UNIVERSITY OF CALIFORNIA

Los Angeles

Design and Implementation of a Web Interface with Epigenetic Pacemaker Model (EPM) for analyzing Epigenetic Data

A thesis submitted in partial fulfillment

of the requirements for the degree

Master of Science in Bioinformatics

By

Wenxi Zhang

2022

ABSTRACT OF THE THESIS

Design and Implementation of a Web Interface
with Epigenetic Pacemaker Model (EPM)
for analyzing Epigenetic Data

by

Wenxi Zhang
Master of Science in Bioinformatics
University of California, Los Angeles, 2022
Professor Matteo Pellegrini, Chair

DNA methylation patterns change with increasing age, therefore contributing to some age related diseases. Epigenetic drift over time might result in measurable differences between biological and chronological age, so epigenetic changes are shown to be reflective of an individual's lifestyle. Consequently, the estimation of the epigenetic state can help with medical and biological research, and may act as functional biomarkers of disease before clinical threshold is reached. A variety of epigenetic clocks have been constructed with regression models to approach age estimation from individuals' DNA methylation patterns. This study involves the use of the Epigenetic Pacemaker Model (EPM), an implementation of a fast conditional expectation maximization algorithm to model non-

linear epigenetic trait associations directly without transformation of the phenotype of interest. EPM models the initial methylation value as well as the rate of change at each locus, therefore allows an intuitive interpretation of the selected sites. Since EPM is computationally heavy, it will be useful to build a succinct model for the users who would like to use EPM in their studies. Therefore, the aim of the project is to make EPM more accessible for users to analyze their epigenetic data, at the same time conserving the accuracy in the epigenetic data prediction. The project is primarily based on python code, and uses the R shiny to develop a website app in order for users from other places to upload their data and implement the EPM graph displaying the logarithmic trend of the relationship between epigenetic state and chronological age.

The thesis of Wenxi Zhang is approved.

Jasmine Zhou

Noah Zaitlen

Mary Sehl

Matteo Pellegrini, Committee Chair

University of California, Los Angeles

2022

To my parents, and my dear friends.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

As humans age, a decay of cellular structures occurs. DNA methylation patterns change with age, and it modifies physiological phenotypes, thus may contribute to some age-related diseases (Bocklandt et al. 2011). DNA methylation is the biological process which has a transfer of a methyl group onto the C5 position of the cytosine to form 5-methylcytosine (Bell et al. 2019). Methylation can change the accessibility of DNA segments without changing the sequence. The covalent attachment of a methyl group to the cytosine is catalyzed by de novo or maintenance methyltransferase, which primarily target the CpG sites. As stem cells differentiate, they will have a distinctive DNA methylation pattern, due to the differential activation of enhancers (Robert et al. 2016). Once an organism reaches its adult stage, the cell type and their expected epigenomes are largely determined. Nonetheless, epigenetic changes continue to occur as an organism ages. The widely accepted definition of epigenetic is "a stably heritable phenotype resulting from changes in a chromosome without alterations in the DNA sequence" (Berger et al. 2009). Thus, the continuous change in epigenetics indicates that as humans age, epigenetic alterations happen because of the changes in DNA methylation through multiple distinct and intersecting age-related mechanisms. Therefore, there might be a certain association between epigenetic modifications and one's age, and the investigation

of the relationship between epigenetic state and chronological age might be helpful for medical research (Farrell et al.).

In order to model the relationship of estimating age and human chronological age (phenotype), various epigenetic clocks have been constructed. Two models, the Horvath and the Hannum model, estimate epigenetic age (Sagi et. al 2019). The Horvath model has a logarithmic transformation for samples younger than 20 years old, and displays a linear relationship for samples older than 20 years old. It uses elastic net regression and can very accurately predict the epigenetic age. The Hannum model, on the other hand, makes the assumption that chronological age and epigenetic age are linearly related for all ages. It uses muli-variate regression model. These two models can both predict the epigenetic age. The Epigenetic Pacemaker Model, abbreviated as EPM, is the model that is used in this project. The python implementation of Epigenetic Pacemaker is an algorithm for accurately estimating the epigenetic state, and can be used to study nonlinear epigenetic aging. Unlike the Horvath model, the EPM has no restriction on sample age divisions (Farrell et al., 2020), so it can identify logarithmic patterns for the association between epigenetic state and age throughout the lifespan. As such, using the EPM to estimate epigenetic age is more convenient than the Horvath model by having the log trend over the life span instead of having different trends for different age groups as Horvath did. Compared to the Hannum model which assumes a linear relationship, EPM is more accurate in predicting the epigenetic age based on the chronological age with a logarithmic relationship. In short, the EPM model seems to be an ideal one in epigenetic age estimation.

In this work, we are interested in the development of computational approaches to interpret epigenetic data by using the EPM model. The focus of the paper is on developing

tools for analyzing the bisulfite sequencing data for measuring DNA methylation. Our goal is to develop an interactive web interface which allows user input files and user defined Pearson Correlation Coefficient (PCC) values, and outputs the corresponding EPM plots based on the calculations. The future implications of our project is that we can use this model to test if the EPM patterns vary in different countries, so that we can explore the influence of ethnicity on one's epigenetic age pattern.

CHAPTER 2

BACKGROUND

2.1 Epigenetic Pacemaker Model (EPM)

2.1.1 Model Description

The EPM was firstly introduced as an extension of the Universal Pacemaker formalism (UPM), in which the evolutionary rate of genes remains constant relative to one another, while at the same time the absolute rate could potentially change according to factors affecting the evolving lineage (Snir et al., 2012). Unlike classical regression based approaches, the EPM does not assume a linear relationship between phenotype and the epigenetic state, therefore becoming an ideal model in modeling the non-linear epigenetic trait associations without phenotypic transformation (Snir et al., 2019). The EPM package is available at https://pypi.org/project/EpigeneticPacemaker/ under the MIT license.

2.1.2 Model Formulation

We started with a dataset which has the methylation sites indexed by i, and sample numbers indexed by j to represent each of the individuals in the dataset. A single methylation site can be described as the following formula:

$$\widehat{m_{ij}} = m_i^0 + r_i s_j + \varepsilon_{ij}$$

in which $\widehat{m_{ij}}$ is the observed methylation value, m_i^0 means the initial methylation value. r_i shows the rate of change, and s_j is the epigenetic state, which is the parameter we are interested in. The last term ε_{ij} shows a normally distributed error term, which we would like it to be as small as possible. The goal of the EPM algorithm is to find the optimal values of m_i^0, r_i, and s_j to minimize the error term between the predicted and observed methylation values across a system of methylation sites.

This optimization is accomplished by a fast conditional expectation maximization algorithm, which maximizes the model likelihood by minimizing the residual sum of square error. Each methylation site is assigned an independent rate of change and starting methylation value when fitting the EPM, and users will provide an initial epigenetic state for each individual. The epigenetic state will then be updated by iterating through the EPM to minimize the error across the observed epigenetic state. After that we will get a series of values corresponding to each individual's chronological age and epigenetic state, therefore we can plot the graphs displaying their logarithmic relationship.

2.2 Site Selection Methods in Preprocessing and Model Fitting

Since the EPM model is computationally heavy, selecting informative loci in the preprocessing step is necessary. In this project, we selected sites based on the Pearson Correlation Coefficient (PCC) method. Building EPM models with sites of high PCC values indicates that these sites are more informative in phenotype traits, compared to those with lower PCC values. The sample data has the first row being the sample names, the second row being the chronological age (phenotype age, denoted as p) of each sample,

and the following rows being each DNA methylation data for each site, which indicates the epigenetic age of each individual (denoted as m). To calculate the Pearson Correlation Coefficient, we compute the mean of p and m, and then calculate the transformed phenotype age and transformed methylation age by p - mean(p) and m-mean(m).

$$transformed_{phenotype} = p - \mu p$$

$$transformed_{methylation} = m - \mu m$$

$$Covariance = transformed_{phenotype} * transformed_{methylation}$$

$$Variance_p = \sqrt{\Sigma(transformed_{phenotype})}$$

$$Variance_m = \sqrt{\Sigma(transformed_{methylation})}$$

$$Correlation = Covariance/(Variance_p * Variance_m)$$

After that, we obtain the covariance of p and m by multiplying the two transformed ages, and get the variance of p and m by taking the square root of the sum of each transformed ages. The PCC value can then be calculated by dividing the covariance by the variance of p and m. Currently, we set the PCC value as a predefined value of 0.85 directly in our python code. But later we will remove this predefined value and allow for the users to determine which correlation value to use on their own.

CHAPTER 3

METHODS

3.1 Shiny Application Design

3.1.1 Reticulate and Virtual Environment

Because the pre-existing EPM package is written in python (Colin et al., 2020), and the goal of this study is to build an interactive web app for users, we will use python and R shiny interactively in this project to reach our goal. R Shiny is an R package that can be used to plot data and build interactive web apps straight from R. It can also be extended to other languages, as in this research, which uses python scripts of the Epigenetic Pacemaker. With the goal of encoding python codes inside the R shiny program, a virtual environment is required to build such an environment in which the R shiny program can read python code directly. Each virtual environment will have its own directory, as well as its own Python binary and its own installed Python packages. With the creation of the virtual environment, we can use Most of the underlying concepts of data structures in R and Python are very similar to each other, we also include a package called “reticulate” so as to use python code inside R by using `Sys.setenv(RETICULATE_PYTHON = PATH)` and `RETICULATE_PYTHON`, given the path being the singlecell server of UCLA.

3.1.2 R and Python: Shiny server implementation

The construction of R shiny can be divided into two primary parts: the user interface object (ui), which contains the code for the display on the R shiny web page, and the server function, in which functions of plots, tables, and formulas in R and python are correlated. The user interface object includes code for the output being displayed in the shiny window, and the server function consists of the functions used for the user interface object, and also it is the site for the connection between R shiny and python code in this project.

3.2 Model Modification

3.2.1 The Train-Train Model

The Train-Train Model is the EPM model I build for single file input. This can be considered a simplified model extended from the original EPM model. Unlike the original model which requires two datasets input, this model only takes in one file and can display the output after fitting EPM. This model is designed for individual users who have smaller datasets.

3.2.2 The Train-Test Model

The Train-Test Model is the EPM model I build for multi-file input, usually two files. This is similar to the original EPM approach being displayed on the website https://pypi.org/project/EpigeneticPacemaker/ which also takes in a training file and a testing file for training and refitting the data. However, the original model only takes in the .gz files and has a different data format as our definition. In my implementation, users

are allowed to put in .txt, .csv, and .gz or .tsv files, which are file types that are more likely to be used in individual projects with smaller datasets.

3.3 Data Acquisition and Approximation

From the Gene Expression Omnibus (GEO) repository, and Illumina (Series GSE40279) was used for studying applications of the EPM. Samples that contain too many missing values or outliers were dropped from the dataset. It contains n = 656 samples and 473034 cpg sites after data preprocessing. In the data preprocessing step, individuals or sites with too many missing values (i.e. when >80% of the total value is missing) are being dropped from the dataset. And then the model will use three equations to find the best-fit trendline for the data: the square root function, the linear regression, and the logarithmic regression. Detailed functions are described in Appendix C for the code of Python. We will determine which function has the best performance in displaying the trend of the data.

3.4 Application Layout

3.4.1 Page Design

The Shiny package is a web application framework for individuals to use with programming language R. One of our aims in designing this Shiny app is to declutter the layout so that users could understand the logic and structure, thus could get interaction with the application in a quick and easy way. For better understanding, the layout of each page in the application followed the same pattern.

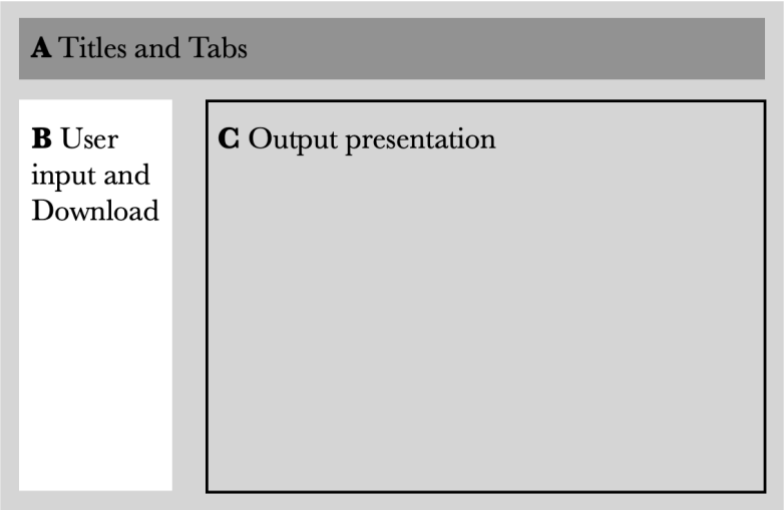

Figure 1 Shiny Application Layout. (A) Titles and Tab information. (B) Primary User Input area. (C) Primary Output area for plots and tables.

As what is displayed in Figure 1, The arrangement of my Shiny app always has a title and tabs to differentiate different sections (Part A). The left panel (Part B) below the header region is called a sidebar panel, which is primarily used for user input as well as file download, while the right panel (Part C), also called the main panel, contains the section that displays a variety of outputs depending on the user selection.

3.4.2 Data Upload

In our designed shiny application, the first widget in the sidebar panel is the box for user file input. This box enables the user to choose a dataset with the required format, including comma-separated values (.csv/.txt) and tab-separated values (.tsv/.txt), to be processed by the application from any location in their local file system. However, if the users want to upload a new file, they might need to re-entering the application to renew an EPM fitting process.

Sample IDs	ID1	ID2	ID3	...
Age	Age1	Age2	Age3	...
site1	value1	value2	value3	...
site2	value1	value2	value3	...
site3	value1	value2	value3	...

Table 1. Proper Formatting of User Upload Dataset. The first row (pink) should be the Sample IDs representing the individuals, the second row (yellow) should be the actual phenotypical age of each individuals, and the following rows should be the sites ID (blue) and corresponding methylation values (grey).

The application requires the users to upload their files conform to our required format. More specifically, none of the cells in the first and second line, as well as in the first column, should be empty for a valid EPM modeling. These lines are the key information for the model to identify individual IDs and ages, as well as the epigenetic sites. Aside from those values, the data in the middle can be stated as NA to represent missing data. Notes that there should not be empty lines at the end of the file. However, just as a notification, a dataset with too many NAs might affect the final result of EPM. An example of the desired file format is shown (Table 1). Detailed information may be illustrated in the EPM Shiny User Guide listed in Appendix A. Users can also use the sample data in the Sample File Download section to get familiar with this web interface.

3.4.3 User Defined PCC values

As stated in the background, Pearson Correlation Coefficient is important in filtering data for EPM to produce a better fitting model. In the shiny application, users have to enter a value between 0 - 1 as a valid PCC value for the EPM to use. The higher value they enter, the more sites will be filtered out and less sites will be left in the resulting data.

3.4.4 Displayed the Output

After the user successfully uploads the dataset and defines a valid PCC value, the application processes the file to extract desired information from the dataset, and the output will be displayed in the main panel. The time for the output to appear on the screen depends on the size of the dataset. With larger datasets, meaning more individuals and more cpg sites to be processed in the EPM model, it will take longer for the output to appear on the application. The presentation of the data is divided into the following tabs: Plot, which displays the dots and fitting lines; Summary, which displays the number of sites after PCC filtration and number of individuals in the dataset; Table1 and Table 2, which displays the detailed information about the selected sites after PCC filtration.

3.4.5 Output for Download

The shiny web interface is capable of producing picture files as well as the tables produced. Users can just click the “Download” button on the left panel to download the table results of EPM, and just drag the plot in the main panel to the desktop to save the figure.

3.5 Application Testing and Validation

As stated before, our aim for this project is to produce a useful and functional application for users to use EPM. Ideally, the program should be able to gracefully handle incorrect user inputs. The actual test files used are provided as the supplementary materials in Appendix D at the end of this document. In brief, the application is tested with files of the following formats:

i. With no data present in the document;

ii. With Row elements being the wrong order as stated in the desired format;

iii. With columns and rows being transposed as stated in the desired format.

Each of the formats that contain potential error listed above is tested in our model by using a comma-separated values (.csv) file, a tab-separated values (.tsv) file, and a semicolon-separated values (.txt) file. By verifying the testing datasets under these formatting options when uploaded successfully and produced the desired output, the shiny application was then used to analyze real dataset obtained from https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi by choosing different numbers in the GEO accession tab. Other than that, I also used the dataset obtained from my fellows in the lab, which is appended as *dogmeth.csv* at the end of this paper.

CHAPTER 4

RESULTS

4.1 Sample File Download and Data Visualization

4.1.1 Sample for Train-Train Model

Since the Train-Train Model only takes in one file, we have included a sample file on the page “Sample for Train-Train” which can be accessed under the tab “Before Starting”. The main panel also displays the sample result for this dataset. A sample output for PCC = 0.85 is shown in Figure 2a. From our implementation, we can see that the EPM model performs a logarithmic trend between epigenetic state and chronological age. This indicates that EPM is a good non-linear model in predicting the epigenetic age according to one’s chronological age.

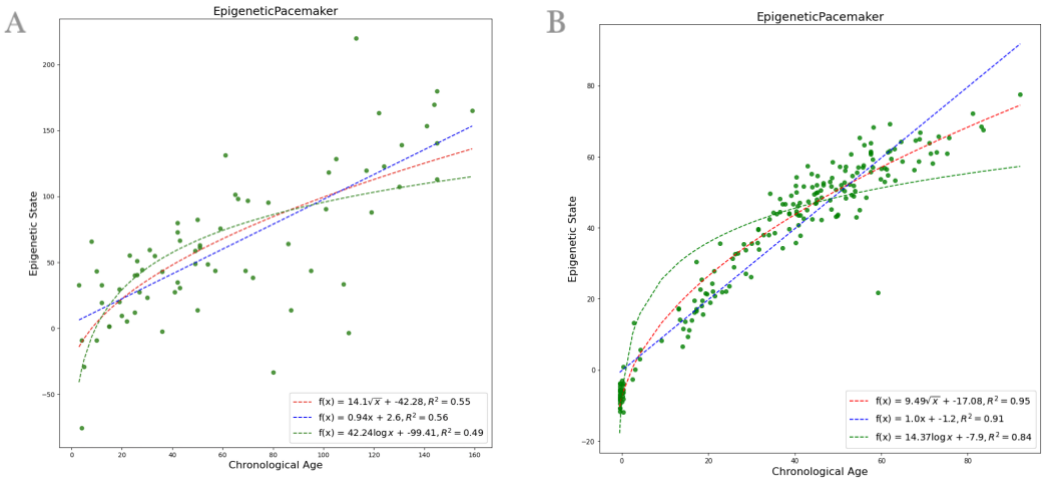

Figure 2. Sample Output for EPM model with sample dataset. (a) For the Train-Train model. (b) For the Train-Test model.

4.1.2 Sample for Train-Test Model

Since the Train-Test Model takes in two files, we have included two sample files on the page “Sample for Train-Train” which can be accessed under the tab “Before Starting”. The main panel also displays the sample result for this dataset. A sample output for PCC = 0.85 is shown in Figure 2b. We can see that since the dataset has more individuals in their original dataset, there are more points being displayed after PCC filtration.

4.2 Data Visualization

Other than the sample file being downloaded, we use two other datasets, called the dog_methlation.txt (12MB) and sample100.txt (3.6GB) which will be attached in the appendix. The result will be displayed as the following:

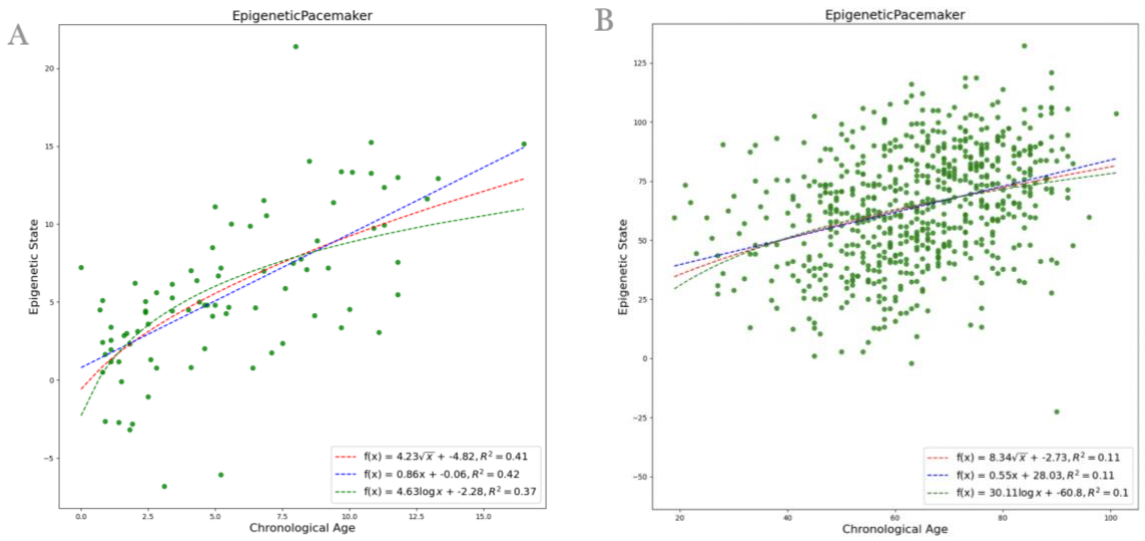

Figure 3: Output for real dataset from (a) dog methylation value and (b) human methylation value.

The datasets are being modeled by three functions as described in Section 3.3. Through a variety of testing, we found that the square root seems to be the best-fit line in all three functions, displayed as the red line. The result might not be very obvious in

this real dataset output, but the trend is clear in our sample data output in Section 4.1.2 (Fig 2b).

4.3 PCC value Effect to the Output

By analyzing the same dataset but only changing the default correlation value in the site selection step, we found that with higher PCC correlation value, the points in the graph are less scattered than those with lower PCC value.

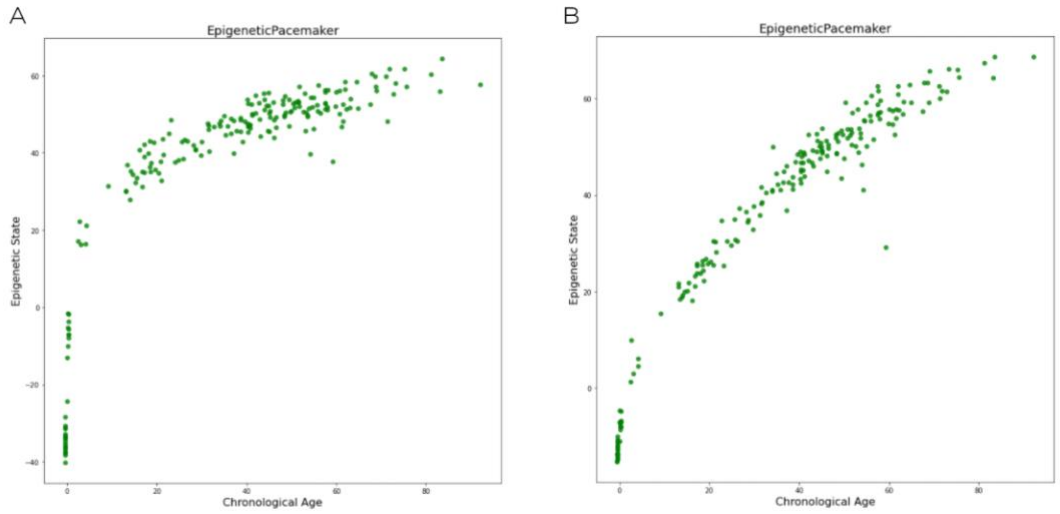

Figure 4: Exploration on the effect of Pearson Correlation Coefficient in site selection. (A) Select sites with PCC value > 0.6. (B) Select sites with PCC value > 0.85. The logarithmic trend seems better, and the points are more aggregated around the trend line.

The results in figure 2 are expected since higher PCC value means at this site, the epigenetic state accurately predicts the chronological age, whereas lower PCC value means the site has a poor prediction. Therefore, as we set the PCC minimum value to be higher, sites with lower correlation between epigenetic state and chronological age will be filtered out, consequently the sites left in our plot will concentrate closer on the logarithmic trendline.

CHAPTER 5

DISCUSSION

In our implementation of the web interface, we faced a lot of challenges but finally got a usable web app for other users to use. This web interface will be publicized in the future, and will soon allow users to upload files on their own computers, and get the resulting EPM plot. EPM is an ideal algorithm for epigenetic age prediction based on a biological sample alone, therefore it will provide insightful information in forensic science (Bocklandt et al. 2011). Furthermore, by measuring the relevant sites in the genome, researchers can predict the risk of age-related diseases in routine medical screening, and thus they can tailor interventions based on the epigenetic bio-age instead of the chronological age (Bocklandt et al. 2011). For example, recently, researchers have found that the change of the correlation between chronological age and epigenetic state has been stably associated with pathological states, and this usually occurs long before the reveal of clinical signs of diseases (Di Lena et al., 2021). Under this circumstance, the epigenetic pacemaker model will be a potential choice for disease diagnosis and forensic applications. The application of the epigenetic pacemaker model in estimating the non-linear pattern of change over time for epigenetic age has also been explored in a study on hibernation (Pinho et al., 2021). The result proves that the epigenetic pacemaker model performs better in modeling the dynamics of methylation across the genome, while the other

epigenetic clocks can only estimate the age of samples according to the sum of their methylation values (Pinho et al., 2021). All these evidence shows that the EPM is a good selection for users to estimate the non-linear relationship between actual age and methylation values.

In the future, we will continuously work on optimizing our implementation so that individuals can have a better user experience. We hope this web interface can benefit more researchers and save their time for getting the resulting plot.

APPENDIX

6.1 Appendix A: EPM User Guide

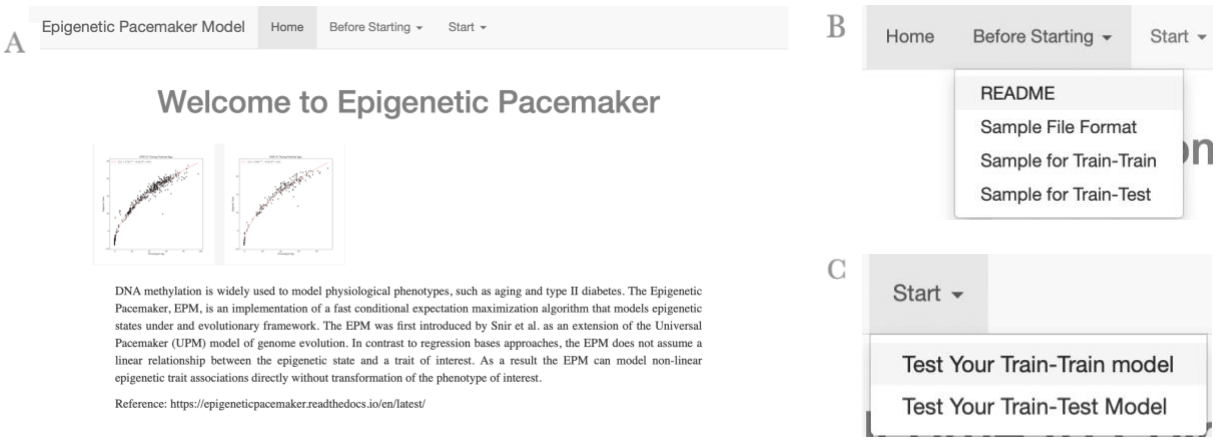

Figure s5: Output of R shiny web interface. (A) The welcome page of the web interface. (B) The tab "Before Starting" for sample files selection and sample output. (C) The tab "Start" for user input and their own dataset output.

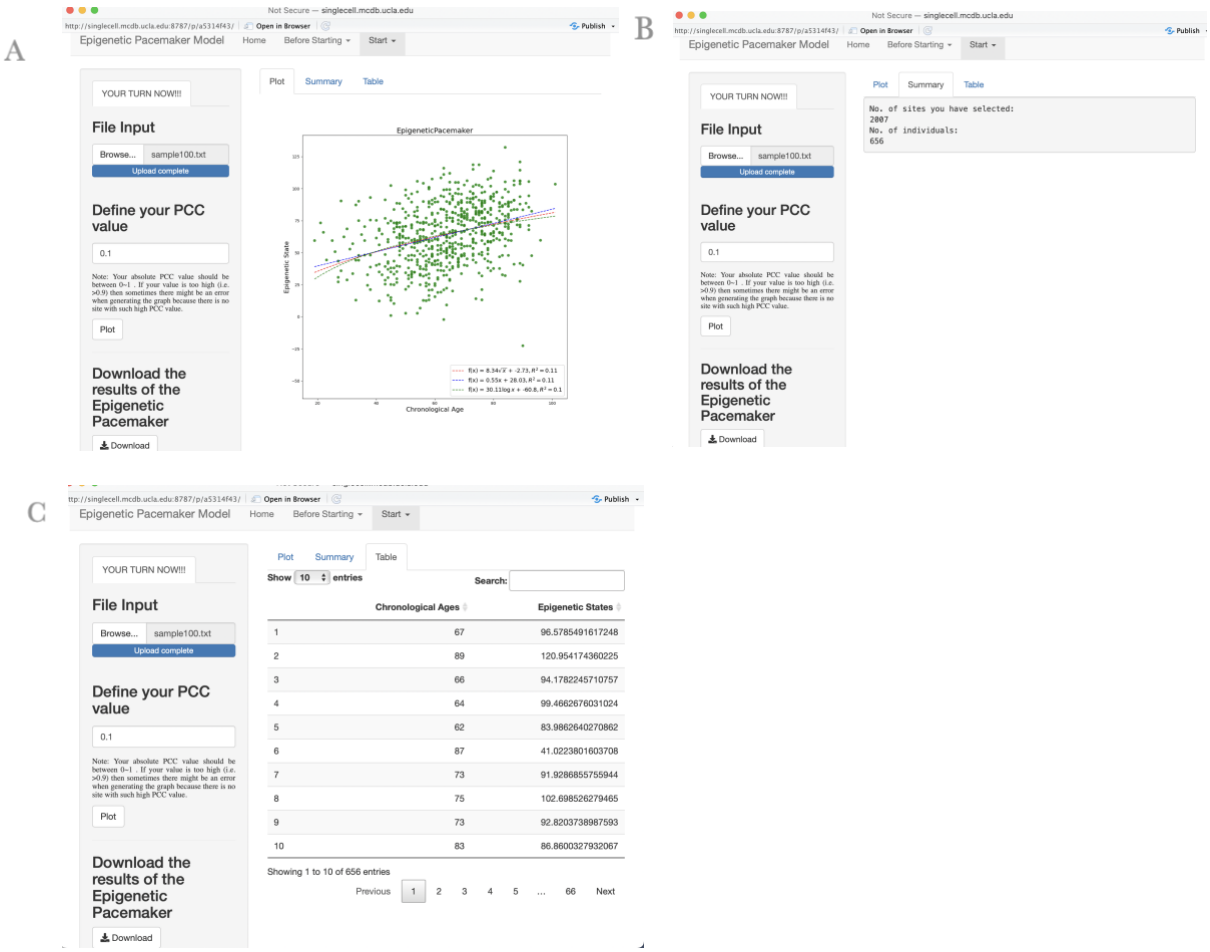

Figure s6: Output of EPM (a) EPM output plot (b) EPM output summary of selected sites (c) Detailed Information about selected sites.

6.2 Appendix B: R Code

```
Sys.setenv(RETICULATE_PYTHON = ".epmv/bin/python3")
library(reticulate)
library(ggplot2)
library(DT)

#py_install('matplotlib',pip=TRUE)
#py_module_available('numpy')
source_python("data_get.py")
source_python("epi_fortest.py")
### SHINY UI ###
ui <- fluidPage(
  # Application title
  navbarPage(title="Epigenetic Pacemaker Model",id="mainpage",
    tabPanel("Home",value="tabhome",
      h1("Welcome to Epigenetic Pacemaker",
      style="color:grey;front-size:35px;text-align:center;"),
      br(),
      fluidRow(
        column(4,offset=1,align="center",
          fluidRow(
            plotOutput("home_fig",height="20%")
          )
        )
      ),
      br(),
      p("DNA methylation is widely used to model physiological phenotypes,
        such as aging and type II diabetes. The Epigenetic Pacemaker, EPM,
        is an implementation of a fast conditional expectation maximization algorithm that models
epigenetic
        states under and evolutionary framework. The EPM was first introduced by Snir et al.
        as an extension of the Universal Pacemaker (UPM) model of genome evolution.
        In contrast to regression bases approaches, the EPM does not assume a linear relationship
        between the epigenetic state and a trait of interest. As a result the EPM can model non-linear
        epigenetic trait associations directly without transformation of the phenotype of interest.",
        style = "text-align:justify;font-family:'times'; font-si16pt; width:80%; margin-left:10%;
margin-right:10%",
        align="center"),
      p("Reference: https://epigeneticpacemaker.readthedocs.io/en/latest/",
        style = "font-family:'times'; font-si16pt; width:80%; margin-left:10%; margin-right:10%")
      ),
    navbarMenu("Before Starting",
            tabPanel("README",value="sample_download",
                  h3("How to Start",
```

```
                  style="color:grey;front-size:35px;text-align:center;"),
                br(),
                p(strong("Home")," Page has the description for general EPM model.
                  More info could be found at:",
                  tags$a(href="https://epigeneticpacemaker.readthedocs.io/en/latest/",
"https://epigeneticpacemaker.readthedocs.io/en/latest/"),
                  style = "font-family:'times';
                  font-si16pt; width:80%; margin-left:10%; margin-right:10%"
                 ),
                p(strong("Before starting,"), "users could find the desired format for input files,
                  shown in the", strong("Sample File Format section."),
                  "The first line should be the sample IDs.
                  The second line should be the age of the sites,
                  and the following rows being different sites and their values.
                  Notes that there should not be empty lines at the end of the file.
                  Users can also use the sample data in", strong("Sample File Download"),
                  "section to get familiar with this web interface.",
                  style = "font-family:'times'; font-si16pt; width:80%; margin-left:10%; margin-
right:10%"),
                p("After getting used to this web interface,
                  users can start uploading their own files in the", strong("Test Your Dataset section"),
                  "under the", strong("Start"),"tab.
                  A file should be uploaded and PCC value should be defined for a successful plot.
                  When users upload their files and enter a PCC value and click the",strong("Plot"),
"button,
                  the main panel will show the EPM output for their values.
                  The Summary section contains how many sites have been selected based on the
PCC value,
                  and the number of individuals after selection.
                  Users can find the information for the selected sites under Table,
                  and download the results of Epigenetic Pacemaker when they click the",strong
("Download")," button.",
                  style = "font-family:'times'; font-si16pt; width:80%; margin-left:10%; margin-
right:10%"),

          ),
     tabPanel("Sample File Format",value="sample_file",
          sidebarPanel(h3("Before uploading your file..."),
                  hr(),
                  h5("NOTE: This sample on the right shows the format of the file you should
                    upload to the test_dataset part (The first line with Col1, Col2...
                    is not required in your file. For a correct output,
                    your file should have exactly the same format
                    starting from the row with Sample IDs.",
                    style = "text-align:justify;font-family:'times'")
```

```
                    ),
            mainPanel(dataTableOutput("sample_table"))
      ),
      tabPanel("Sample for Train-Train",value="sample_download",
            sidebarPanel(h3("Download the sample dataset for the train-train model"),
                  a(href="methages.txt", "Download Sample Data", download=NA,
target="_blank"),
                  h5("This is the sample file to start getting familiar with EPM.
                    The plot on the right is an expected output for each dataset.")
            ),
            mainPanel(
              plotOutput("trainsample_fig")
            )
      ),
      tabPanel("Sample for Train-Test",
            sidebarPanel(h3("Download the sample dataset for the train-test model"),
                  a(href="GSE74193_train.tsv.gz", "Download Sample Training Data",
download=NA, target="_blank"),
                  a(href="GSE74193_test.tsv.gz", "Download Sample Testing Data",
download=NA, target="_blank"),
                  h5("Sample training and testing dataset for getting familiar with the model")
            ),
            mainPanel(
              plotOutput("traintest_fig")
            )
      )

    ),

    navbarMenu("Start",
          tabPanel("Test Your Train-Train model",value="test_dataset",
                sidebarLayout(
                  sidebarPanel(
                    tabsetPanel(id='tabset',
                          tabPanel('YOUR TURN NOW!!!',
                                fileInput("test_file",label=h3("File Input")),

                                textInput('num_input',
                                      h3('Define your PCC value'),
                                      value="Enter text..."),
                                h6('Note: Your absolute PCC value should be between 0~1 .
                              If your value is too high (i.e. >0.9) then sometimes there might be an error
                              when generating the graph because there is no site with such high PCC
value.',
                                    style = "text-align:justify;font-family:'times'")
```

```
                )
            ),
            actionButton('go','Plot'),
            hr(),
            h3("Download the results of EPM"),
            downloadButton("downloadData","Download_age"),
            downloadButton("downloadSites","Download_sites")
          ),
          mainPanel(
            tabsetPanel(
              tabPanel('Plot',plotOutput("test_plot")),
              tabPanel('Summary',verbatimTextOutput("selected_sites")),
              tabPanel('Table1',dataTableOutput("table")),
              tabPanel('Table2',dataTableOutput("table_sites"))
            )
          )
        )
 ),
tabPanel("Test Your Train-Test Model",value="traintest_data",
         sidebarLayout(
           sidebarPanel(
             tabsetPanel(id = 'tabset',
                         tabPanel("Your Turn Now",
                                  fileInput("training_file",label = h3("Training file Input")),
                                  fileInput("testing_file",label = h3("Testing file Input")),
                                  textInput('num_input2',
                                            h3('Define your PCC value'),
                                            value="Enter text..."),
                         )
             ),
             actionButton('go2','Plot_TrainTestModel'),
             hr(),
             h3("Download the results"),
             downloadButton("downloadTrainTestData","Download_age_traintest"),
           ),
           mainPanel(
             tabsetPanel(
               tabPanel('Plot',plotOutput("traintest_plot")),
               tabPanel('Summary',verbatimTextOutput("selected_sites_traintest")),
               tabPanel('Table1',dataTableOutput("table_traintest"))
               #tabPanel('Table2')
             )
           )
         )
```

```
          )
    )
  )
)

# Define server logic required to draw a histogram
server <- function(input, output) {
  options(shiny.maxRequestSize = 5000*1024^2)
  output$prediction <- renderPlot({
    plot(test_ages, test_predict, main="EpigeneticPacemaker",
         xlab="Chronological Age",ylab="Epigenetic State",pch=19)
    abline(lm(log(test_predict)~log(test_ages)),col="red")
  })

  output$sample_table<-renderDataTable({
    df <- rbind(c("Sample IDs","ID1","ID2","ID3","..."),c("Age","Age1","Age2","Age3","..."),
                c("site1","value1","value2","value3","..."),c("site2","value1","value2","value3","..."),
                c("site3","value1","value2","value3","..."),c("site...","...","...","...","..."))
    colnames(df) <- c('Col1','Col2',"Col3","Col4","...")
    df
  })

  output$home_fig <- renderImage({
    list(src="mainplot.jpg",contentType="image/png",width="100%")
  },
  deleteFile = FALSE)

  output$traintest_fig <- renderImage({
    list(src="traintest_plot.png",contentType="image/png",width="100%")
  },
  deleteFile = FALSE)

  output$trainsample_fig <- renderImage({
    list(src="sample_data_output.png",contentType="image/png",width="100%")
  },
  deleteFile = FALSE)

  # ------------------------------------------------------------------------------
  # For Train-train model
  # ------------------------------------------------------------------------------
  pcc_val <- reactiveValues(doPlot=FALSE)
  observeEvent(input$go,{
    pcc_val$doPlot <- input$go
  })
```

```
output$test_plot <- renderImage({
  if(pcc_val$doPlot==FALSE)
    return()
  isolate({
    req(input$test_file)
    tryCatch({
      filename <- (input$test_file)$name
      pcc <- as.numeric(input$num_input)
    },
    error = function(e) {
      stop(safeError(e))
    })
    r <- plot_testdata(filename,pcc)
    list(src='myplot.png',contentType='image/png',width="100%")
  })
},
deleteFile=TRUE
)

output$selected_sites <- renderText({
  if(pcc_val$doPlot==FALSE)
    return("Please enter the PCC value")
  isolate({
    filename <- (input$test_file)$name
    pcc <- as.numeric(input$num_input)
    r <- plot_testdata(filename,pcc)
    paste("No. of sites you have selected:",r[[1]],
          "No. of individuals:",r[[2]], sep='\n')
  })
})

output$table <- renderDataTable({
  filename <- (input$test_file)$name
  pcc <- as.numeric(input$num_input)
  r <- plot_testdata(filename,pcc)
  df <- data.frame(r[[4]],r[[5]])
  colnames(df) <- c('Chronological Ages','Epigenetic States')
  df
})

output$table_sites <- renderDataTable({
  filename <- (input$test_file)$name
  pcc <- as.numeric(input$num_input)
  r <- plot_testdata(filename,pcc)
  df <- data.frame(r[[3]])
```

```
  colnames(df) <- c('CPG Sites')
  df
})

output$downloadSites <- downloadHandler(
  filename = function(){
    paste(input$test_file$name,".csv",sep = "")
  },
  content = function(file){
    filename <- (input$test_file)$name
    pcc <- as.numeric(input$num_input)
    r <- plot_testdata(filename,pcc)
    df <- data.frame(r[[3]])
    colnames(df) <- c('CPG Sites')
    write.csv(df, file, row.names = FALSE)
  }
)

# Downloadable csv of selected dataset ----
output$downloadData <- downloadHandler(
  filename = function() {
    paste(input$test_file$name, ".csv", sep = "")
  },
  content = function(file) {
    filename <- (input$test_file)$name
    pcc <- as.numeric(input$num_input)
    r <- plot_testdata(filename,pcc)
    df <- data.frame(r[[4]],r[[5]])
    colnames(df) <- c('Chronological Ages','Epigenetic States')
    write.csv(df, file, row.names = FALSE)
  }
)

# ------------------------------------------------------------------------
# For Train-Test model
# ------------------------------------------------------------------------
pcc_val2 <- reactiveValues(doPlot=FALSE)
observeEvent(input$go2,{
  pcc_val2$doPlot <- input$go2
})

output$traintest_plot <- renderImage({
  if(pcc_val2$doPlot==FALSE)
    return()
  isolate({
```

```
    req(input$training_file)
    req(input$testing_file)
    tryCatch({
      train_filename <- (input$training_file)$name
      test_filename <- (input$testing_file)$name
      pcc_value <- as.numeric(input$num_input2)
      message(train_filename,test_filename,pcc_value,typeof(pcc_value))
    },
    error = function(e) {
      stop(safeError(e))
    })
    r <- plot_traintestdata(train_filename,test_filename,pcc_value)
    list(src='myplot.png',contentType='image/png',width="100%")
  })
},
deleteFile=TRUE
)

output$selected_sites_traintest <- renderText({
  if(pcc_val2$doPlot==FALSE)
    return("Please enter the PCC value")
  isolate({
    train_filename <- (input$training_file)$name
    test_filename <- (input$testing_file)$name
    pcc_value <- as.numeric(input$num_input2)
    r <- plot_traintestdata(train_filename,test_filename,pcc_value)
    paste("No. of sites you have selected:",r[[1]],
          "No. of individuals:",r[[2]], sep='\n')
  })
})

output$table_traintest <- renderDataTable({
  train_filename <- (input$training_file)$name
  test_filename <- (input$testing_file)$name
  pcc_value <- as.numeric(input$num_input2)
  r <- plot_traintestdata(train_filename,test_filename,pcc_value)
  df <- data.frame(r[[3]],r[[4]])
  colnames(df) <- c('Chronological Ages','Epigenetic States')
  df
})

output$downloadTrainTestData <- downloadHandler(
  filename = function() {
    paste(input$testing_file$name, ".csv", sep = "")
  },
```

```
    content = function(file) {
      train_filename <- (input$training_file)$name
      test_filename <- (input$testing_file)$name
      pcc_value <- as.numeric(input$num_input2)
      r <- plot_traintestdata(train_filename,test_filename,pcc_value)
      df <- data.frame(r[[3]],r[[4]])
      colnames(df) <- c('Chronological Ages','Epigenetic States')
      write.csv(df, file, row.names = FALSE)
    }
  )

}

# ---------------------------------------------------------------------------------
# Run the application
# ---------------------------------------------------------------------------------
shinyApp(ui = ui, server = server)
```

6.3 Appendix C: Python Code

```
# ---------------------------------------------------------------------------------
# Python code for testing and plotting
# ---------------------------------------------------------------------------------
import numpy as np
import matplotlib.pyplot as plt
from EpigeneticPacemaker.EpigeneticPacemaker import EpigeneticPacemaker
from matplotlib import rc
from scipy import optimize
from data_get import get_testdata
import scipy.stats as stats
from matplotlib.offsetbox import AnchoredText
from matplotlib.patches import BoxStyle
import matplotlib.patches as mpatches

def pearson_correlation(meth_matrix: np.array, phenotype: np.array) -> np.array:
    """calculate pearson correlation coefficient between rows of input matrix and phenotype"""
    # calculate mean for each row and phenotype mean
    matrix_means = np.nanmean(meth_matrix, axis=1)
    phenotype_mean = np.mean(phenotype)

    # subtract means from observed values
    # https://stackoverflow.com/questions/18691084/what-does-1-mean-in-numpy-reshape
    transformed_matrix = meth_matrix - matrix_means.reshape([-1,1])
    transformed_phenotype = phenotype - phenotype_mean
```

```
    # calculate covariance
    covariance = np.nansum(transformed_matrix * transformed_phenotype, axis=1)
    transformed_matrix = np.delete(transformed_matrix,np.where(covariance==0),axis=0)
    covariance = covariance[covariance!=0]
    variance_meth = np.sqrt(np.nansum(transformed_matrix ** 2, axis=1))
    variance_phenotype = np.sqrt(np.sum(transformed_phenotype ** 2))
    return covariance / (variance_meth * variance_phenotype)

def r2(x,y):
    # return r squared
    return stats.pearsonr(x,y)[0] **2

# Plot Prediction: known_ages
def plot_prediction(sites, known_ages, predicted_ages, label = None):
    # define optimization function
    def func(x, a, b):
        return a * np.asarray(x)**0.5 + b
    def func_lin(x, a, b):
        return a * np.asarray(x) + b
    def func_log(x, a, b):
        return a * np.log(x) + b

    # fit trend line
    popt, pcov = optimize.curve_fit(func, [1 + x for x in known_ages], predicted_ages)
    popt_lin, pcov_lin = optimize.curve_fit(func_lin, [1 + x for x in known_ages], predicted_ages)
    popt_log, pcov_log = optimize.curve_fit(func_log, [1 + x for x in known_ages], predicted_ages)

    # get r squared
    rsquared = r2(predicted_ages, func([1 + x for x in known_ages], *popt))
    rsquared_lin = r2(predicted_ages, func_lin([1 + x for x in known_ages], *popt_lin))
    rsquared_log = r2(predicted_ages, func_log([1 + x for x in known_ages], *popt_log))

    # initialize plt plot
    fig, ax = plt.subplots(figsize=(12,12))

    # scatter plot
    plt.scatter(known_ages, predicted_ages, marker='o', alpha=0.8, color='g')

    # formula
    eqn = 'f(x) = '+str(round(popt[0],2)) + r'$\sqrt{x}$' ' + ' + str(round(popt[1],2))+r'$,
R^{2}=$'+str(round(rsquared,2))
    eqn_lin = 'f(x) = '+str(round(popt_lin[0],2))+'x' ' + '+str(round(popt_lin[1],2))+r'$,
R^{2}=$'+str(round(rsquared_lin,2))
```

```
    eqn_log = 'f(x) = '+str(round(popt_log[0],2))+r'$\log{x}$' ' + '+str(round(popt_log[1],2))+r'$, R^{2}=$'+str(round(rsquared_log,2))

    # display lines and legends
    line_eqn, = plt.plot(sorted(known_ages), func(sorted([1 + x for x in known_ages]), *popt), 'r--', label = eqn)
    line_eqn_lin, = plt.plot(sorted(known_ages), func_lin(sorted([1 + x for x in known_ages]), *popt_lin), 'b--', label = eqn_lin)
    line_eqn_log, = plt.plot(sorted(known_ages), func_log(sorted([1 + x for x in known_ages]), *popt_log), 'g--', label = eqn_log)
    legend_for_lines = ax.legend(handles=[line_eqn,line_eqn_lin,line_eqn_log], loc='lower right',fontsize = 14)
    ax.add_artist(legend_for_lines)

   # plt.scatter(predicted_ages, sites, marker='o', alpha=0.8, color='r')
    plt.title(label, fontsize=18)
    plt.xlabel('Chronological Age', fontsize=16)
    plt.ylabel('Epigenetic State', fontsize=16)
    plt.savefig('myplot.png')

# plot the train-train model
def plot_testdata(file,pcc):
    # retrieve the training and testing data
    test_data = get_testdata(file)
    # unpack the training and testing data
    test_samples, test_cpg_sites, test_ages, test_methylation_values = test_data
    #train_samples, train_cpg_sites, train_ages, train_methylation_values = train_data
    abs_pcc_coefficients = abs(pearson_correlation(test_methylation_values, test_ages))

    # site selection based on user defined pcc values
    testing_sites = np.where(abs_pcc_coefficients > pcc)[0]
    # filter the name of the selected sites
    result_cpg_names = []
    for i in testing_sites:
        result_cpg_names.append(test_cpg_sites[i])

    # initialize the EPM model
    epm = EpigeneticPacemaker(iter_limit=100, error_tolerance=0.00001)

    # fit the model using the data
    epm.fit(test_methylation_values[testing_sites,:], test_ages)

    # generate predicted ages using the test data
    test_sites = test_methylation_values[testing_sites,:]
    test_predict = epm.predict(test_sites)
```

```
    # plot the model results
    plot_prediction(test_sites, test_ages, test_predict, "EpigeneticPacemaker")

    num_sites = len(test_sites)
    num_predicts = len(test_predict)
    return num_sites,num_predicts,result_cpg_names,test_ages,test_predict
# plot the train-test model
def plot_traintestdata(trainfile,testfile,pcc):
    # retrieve the training and testing data
    train_data,test_data = get_testtrain_data(trainfile,testfile)
    # unpack the training and testing data
    train_samples, train_cpg_sites, train_ages, train_methylation_values = train_data
    test_samples, test_cpg_sites, test_ages, test_methylation_values = test_data

    abs_pcc_coefficients = abs(pearson_correlation(train_methylation_values, train_ages))
    # site selection based on user defined pcc values
    training_sites = np.where(abs_pcc_coefficients > pcc)[0]

    # initialize the EPM model
    epm = EpigeneticPacemaker(iter_limit=100, error_tolerance=0.00001)

    # fit the model using the data
    epm.fit(train_methylation_values[training_sites,:], train_ages)

    # generate predicted ages using the test data
    test_sites = test_methylation_values[training_sites,:]
    test_predict = epm.predict(test_sites)

    # plot the model results
    plot_prediction(test_sites, test_ages, test_predict, "EpigeneticPacemaker")

    num_sites = len(test_sites)
    num_predicts = len(test_predict)

    return num_sites,num_predicts,test_ages,test_predict

# ------------------------------------------------------------------------------
# Python code for reading and formatting data
# ------------------------------------------------------------------------------
import gzip
import io
import numpy as np
import os, sys
```

```
def convert_to_float(line):
    converted_line = []
    for value in line:
        try:
            converted_line.append(float(value))
        except ValueError:
            converted_line.append(value)
    return converted_line

def load_data_set(file_path):
    formatted_data = []
    with io.BufferedReader(gzip.open(file_path, 'rb')) as data:
        for line in data:
            formatted_data.append(convert_to_float(line.decode('utf-8').strip().split('\t')))
    sample_names = formatted_data[0]
    cpg_sites = [line[0] for line in formatted_data[1:-1]]
    phenotypes = np.array(formatted_data[-1][1:])
    methylation_values = np.array([line[1:] for line in formatted_data[1:-1]])
    return sample_names, cpg_sites, phenotypes, methylation_values

def load_testdata(file_path):
    formatted_data = []
    with open(file_path,'r') as testfile:
        for line in testfile:
            formatted_data.append(convert_to_float(line.strip().split('\t')))
    sample_names = formatted_data[0]
    cpg_sites = [line[0] for line in formatted_data[2:]]
    phenotypes = np.array(formatted_data[1][1:])
    methylation_values = np.array([line[1:] for line in formatted_data[2:-1]])
    bad_samples = np.where(sum(np.isnan(methylation_values))>1000)[0]
    methylation_values = np.delete(methylation_values, bad_samples, axis = 1)
    methylation_values = methylation_values[~np.isnan(methylation_values).any(axis=1)]
    phenotypes = np.delete(phenotypes, bad_samples)
    return sample_names, cpg_sites, phenotypes, methylation_values

def get_testtrain_data(trainfile,testfile):
    train_data = load_data_set(trainfile)
    test_data = load_data_set(testfile)
    return train_data, test_data

def get_testdata(file):
    test_data = load_testdata(file)
    return test_data
```

6.4 Appendix D: Dataset being used for Testing

Sample	31001811416722	31019101703602	31019101705914	31019101706567	31019101705884	31019101711421
Age	23	122	116	113	15	25
1	0	0	0.083333333	0.151515152	0	0
2	0.920273349	0.926746167	0.924357035	0.877532228	0.889816361	0.84991274
3	0.071823204	0.088541667	0.111111111	0.416666667	0	0
4	0	0.0234375	0	0.088709677	0	0
5	0.055555556	0.163636364	0.085106383	0.056338028	0.071428571	0.060240964
6	0.20668693	0.168674699	0.082417582	0.230142566	0.108949416	0.067285383
7	0.246987952	0.297202797	0.225490196	0.322751323	0.218181818	0.178217822
8	0.173784978	0.232608696	0.122137405	0.171818182	0.126506024	0.114634146
9	0.007653061	0.023379384	0.125944584	0.118619698	0.025693731	0.010834236
10	0.013157895	0.048484848	0.135	0.161931818	0.024	0.015317287
11	1	1	1	1	1	1
12	0	0	0.272727273	0.454545455	0.5	NaN
13	0	0.714285714	0.785714286	0.5	0.6	0.285714286
14	0	0	0.083333333	0	0	0
15	0	0.057142857	0.112903226	0.125	0	0.081967213
16	0.15	0.068181818	0.315789474	0.333333333	0.097560976	NaN
17	NaN	0	NaN	1	1	0
18	0	0.12	0.066666667	0.076923077	0	0.068965517
19	0.074074074	0.181818182	0.166666667	0.34	0.083333333	0.125
20	0.235294118	0.433333333	0.4	0.3	0.545454545	0.619047619
21	0.012295082	0.013392857	0.005524862	0.011406844	0.011673152	0.011267606
22	NaN	1	0.75	0	1	0
23	NaN	0.725	0.666666667	0.923076923	1	0.965517241
24	1	NaN	1	NaN	NaN	NaN
25	0.75	1	NaN	0.6	0.5	NaN
26	0.774193548	0.898550725	0.813559322	0.684848485	0.782608696	0.755102041
27	0.748663102	0.745762712	0.700421941	0.575280899	0.721568627	0.587458746
28	0.84	0.861952862	0.811267606	0.705	0.800524934	0.704
29	0.730769231	0.769230769	0.655172414	0.685393258	0.6	0.577777778
30	1	0.2	0	0.272727273	0	0.333333333
31	NaN	NaN	NaN	NaN	NaN	NaN
32	0.2	0.235294118	0.285714286	0.217391304	0.6	0.5
33	0.181818182	0.267605634	0.456521739	0.362416107	0.139534884	0.266666667
34	0.444444444	0.727272727	0.75	0.565217391	0.6	0.5
35	0.571428571	0.652173913	0.765957447	0.516666667	0.53125	0.594594595
36	0.580645161	0.652173913	0.769230769	0.409836066	0.4	0.621621622
37	0.512820513	0.722222222	0.477272727	0.363636364	0.621212121	0.422222222
38	0.811594203	0.833333333	0.96	0.644628099	0.888888889	0.797752809
39	0.064516129	0.019607843	0.028571429	0.034482759	0.032258065	0
40	0.014084507	0.039473684	0.076576577	0.070967742	0.02484472	0.025751073

Table s2a. Sample file for Train-Train Model (first 40 rows).

	GSM1913856	GSM1913707	GSM1913799	GSM1913920	GSM1914045
cg04359978	0.6691261558796890	0.755580376581236	0.6884263372648490	0.6360069709266910	0.6042327749819150
cg26003967	0.182535651334174	0.11177993818461600	0.17622902773743400	0.16008643673011	0.183330425497636
cg02288964	0.35923646718219300	0.602980957636736	0.402486410101552	0.30536168811118	0.392255987244429
cg04972348	0.6259602702620650	0.5045143117704340	0.6708358493267 55	0.618341960802162	0.591601205412541
cg06618740	0.643175488636574	0.5015163114286360	0.6320253599786720	0.593328681814888	0.632180850202221
cg23400617	0.08958452720384100	0.0650417618691987	0.0945257158465603	0.10902858729671500	0.11851936023493300
cg19237691	0.30749769018077500	0.20907663859722900	0.388848580432226	0.296402918233516	0.32275090230231400
cg05638471	0.657382467413806	0.769357683901547	0.618843468024678	0.62919695549747	0.5509659740007810
cg19945840	0.434572177164881	0.40413643152342	0.4593042352879210	0.44408936267849500	0.441677062385617
cg00825734	0.545341170174377	0.680640442417516	0.52457668125372	0.545854296461464	0.539800703247162
cg21265287	0.704544735486582	0.87953397768263	0.72070284360508	0.677603376246936	0.6634085988950970
cg18163062	0.61369902763438 20	0.756787238041279	0.6504537156458540	0.6664258146379290	0.6021447971904130
cg18565510	0.535960153353472	0.735658707053701	0.523317854173614	0.5152430579947040	0.544335761637666
cg04098194	0.071498651421798	0.197991145638874	0.0810366916818975	0.0702801246935857	0.0725902698110503
cg00002593	0.796243158387644	0.844075607550058	0.8009839549891170	0.803999295094053	0.7707933649478050
cg13932473	0.786389552456 08	0.862130760538024	0.7915153013740500	0.795971205219374	0.7057057141738950
cg04481810	0.783402086228587	0.830323174534259	0.7612736284846250	0.769318413749293	0.76893815223711
cg09064061	0.635865640865332	0.6514620038360490	0.6130189924870700	0.6403827566545440	0.621360786401312
cg15081886	0.572755715402842	0.784044548786123	0.575577430629191	0.47740819221258800	0.5520989625332260
cg20429981	0.516747140592943	0.702554478697411	0.5259373122546440	0.530737681526932	0.499315152874 38
cg23678594	0.444447335753941	0.403192290044525 00	0.46078366178982500	0.46743832664949100	0.49604696164329
cg24163210	0.28676866786233200	0.34060380545871500	0.304426198856736	0.24990401593886900	0.241748646701299
cg14417954	0.32871917553474600	0.218912534149745	0.32774942706460500	0.30554712310647	0.298369735859299
cg05929117	0.37774626392803200	0.336228890399545	0.37258302658113100	0.37492554102766300	0.4484057181352 9800
cg01970427	0.49691533135423800	0.619247616418771	0.463122503102109	0.48896475207172500	0.4389309196177 2400
cg25845463	0.486948285696118	0.5554243774172230	0.43882973820031900	0.382795560713493	0.364496119349449
cg07836663	0.490830405627768	0.7914700697153770	0.585707083965583	0.5411374642024870	0.5175833102907860
cg24776480	0.672843084848056	0.790256814887713	0.71002318774727	0.663345058117641	0.6448580991421750
cg07842327	0.496043592795251	0.7707137042712760	0.495829382884429 00	0.4692602819687 66	0.511794505151685
cg19770723	0.38377750199353300	0.5444043102860810	0.406174987126006	0.3896763166995220	0.360460808847501 00
cg07114422	0.40299013771944900	0.586742286498577	0.500453726823322	0.4068044483283 49	0.4499385474334 0600
cg00618291	0.6602480251995260	0.8130533407641960	0.702476213245328	0.664712700198674	0.61338840142312
cg04619852	0.537006247463337	0.739521716674496	0.539835713477548	0.5658332324341 72	0.5457322087170 87
cg21405929	0.48540875176568200	0.7478775371317210	0.5071184529677480	0.509113170741161	0.4971758843678 64
cg27054514	0.681798099842339	0.7351290704249080	0.6618763111234180	0.672385895234648	0.5242514006661 88
cg11142617	0.365681980880197	0.59620115117856	0.41756186966936400	0.433573100980786	0.4157964708474 8700
cg09024606	0.496304560826699	0.7040850500570520	0.527816552307135	0.493076117273108	0.4893384487880 79
cg25154306	0.854403428019342	0.8459988059420410	0.8468106565249320	0.8181438549897280	0.8038326351840920
cg12721952	0.5961665683637310	0.715248786833882	0.628063784751933	0.605919097924301	0.5695017517754 85
cg04716261	0.40091930448085700	0.293341753593013	0.426495554170519	0.398422126458476	0.4947348413237 69

Table s2b. Sample Dataset for Testing in Train-Test Model (first 40 rows).

	GSM1914004	GSM1913907	GSM1914006	GSM1914270	GSM1913710	GSM1914143
cg04359978	0.652869585730449	0.6842519468968210	0.6897300997948868	0.634749302506281	0.741135956323165	0.6591375995988190
cg26003967	0.15827264308502700	0.193034157811251 00	0.209855883018396 00	0.18195680288139600	0.13242296785758700	0.177622625324618
cg02288964	0.3532870926797270	0.406465511937033 00	0.395162003601807	0.2877066485186 88	0.6073649088664 06	0.3720216482551 4200
cg04972348	0.6668337373522230	0.6040842861652090	0.6348261781756280	0.7277295592392720	0.5044474057479 76	0.6533439148755 47
cg06618740	0.648571420492469	0.6414785526186370	0.691188386099965	0.7250159149559190	0.4967956798607 6700	0.6843825686489660
cg23400617	0.1252220028176218	0.0977979171093558	0.119084452918578	0.116336786838216	0.0902197337630865	0.1052971868308 91
cg19237691	0.37025279362175100	0.322519979859651	0.359928094333184	0.3543072476362 89	0.2152393231838 7200	0.3612620914897 1
cg05638471	0.586706040017141	0.6252155061893 63	0.5712390526839160	0.5017214062187480	0.7621789395342600	0.5815095880818 83
cg19945840	0.382184971540705	0.4387232569888 7	0.4090174646448 07	0.4442505210579 8300	0.3922007216896 23	0.4753361376504 75
cg00825734	0.517571788926795	0.5478140193880 62	0.5263056510417 65	0.5018905237256 63	0.6910733119410 64	0.5615663846042 88
cg21265287	0.6617374631569 95	0.7364393794192 29	0.7487306302422 580	0.7177995679903 78	0.8430600403372 22	0.7277951887548 510
cg18163062	0.6433523252835 66	0.6648707735482 13	0.6418417955454 03	0.6202926527382 63	0.8150204019053 390	0.6503934719814 07
cg18565510	0.5436337482683 88	0.5931981234769 79	0.5753497132261 15	0.5388978878474 720	0.7411309174327 98	0.5087588540747 270
cg04098194	0.0728895518040313	0.0780792675729 2800	0.0835726958587 9200	0.0517449146314852	0.202995180470 6	0.0667047953259 31
cg00002593	0.7899248860157 05	0.7766667431366 01	0.8022377021068 580	0.7486112379834 33	0.7938689429893 550	0.7710284420439 830
cg13932473	0.7573678785266	0.7749012436008 730	0.7894354115409	0.6902996409248 41	0.8652407973842 420	0.7682209339983 77
cg04481810	0.7392580596426200	0.8048735908760 3	0.7866538737967 100	0.7173269652743 8	0.8607578710726 380	0.7137902474058 140
cg09064061	0.6094117877533560	0.6350912314194 64	0.6132793396094 47	0.5827398915838 55	0.6943399531350 5	0.6347893329018 550
cg15081886	0.5085297406517 75	0.6399348230606 15	0.5475479828663 06	0.4779871075725 21	0.8187087795155 5	0.4945569632945 2200
cg20429981	0.5318824436609 11	0.5655364654229 91	0.5715308857826 78	0.5070213648576	0.7528760882405 780	0.5272560707488 3
cg23678594	0.4794148301528 84	0.4408206182457 8300	0.5028788586382 15	0.5065124690690 34	0.3906203091300 1900	0.4900424909138 8900
cg24163210	0.2490186905797 18	0.2706135007246 2300	0.2661741481857 45	0.2355572925572 19	0.3255756940606 84	0.2377830556306 5800
cg14417954	0.3618042845121 74	0.3098941033523 8700	0.3574431749301 54	0.3727984486783 5600	0.1738356907331 78	0.3836073520289 71
cg05929117	0.3934415372027 6500	0.4118003757715 820	0.4335490663269 7000	0.4230584507599 09	0.2663779553225 44	0.4505315270451 0300
cg01970427	0.4837824801397 4900	0.4754905343269 21	0.4491860141870 72	0.4287955081396 69	0.6464054611787 14	0.4486577424060 79
cg25845463	0.3705704212344 5500	0.3881827139507 0900	0.4089231802145 1900	0.3304360201933 78	0.5968673894419 880	0.3624348016168 61
cg07836663	0.5149196735015 870	0.6083121903906 850	0.5343925597575 490	0.5600945305369 450	0.8015647402270 73	0.5355443829693 36
cg24776480	0.671291671059801	0.7079158311447 05	0.6481574603836 96	0.6181567695200 66	0.7985617181966 620	0.6469692963424 400
cg07842327	0.5227922404552 96	0.5904241833265 620	0.5299440151719 11	0.4854440144554 4500	0.7958340057876 65	0.5030286060404 760
cg19770723	0.3732084547593 13	0.4320826274381 5800	0.3887881940672 8600	0.3695494779006 83	0.5588733236810 22	0.3803512810837 06
cg07114422	0.4281942565027 8200	0.4810569103436 09	0.4321621244403 7800	0.4307199437070 0400	0.5743404623438 84	0.4311483677587 57
cg00618291	0.6323025857529 18	0.6975347406780 44	0.6266686893383 81	0.6476989446508 19	0.7823276518465 07	0.6855037654671 813
cg04619852	0.5667754937946 96	0.5977071288467 27	0.5643112334475 54	0.5501205266398 6	0.7350537157647 740	0.5696423035691 17
cg21405929	0.4872031896277 4600	0.5678296798851 41	0.5295552872247 84	0.5088283150214 33	0.6743558325749 23	0.5480596549857 04
cg27054514	0.5735285323404 630	0.6603807656087 62	0.5623615647610 47	0.6109326601683 96	0.7351213424894 33	0.6287886539232 05
cg11142617	0.3876800676652 970	0.4219480206932 19	0.4176459957033 6800	0.3657990089555 56000	0.5929622907964 72	0.4057391586096 74
cg09024606	0.5009412193055 280	0.5348333735185 04	0.4774818170808 430	0.4736754917077 9200	0.7269735851030 15	0.4654629229030 09
cg25154306	0.8219898839639 3	0.8182032711632 020	0.7903179207859 190	0.7610771562338 07	0.8822483154045 260	0.7761431792597 34
cg12721952	0.5622574804816 92	0.6035348975159 42	0.5565451815572 68	0.5223578724551 97	0.7001691766735 22	0.5864649596633 44
cg04716261	0.3891462263910 170	0.4364653556312 7200	0.4682389000527 94	0.4897671686947 54	0.2225090103755 8000	0.4858081667971 0900

Table s2c. Sample Dataset for Training in the Train-Test Model (first 40 rows).

sampleid	31019101710432	31201050607833	31201050607826	31201050607819	31201050607671	31201050607687
age	5.5	6.8	12.9	5.6	1.1	1.1
31564916	0.047619048	0	0.074626866	0	0	0
59504758	0.922865014	0.930715935	0.91609589	0.935727788	0.874418605	0.93452381
34847430	0.171390013	0.152963671	0.160997732	0.149597238	0.098290598	0.020671835
34847860	0.07530648	0.03125	0.073529412	0.098214286	0.032679739	0.003174603
42897202	0.043478261	0.083333333	0.030075188	0.025423729	0.050251256	0.036649215
4905186	0.126909518	0.171189979	0.153103448	0.136144578	0.056179775	0.143330572
13552690	0.161389173	0.184745763	0.29326288	0.237096774	0.151131222	0.182247403
13552810	0.096330275	0.122418879	0.174193548	0.150116369	0.10430322	0.112078978
39568448	0.032172471	0.018195603	0.057585825	0.018518519	0.012573345	0.023320158
39568328	0.051987768	0.020033389	0.046767538	0.027855153	0.014285714	0.026058632
34029898	0.855263158	0.923076923	0.943396226	0.9	0.901098901	0.919191919
44351369	0.554545455	0.318181818	0.491803279	0.528571429	0.25	0.344444444
14488524	0.658088235	0.77037037	0.734375	0.793103448	0.614285714	0.720379147
22045013	0.054773083	0.030959752	0.023041475	0.032520325	0.012718601	0.050632911
64563007	0.082004556	0.07523511	0.10944206	0.057208238	0.00927357	0.079726651
44642464	0.076923077	0.054054054	0.196363636	0.073482428	0.155021834	0.161803714
35769239	1	0.75	1	1	0.8	1
8409245	0.043478261	0.105263158	0.138297872	0.155555556	0.036697248	0.038961039
8409365	0.109730849	0.117808219	0.170483461	0.158808933	0.097701149	0.077989601
33125571	0.394957983	0.479674797	0.464480874	0.314285714	0.339483395	0.363636364
28303261	0.014521452	0.01312336	0.013991163	0.024275646	0.00665336	0.006795017
19321324	0	0.666666667	0.585365854	0.765957447	0.595744681	0.647058824
42994023	0.928571429	1	0.846153846	NaN	0.826666667	0.727272727
15824407	0.75	0.714285714	0.8	0.617647059	0.9	0.666666667
15861309	NaN	NaN	NaN	NaN	NaN	NaN
15867340	0.830083565	0.859649123	0.761904762	0.835585586	0.740331492	0.796407186
15867440	0.761742101	0.790697674	0.725394897	0.802184466	0.667844523	0.745880312
15867560	0.757296467	0.769516729	0.761399788	0.82617801	0.708299758	0.806792453
15925230	0.579545455	0.528571429	0.526315789	0.547945205	0.697749196	0.60952381
15983685	0.291079812	0.196078431	0.366863905	0.304347826	0.537234043	0.32
15987187	0.230769231	NaN	0.5	0	0.444444444	0.285714286
15988081	0.21686747	0.146341463	0.177419355	0.175	0.336956522	0.23255814
15988201	0.18616145	0.090592334	0.198614319	0.123569794	0.223300971	0.124100719
16070601	0.728643216	0.760869565	0.75	0.76969697	0.618025751	0.625
16070721	0.75596817	0.824915825	0.749450549	0.792843691	0.60727729	0.686813187
16070841	0.673938002	0.689759036	0.661870504	0.718811881	0.527872582	0.617741935
16132883	0.592307692	0.745762712	0.651162791	0.626086957	0.349206349	0.581196581
16143632	0.827848101	0.721153846	0.908424908	0.824626866	0.80349345	0.817813765

Table s2d. Dog_methylation dataset (first 40 rows).

REFERENCE

Epigenetic Pacemaker Usage, https://epigeneticpacemaker.readthedocs.io/en/latest/epm_tutorial/

Bell, Christopher G et al. “DNA methylation aging clocks: challengings and recommendations.” Genome biology vol. 20,1 249.25 Nov.2019, doi: 10.1186/s13059-019-1824-y

Frouin, A., Dandine-Roulland, C., Pierre-Jean, M., Deleuze, J.-F., Ambroise, C., & Floch, E. L. (2020). High heritability does not imply accurate prediction under the small additive effects hypothesis. *ArXiv:2007.05424 [q-Bio, Stat]*. http://arxiv.org/abs/2007.05424

Farrell C, Snir S, Pellegrini M. The Epigenetic Pacemaker: modeling epigenetic states under an evolutionary framework. Bioinformatics. 2020 Nov 1;36(17):4662-4663. doi: 10.1093/bioinformatics/btaa585. PMID: 32573701; PMCID: PMC7750963.

Horvath, S. DNA methylation age of human tissues and cell types. Genome Biol. 14, R115 (2013)

Snir, S., Wolf, Y. I., & Koonin, E. V. (2012). Universal pacemaker of genome evolution. *PLoS computational biology*, *8*(11), e1002785.

Sagi Snir, Colin Farrell & Matteo Pellegrini (2019) Human epigenetic ageing is logarithmic with time across the entire lifespan, Epigenetics, 14:9, 912-926, DOI: 10.1080/15592294.2019.1623634

Snir, Sagi. (2020). Epigenetic pacemaker: closed form algebraic solutions. BMC Genomics. 21. 10.1186/s12864-020-6606-0.

Johnson, A. A., Akman, K., Calimport, S. R. G., Wuttke, D., Stolzing, A., & de Magalhães, J. P. (2012). The Role of DNA Methylation in Aging, Rejuvenation, and Age-Related Disease. *Rejuvenation Research*, *15*(5), 483–494. https://doi.org/10.1089/rej.2012.1324

Robert G. Wallace, Laura C. Twomey, Marc-Antoine Custaud, Niall Moyna, Philip M. Cummins, Marco Mangone, Ronan P. Murphy, "Potential Diagnostic and Prognostic Biomarkers of Epigenetic Drift within the Cardiovascular Compartment", *BioMed Research International*, vol. 2016, Article ID 2465763, 10 pages, 2016. https://doi.org/10.1155/2016/2465763

Bocklandt S, Lin W, Sehl ME, Sánchez FJ, Sinsheimer JS, et al. (2011) Epigenetic Predictor of Age. PLOS ONE 6(6): e14821. https://doi.org/10.1371/journal.pone.0014821

S. L. Berger, T. Kouzarides, R. Shiekhattar, and A. Shilatifard, "An operational definition of epigenetics," *Genes and Development*, vol. 23, no. 7, pp. 781–783, 2009.

Pinho, G., Martin, J., Farrell, C., Haghani, A., Zoller, J., Zhang, J., ... & Horvath, S. (2021). Hibernation slows epigenetic aging in yellow-bellied marmots.

Di Lena, P., Sala, C., & Nardini, C. (2021). Estimage: a webserver hub for the computation of methylation age. *Nucleic Acids Research*.