    sampleNames: GSM491937 GSM491938 ... GSM492476 (540 total)
    varLabels: title geo_accession ... stage:ch1 (58 total)
    varMetadata: labelDescription
  featureData
    featureNames: cg00000292 cg00002426 ... cg27665659 (27578 total)
    fvarLabels: ID Name ... ORF (38 total)
    fvarMetadata: Column Description labelDescription
  experimentData: use 'experimentData(object)'
    pubMedIds: 20219944
  Annotation: GPL8490
```

otalleftmargin inipagefalse

Let us imagine we are interested in comparing the accelerated age between cases and controls. Age and case/control status information can be obtained by:

otalleftmargin@ etminipage

```
pheno <- pData(gse19711)
age <- as.numeric(pheno$`ageatrecruitment:ch1`)
disease <- pheno$`sample type:ch1`
table(disease)
  disease
     bi-sulphite converted genomic whole blood DNA from Case
                                                        266
  bi-sulphite converted genomic whole blood DNA from Control
                                                        274
disease[grep("Control", disease)] <- "Control"
disease[grep("Case", disease)] <- "Case"
disease <- factor(disease, levels=c("Control", "Case"))
table(disease)
  disease
  Control    Case
      274     266
```

otalleftmargin inipagefalse

The DNAmAge estimates of different methods is computed by

otalleftmargin@ etminipage

```
age.gse19711 <- DNAmAge(gse19711, age=age)
  Imputing missing data of the entire matrix ....
  Data imputed. Starting DNAm clock estimation ...
```

otalleftmargin inipagefalse

We can observe there are missing data. The funcion automatically impute those using `impute.knn` function from `impute` package since complete cases are required to compute the different methylation clocks. The estimates are:

otalleftmargin@ etminipage

```
age.gse19711
  # A tibble: 540 x 17
     id    Horvath ageAcc.Horvath ageAcc2.Horvath ageAcc3.Horvath Hannum Levine
     <fct>   <dbl>          <dbl>           <dbl>           <dbl> <lgl>   <dbl>
   1 GSM4~    62.9          -5.14          -0.351          -1.10  NA       61.1
   2 GSM4~    68.8         -12.2           -2.85           -2.13  NA       57.0
   3 GSM4~    60.0           3.96           4.54            4.37  NA       43.0
   4 GSM4~    57.9          -4.13          -1.45           -1.38  NA       40.9
   5 GSM4~    59.0         -13.0           -6.79           -6.98  NA       57.0
   6 GSM4~    57.0          -4.00          -1.66           -1.09  NA       44.7
   7 GSM4~    61.9          -3.08           0.657           0.183 NA       47.9
   8 GSM4~    59.1         -11.9           -6.07           -5.53  NA       50.0
   9 GSM4~    60.7         -16.3           -8.33           -9.33  NA       47.7
  10 GSM4~    51.1          -7.93          -6.30           -6.33  NA       52.5
  # ... with 530 more rows, and 10 more variables: ageAcc.Levine <dbl>,
  #   ageAcc2.Levine <dbl>, ageAcc3.Levine <dbl>, BNN <dbl>, ageAcc.BNN <dbl>,
  #   ageAcc2.BNN <dbl>, ageAcc3.BNN <dbl>, skinHorvath <lgl>, PedBE <lgl>,
  #   age <dbl>
```

otalleftotalleftmargin inipagefalse

The association between disease status and DNAmAge estimated using Horvath's method can be computed by

otalleftmargin@ etminipage

```
mod.horvath1 <- glm(disease ~ ageAcc.Horvath ,
                   data=age.gse19711,
                   family="binomial")
summary(mod.horvath1)

  Call:
  glm(formula = disease ~ ageAcc.Horvath, family = "binomial",
      data = age.gse19711)

  Deviance Residuals:
     Min       1Q   Median       3Q      Max
  -1.358   -1.160   -1.030    1.184    1.771

  Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
  (Intercept)    -0.10995    0.09771  -1.125   0.2605
  ageAcc.Horvath -0.02023    0.01154  -1.753   0.0795 .
  ---
  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

  (Dispersion parameter for binomial family taken to be 1)
```

otalleftotalleftmargin

```
      Null deviance: 748.48  on 539  degrees of freedom
  Residual deviance: 745.25  on 538  degrees of freedom
  AIC: 749.25

  Number of Fisher Scoring iterations: 4

mod.horvath2 <- glm(disease ~ ageAcc2.Horvath ,
                   data=age.gse19711,
                   family="binomial")
summary(mod.horvath2)

  Call:
  glm(formula = disease ~ ageAcc2.Horvath, family = "binomial",
      data = age.gse19711)

  Deviance Residuals:
     Min      1Q  Median      3Q     Max
  -1.279  -1.163  -1.082   1.189   1.589

  Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
  (Intercept)     -0.02970    0.08617  -0.345    0.730
  ageAcc2.Horvath -0.01315    0.01209  -1.087    0.277

  (Dispersion parameter for binomial family taken to be 1)

      Null deviance: 748.48  on 539  degrees of freedom
  Residual deviance: 747.27  on 538  degrees of freedom
  AIC: 751.27

  Number of Fisher Scoring iterations: 3

mod.horvath3 <- glm(disease ~ ageAcc3.Horvath ,
                   data=age.gse19711,
                   family="binomial")
summary(mod.horvath3)

  Call:
  glm(formula = disease ~ ageAcc3.Horvath, family = "binomial",
      data = age.gse19711)

  Deviance Residuals:
     Min      1Q  Median      3Q     Max
  -1.338  -1.163  -1.046   1.185   1.771

  Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
  (Intercept)     -0.02993    0.08626  -0.347    0.729
  ageAcc3.Horvath -0.01927    0.01283  -1.502    0.133
```

otalleftalleftmargin

```
  (Dispersion parameter for binomial family taken to be 1)

      Null deviance: 748.48  on 539  degrees of freedom
  Residual deviance: 746.13  on 538  degrees of freedom
  AIC: 750.13

  Number of Fisher Scoring iterations: 4
```

otalleftmargin inipagefalse

We do not observe statistical significant association between age acceleration estimated using Horvath method and the risk of developing lung cancer. It is worth to notice that Horvath's clock was created to predict chronological age and the impact of age acceleration of this clock on disease may be limited. On the other hand, Levine's clock aimed to distinguish risk between same-aged individuals. Let us evaluate whether this age acceleration usin Levine's clock is associated with lung cancer

otalleftmargin@ etminipage

```
mod.levine1 <- glm(disease ~ ageAcc.Levine , data=age.gse19711,
           family="binomial")
summary(mod.levine1)

  Call:
  glm(formula = disease ~ ageAcc.Levine, family = "binomial", data = age.gse19711)

  Deviance Residuals:
     Min      1Q  Median      3Q     Max
  -1.592  -1.149  -0.939   1.174   1.733

  Coefficients:
                Estimate Std. Error z value Pr(>|z|)
  (Intercept)    0.40956    0.17894   2.289  0.02209 *
  ageAcc.Levine  0.03178    0.01133   2.806  0.00502 **
  ---
  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

  (Dispersion parameter for binomial family taken to be 1)

      Null deviance: 748.48  on 539  degrees of freedom
  Residual deviance: 740.17  on 538  degrees of freedom
  AIC: 744.17

  Number of Fisher Scoring iterations: 4

mod.levine2 <- glm(disease ~ ageAcc2.Levine , data=age.gse19711,
           family="binomial")
summary(mod.levine2)

  Call:
  glm(formula = disease ~ ageAcc2.Levine, family = "binomial",
      data = age.gse19711)
```

otalleftmargin

```
  Deviance Residuals:
      Min       1Q   Median       3Q      Max
  -1.7053  -1.1328  -0.8614   1.1529   1.8015

  Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
  (Intercept)    -0.02925    0.08718  -0.336 0.737225
  ageAcc2.Levine  0.04430    0.01234   3.589 0.000332 ***
  ---
  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

  (Dispersion parameter for binomial family taken to be 1)

      Null deviance: 748.48  on 539  degrees of freedom
  Residual deviance: 734.49  on 538  degrees of freedom
  AIC: 738.49

  Number of Fisher Scoring iterations: 4

mod.levine3 <- glm(disease ~ ageAcc3.Levine , data=age.gse19711,
           family="binomial")
summary(mod.levine3)

  Call:
  glm(formula = disease ~ ageAcc3.Levine, family = "binomial",
      data = age.gse19711)

  Deviance Residuals:
     Min      1Q  Median      3Q     Max
  -1.354  -1.161  -1.057   1.187   1.408

  Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
  (Intercept)    -0.02962    0.08622  -0.344    0.731
  ageAcc3.Levine  0.01679    0.01244   1.350    0.177

  (Dispersion parameter for binomial family taken to be 1)

      Null deviance: 748.48  on 539  degrees of freedom
  Residual deviance: 746.62  on 538  degrees of freedom
  AIC: 750.62

  Number of Fisher Scoring iterations: 3
```

otalleotalleftmargin inipagefalse

Here we observe as the risk of developing lung cancer increases 3.23 percent per each unit in the age accelerated variable (`ageAcc`). Similar conclusion is obtained when using `ageAcc2` and `ageAcc3` variables.

In some occasions cell composition should be used to assess association. This information is calculated in `DNAmAge` function and it can be incorporated in the model by:

otalleftmargin@ etminipage

```
cell <- attr(age.gse19711, "cell_proportion")
mod.cell <- glm(disease ~ ageAcc.Levine + cell, data=age.gse19711,
           family="binomial")
summary(mod.cell)

  Call:
  glm(formula = disease ~ ageAcc.Levine + cell, family = "binomial",
      data = age.gse19711)

  Deviance Residuals:
      Min       1Q   Median       3Q      Max
  -1.9605  -1.0832  -0.6241   1.0742   2.3395

  Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
  (Intercept)   -9.768206   4.380382  -2.230 0.025748 *
  ageAcc.Levine  0.003959   0.012208   0.324 0.745746
  cellCD4T      -3.339693   3.833531  -0.871 0.383656
  cellMono      10.165096   4.594096   2.213 0.026922 *
  cellNeu       16.319534   4.584745   3.560 0.000372 ***
  cellNK        -0.882134   4.296498  -0.205 0.837326
  ---
  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

  (Dispersion parameter for binomial family taken to be 1)

      Null deviance: 748.48  on 539  degrees of freedom
  Residual deviance: 686.56  on 534  degrees of freedom
  AIC: 698.56

  Number of Fisher Scoring iterations: 4
```

otalleftmargin inipagefalse

Here we observe as the positive association disapears after adjusting for cell counts.

5 Gestational DNAm Age estimation

Let us start by reproducing the example provided in Knight et al. (2016) as a test data set (file 'TestDataset.csv'). It consists on 3 individuals whose methylation data are available as supplementary data of their paper. The data is also available at `methylclock` package as a data frame.

otalleftmargin@ etminipage

```
TestDataset[1:5,]
       CpGName    Sample1    Sample2    Sample3
  1 cg00000292 0.72546496 0.72350947 0.69023377
  2 cg00002426 0.85091763 0.80077888 0.80385777
```

otalleftmargin

```
  3 cg00003994 0.05125853 0.05943935 0.05559333
  4 cg00005847 0.08775420 0.11722333 0.10845113
  5 cg00006414 0.03982478 0.06146891 0.03491992
```

otalleftmarginotalleftmargin inipagefalse

The Gestational Age (in months) is simply computed by

otalleftmargin@ etminipage

```
ga.test <- DNAmGA(TestDataset)
ga.test
  # A tibble: 3 x 5
    id       Knight Bohlin Mayne Lee
    <fct>     <dbl> <lgl>  <dbl> <lgl>
  1 Sample1    38.2 NA      35.8 NA
  2 Sample2    38.8 NA      36.5 NA
  3 Sample3    40.0 NA      36.6 NA
```

otalleftmarginotalleftmargin inipagefalse

The results are the same as those described in the additional file 7 of Knight et al. (2016) (link here)

Let us continue by illustrating how to compute GA of real examples. The PROGRESS cohort data is available in the additional file 8 of Knight et al. (2016). It is availabel at `methylclock` as a `tibble`:

otalleftmargin@ etminipage

```
progress_data
  # A tibble: 148 x 151
     CpGmarker `784` `1052` `1048` `1017` `956` `1038` `989` `946` `941` `1024`
     <chr>     <dbl>  <dbl>  <dbl>  <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl>
   1 cg000228~ 0.289  0.372  0.347  0.351 0.313  0.300 0.298 0.294 0.322  0.313
   2 cg004662~ 0.658  0.724  0.700  0.717 0.695  0.665 0.710 0.686 0.692  0.704
   3 cg005468~ 0.682  0.711  0.684  0.717 0.627  0.605 0.684 0.716 0.684  0.666
   4 cg005757~ 0.312  0.381  0.300  0.331 0.294  0.348 0.284 0.305 0.319  0.325
   5 cg006893~ 0.566  0.576  0.556  0.571 0.521  0.569 0.599 0.575 0.532  0.564
   6 cg010565~ 0.558  0.620  0.529  0.600 0.577  0.574 0.590 0.576 0.548  0.555
   7 cg011844~ 0.712  0.718  0.667  0.744 0.668  0.676 0.710 0.744 0.685  0.717
   8 cg013480~ 0.195  0.186  0.180  0.194 0.212  0.208 0.183 0.129 0.161  0.144
   9 cg021006~ 0.329  0.330  0.340  0.344 0.268  0.280 0.288 0.314 0.283  0.346
  10 cg028138~ 0.819  0.858  0.832  0.874 0.861  0.830 0.894 0.873 0.895  0.863
  # ... with 138 more rows, and 140 more variables: `1047` <dbl>,
  #   `1035` <dbl>, `988` <dbl>, `939` <dbl>, `936` <dbl>, `748` <dbl>,
  #   `1031` <dbl>, `903` <dbl>, `864` <dbl>, `874` <dbl>, `898` <dbl>,
  #   `1013` <dbl>, `971` <dbl>, `966` <dbl>, `866` <dbl>, `924` <dbl>,
  #   `931` <dbl>, `1007` <dbl>, `954` <dbl>, `958` <dbl>, `1037` <dbl>,
  #   `965` <dbl>, `1008` <dbl>, `1005` <dbl>, `962` <dbl>, `979` <dbl>,
  #   `881` <dbl>, `876` <dbl>, `764` <dbl>, `743` <dbl>, `987` <dbl>,
  #   `930` <dbl>, `1023` <dbl>, `928` <dbl>, `910` <dbl>, `897` <dbl>,
  #   `1036` <dbl>, `904` <dbl>, `769` <dbl>, `907` <dbl>, `821` <dbl>,
  #   `990` <dbl>, `747` <dbl>, `753` <dbl>, `843` <dbl>, `761` <dbl>,
```

otalleftmarginotalleftmargin

```
#   `819` <dbl>, `820` <dbl>, `802` <dbl>, `805` <dbl>, `870` <dbl>,
#   `817` <dbl>, `1040` <dbl>, `815` <dbl>, `952` <dbl>, `974` <dbl>,
#   `951` <dbl>, `929` <dbl>, `980` <dbl>, `911` <dbl>, `927` <dbl>,
#   `914` <dbl>, `841` <dbl>, `912` <dbl>, `969` <dbl>, `754` <dbl>,
#   `1053` <dbl>, `884` <dbl>, `878` <dbl>, `909` <dbl>, `810` <dbl>,
#   `863` <dbl>, `925` <dbl>, `853` <dbl>, `857` <dbl>, `850` <dbl>,
#   `950` <dbl>, `1027` <dbl>, `948` <dbl>, `970` <dbl>, `831` <dbl>,
#   `813` <dbl>, `1051` <dbl>, `913` <dbl>, `1015` <dbl>, `1054` <dbl>,
#   `937` <dbl>, `1006` <dbl>, `940` <dbl>, `827` <dbl>, `791` <dbl>,
#   `991` <dbl>, `839` <dbl>, `818` <dbl>, `828` <dbl>, `774` <dbl>,
#   `845` <dbl>, `797` <dbl>, `998` <dbl>, `767` <dbl>, ...
```

otalleftmargin inipagefalse

This file also contains different variables that are available in this `tibble`. The

otalleftmargin@ etminipage

```
progress_vars
  # A tibble: 150 x 4
     id    birthweight   EGA    acc
     <chr>       <dbl> <dbl>  <dbl>
   1 784          2.62  38    0.792
   2 1052         2.59  38.3 -1.05
   3 1048         3.20  38    2.29
   4 1017         3.28  38.6  0.643
   5 956          2.79  37.1  1.75
   6 1038         2.89  38.1  1.09
   7 989          2.47  38   -0.774
   8 946          2.42  37.7 -2.36
   9 941          2.96  36.7 -3.18
  10 1024         2.61  38.6 -1.12
  # ... with 140 more rows
```

otalleftmargin inipagefalse

Clinical Variables including clinical assesment of gestational age (EGA) are available at this `tibble`

otalleftmargin@ etminipage

```
progress_vars
  # A tibble: 150 x 4
     id    birthweight   EGA    acc
     <chr>       <dbl> <dbl>  <dbl>
   1 784          2.62  38    0.792
   2 1052         2.59  38.3 -1.05
   3 1048         3.20  38    2.29
   4 1017         3.28  38.6  0.643
   5 956          2.79  37.1  1.75
   6 1038         2.89  38.1  1.09
   7 989          2.47  38   -0.774
   8 946          2.42  37.7 -2.36
```

otalleftmargin

```
   9 941          2.96  36.7 -3.18
  10 1024         2.61  38.6 -1.12
  # ... with 140 more rows
```

otalleftmargin inipagefalse

The Gestational Age (in months) is simply computed by

otalleftmargin@ etminipage

```
ga.progress <- DNAmGA(progress_data)
ga.progress
  # A tibble: 150 x 5
     id    Knight Bohlin Mayne Lee
     <fct>  <dbl> <lgl>  <lgl> <lgl>
   1 784     38.8 NA     NA    NA
   2 1052    37.2 NA     NA    NA
   3 1048    40.3 NA     NA    NA
   4 1017    39.2 NA     NA    NA
   5 956     38.9 NA     NA    NA
   6 1038    39.2 NA     NA    NA
   7 989     37.2 NA     NA    NA
   8 946     35.4 NA     NA    NA
   9 941     33.5 NA     NA    NA
  10 1024    37.4 NA     NA    NA
  # ... with 140 more rows
```

otalleftmargin inipagefalse

We can compare these results with the clinical GA available in the variable EGA

otalleftmargin@ etminipage

```
plotDNAmAge(ga.progress$Knight, progress_vars$EGA,
            tit="GA Knight's method",
            clock="GA")
```

otalleftmargin inipagefalse

Figure 3b (only for PROGRESS dataset) in Knight et al. (2016) representing the correlation between GA acceleration and birthweight can be reproduced by

otalleftmargin@ etminipage

```
library(ggplot2)
progress_vars$acc <- ga.progress$Knight - progress_vars$EGA
p <- ggplot(data=progress_vars, aes(x = acc, y = birthweight)) +
  geom_point() +
  geom_smooth(method = "lm", se=FALSE, color="black") +
  xlab("GA acceleration") +
  ylab("Birthweight (kgs.)")
p
```

otalleftmargin

otalleftmargin inipagefalse

Finally, we can also estimate the "accelerated gestational age" using two of the the three different estimates previously described (`accAge`, `accAge2`) by provinding information of gestational age through `age` argument. Notice that in that case `accAge3` cannot be estimates since we do not have all the CpGs required by the default reference panel to estimate cell counts for gestational age which is "andrews and bakulski cord blood".

otalleftmargin@ etminipage

```
accga.progress <- DNAmGA(progress_data,
                         age = progress_vars$EGA,
                         cell.count=FALSE)
accga.progress
  # A tibble: 150 x 8
     id    Knight ageAcc.Knight ageAcc2.Knight Bohlin Mayne Lee     age
     <fct>  <dbl>         <dbl>          <dbl> <lgl>  <lgl> <lgl> <dbl>
   1 784     38.8         0.792          1.27  NA     NA    NA     38  
   2 1052    37.2        -1.05          -0.488 NA     NA    NA     38.3
   3 1048    40.3         2.29           2.77  NA     NA    NA     38  
   4 1017    39.2         0.643          1.28  NA     NA    NA     38.6
   5 956     38.9         1.75           1.99  NA     NA    NA     37.1
   6 1038    39.2         1.09           1.61  NA     NA    NA     38.1
   7 989     37.2        -0.774         -0.292 NA     NA    NA     38  
   8 946     35.4        -2.36          -1.96  NA     NA    NA     37.7
   9 941     33.5        -3.18          -3.06  NA     NA    NA     36.7
  10 1024    37.4        -1.12          -0.486 NA     NA    NA     38.6
  # ... with 140 more rows
```

otalleftmargin inipagefalse

6 Correlation among DNAm clocks

We can compute the correlation among biological clocks using the function `plotCorClocks` that requires the package `ggplot2` and `ggpubr` to be installed in your computer.

We can obtain, for instance, the correlation among the clocks estimated for the healthy individuals study previosuly analyze (GEO accession number GSE58045) by simply executing:

otalleftmargin@ etminipage

```
plotCorClocks(age.gse58045)
```

otalleftmargin inipagefalse

References

Alfonso, Gerardo, and Juan R Gonzalez. 2018. "Bayesian Neural Networks Improve Methylation Age Estimates." *bioRxiv* XX (X): XX.

Bakulski, Kelly M, Jason I Feinberg, Shan V Andrews, Jack Yang, Shannon Brown, Stephanie L. McKenney, Frank Witter, Jeremy Walston, Andrew P Feinberg, and M Daniele Fallin. 2016. "DNA Methylation of Cord Blood Cell Types: Applications for Mixed Cell Birth Studies." *Epigenetics* 11 (5): 354–62.

Bohlin, Jon, Siri Eldevik Håberg, Per Magnus, Sarah E Reese, Håkon K Gjessing, Maria Christine Magnus, Christine Louise Parr, CM Page, Stephanie J London, and Wenche Nystad. 2016. "Prediction of Gestational Age Based on Genome-Wide Differentially Methylated Regions." *Genome Biology* 17 (1): 207.

Chen, Wei, Ting Wang, Maria Pino-Yanes, Erick Forno, Liming Liang, Qi Yan, Donglei Hu, et al. 2017. "An Epigenome-Wide Association Study of Total Serum Ige in Hispanic Children." *Journal of Allergy and Clinical Immunology* 140 (2): 571–77.

Goede, Olivia M de, Hamid R Razzaghian, E Magda Price, Meaghan J Jones, Michael S Kobor, Wendy P Robinson, and Pascal M Lavoie. 2015. "Nucleated Red Blood Cells Impact Dna Methylation and Expression Analyses of Cord Blood Hematopoietic Cells." *Clinical Epigenetics* 7 (1): 95.

Hannum, Gregory, Justin Guinney, Ling Zhao, Li Zhang, Guy Hughes, SriniVas Sadda, Brandy Klotzle, et al. 2013. "Genome-Wide Methylation Profiles Reveal Quantitative Views of Human Aging Rates." *Molecular Cell* 49 (2): 359–67.

Horvath, Steve. 2013. "DNA Methylation Age of Human Tissues and Cell Types." *Genome Biology* 14 (10): 3156.

Horvath, Steve, Junko Oshima, George M Martin, Ake T Lu, Austin Quach, Howard Cohen, Sarah Felton, et al. 2018. "Epigenetic Clock for Skin and Blood Cells Applied to Hutchinson Gilford Progeria Syndrome and Ex Vivo Studies." *Aging (Albany NY)* 10 (7): 1758.

Knight, Anna K, Jeffrey M Craig, Christiane Theda, Marie Bækvad-Hansen, Jonas Bybjerg-Grauholm, Christine S Hansen, Mads V Hollegaard, et al. 2016. "An Epigenetic Clock for Gestational Age at Birth Based on Blood Methylation Data." *Genome Biology* 17 (1): 206.

Lee, Yunsung, Sanaa Choufani, Rosanna Weksberg, Samantha L Wilson, Victor Yuan, Amber Burt, Carmen Marsit, et al. 2019. "Placental Epigenetic Clocks: Estimating Gestational Age Using Placental Dna Methylation Levels." *Aging (Albany NY)* 11 (12): 4238.

Levine, Morgan E, Ake T Lu, Austin Quach, Brian H Chen, Themistocles L Assimes, Stefania Bandinelli, Lifang Hou, et al. 2018. "An Epigenetic Biomarker of Aging for Lifespan and Healthspan." *Aging (Albany NY)* 10 (4): 573.

Mayne, Benjamin T, Shalem Y Leemaqz, Alicia K Smith, James Breen, Claire T Roberts, and Tina Bianco-Miotto. 2017. "Accelerated Placental Aging in Early Onset Preeclampsia Pregnancies Identified by Dna Methylation." *Epigenomics* 9 (3): 279–89.

McEwen, Lisa M, Kieran J O?Donnell, Megan G McGill, Rachel D Edgar, Meaghan J Jones, Julia L MacIsaac, David Tse Shen Lin, et al. 2019. "The Pedbe Clock Accurately Estimates Dna Methylation Age in Pediatric Buccal Cells." *Proceedings of the National Academy of Sciences*, 201820843.

Min, JL, G Hemani, G Davey Smith, C Relton, M Suderman, and John Hancock. 2018. "Meffil: Efficient Normalization and Analysis of Very Large Dna Methylation Datasets." *Bioinformatics*.

Reinius, Lovisa E, Nathalie Acevedo, Maaike Joerink, Göran Pershagen, Sven-Erik Dahlén, Dario Greco, Cilla Söderhäll, Annika Scheynius, and Juha Kere. 2012. "Differential Dna Methylation in Purified Human Blood Cells: Implications for Cell Lineage and Studies on Disease Susceptibility." *PloS One* 7 (7): e41361.

Slieker, Roderick C, Steffan D Bos, Jelle J Goeman, Judith VMG Bovée, Rudolf P Talens, Ruud van der Breggen, H Eka D Suchiman, et al. 2013. "Identification and Systematic Annotation of Tissue-Specific Differentially Methylated Regions Using the Illumina 450k Array." *Epigenetics & Chromatin* 6 (1): 26.

Teschendorff, Andrew E, Francesco Marabita, Matthias Lechner, Thomas Bartlett, Jesper Tegner, David Gomez-Cabrero, and Stephan Beck. 2012. "A Beta-Mixture Quantile Normalization Method for Correcting Probe Design Bias in Illumina Infinium 450 K Dna Methylation Data." *Bioinformatics* 29 (2): 189–96.

Wang, Ting, Weihua Guan, Jerome Lin, Nadia Boutaoui, Glorisa Canino, Jianhua Luo, Juan Carlos Celedón, and Wei Chen. 2015. "A Systematic Study of Normalization Methods for Infinium 450K Methylation Data Using Whole-Genome Bisulfite Sequencing Data." *Epigenetics* 10 (7): 662–69.