Title

DNA Methylation Based Biomarkers, Imputation, and Prediction Algorithms

Author

McGreevy, Kristen Mae

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

DNA Methylation Based Biomarkers, Imputation, and Prediction Algorithms

A dissertation submitted in partial satisfaction of the requirements

for the degree Doctor of Philosophy in Biostatistics

by

Kristen McGreevy

2024

ABSTRACT OF THE DISSERTATION

DNA Methylation Based Biomarkers,

Imputation, and Prediction Algorithms

by

Kristen McGreevy

Doctor of Philosophy in Biostatistics

University of California, Los Angeles, 2024

Professor Donatello Telesca, Chair

DNA methylation (DNAm) is commonly used to develop aging biomarkers such as predictors of age, mortality risk, and blood cell counts. Challenges arise due to its high-dimensionality, variability in cytosine-phosphate-guanine (CpG) loci coverage across different arrays, and measuring the most relevant tissue DNAm. This dissertation introduces novel approaches to harness DNAm data for biomarker development, imputation accuracy enhancement, and cross-tissue prediction through three interconnected studies.

Because DNAm data are often high dimensional, they require regularized regression frameworks to construct practical prediction models. In the first arm, I developed DNAm-based biomarkers for fitness parameters, like maximum handgrip strength and VO2max, using regularized linear regression. These biomarkers demonstrate significant associations with physical activity across diverse groups, from individuals with low to intermediate activity levels to elite athletes, showcasing their potential for evaluating the epigenetic impacts of physical fitness.

DNAm data has common missingness challenges, and my second arm presents tools that utilize Copula models to enhance imputation accuracy. Unmeasured loci become problematic when DNAm biomarkers require those methylation levels for their algorithm, however, DNAm data do not commonly meet the underlying normality assumption needed for imputation tools. Therefore, we developed algorithms that can improve DNAm imputation by transforming DNAm into gaussian variables using their inherent distribution. While designed with DNAm in mind, our algorithms extend to any continuous variable needing gaussian structure, offering a versatile tool for all research projects.

The final arm explores Transfer Learning (TL) methodologies to facilitate the prediction of DNAm biomarkers across tissues, addressing the limitation of tissue accessibility in biomarker development and measurement. By enabling the use of saliva DNAm to predict blood DNAm biomarker values, this approach significantly broadens the scope of non-invasive epigenetic studies, providing researchers with robust algorithms for cross-tissue biomarker prediction and tools for development of new biomarkers. In doing so, we demonstrate how information from other tissues' DNAm can enhance biomarker prediction, and provide guidelines for researchers to implement our TL methods.

Collectively, this dissertation uncovers novel strategies for extracting valuable insights from high-dimensional DNAm data, contributing new biomarkers for physical fitness, enhancing DNAm imputation methods, and pioneering cross-tissue prediction algorithms. These offer significant advancements integrating epigenetics with the biostatistics field, facilitating a deeper understanding of DNAm and their implications for human health and aging.

The dissertation of Kristen McGreevy is approved.

Catherine Crespi-Chun

Christina Michelle Ramirez

Matteo Pellegrini

Donatello Telesca, Committee Chair

University of California, Los Angeles

2024

Contents

Vita / Biographical Sketch

Kristen McGreevy received her M.S. in Biostatistics from University of California, Los Angeles in 2020. Previously, she received a B.S in Biology and B.S.P.H in Biostatistics from University of North Carolina at Chapel Hill.

Her publications and publications under review are listed below.

- Alterations of the gut microbiome are associated with epigenetic age acceleration and physical fitness. Aging Cell, 2024.
- Associations between cardiorespiratory fitness and lifestyle-related factors with DNA methylation-based ageing clocks in older men: WASEDA'S Health Study. Aging Cell, 2024.
- Liver-derived plasminogen mediates muscle stem cell expansion during caloric restriction through the plasminogen receptor, Plg-RKT. Cell Reports, 2024.
- The Circulating Level of Klotho Is Not Dependent upon Physical Fitness and Age-Associated Methylation Increases at the Promoter Region of the Klotho Gene. Genes, 2023.
- DNA methylation clock DNAmFitAge shows regular exercise is associated with slower aging and systemic adaptation. Geroscience, 2023.
- DNAmFitAge: Biological Age Indicator Incorporating Physical Fitness. Aging Albany, 2023.
- Intravenous Administration of Umbilical Cord-Derived Mesenchymal Stem Cells (UC-MSC) for Acute Respiratory Distress Syndrome Due to COVID-19 Infection. Cureus, 2023.
- Active Learning: Subtypes, Intra-Exam Comparison, and Student Survey in an Undergraduate Biology Course. Education Sciences, 2020.
- Fluorescent Cell Staining Methods for Living Hypsibius exemplaris embryos. Cold Spring Harbor Protocols, 2018.

Under Review:

- Healthy Japanese Dietary Pattern Is Associated with Slower Biological Aging in Older Men: WASEDA'S Health Study. Submitted to Frontiers in Nutrition, Jan 2024.
- Pseudo Copula Transformations for Large Scale Missing Data Imputation of Methylation Data. Submitted to New England Journal of Statistics in Data Sciences, Oct 2023.

1 Introduction

Data sets with tens of thousands of samples and many times more variables are the new norm to statisticians in the field of epigenetics. Many classical statistical methods can be readily and robustly applied to these large data sets directly, such as epigenome-wide association studies (EWAS) for DNA methylation data. The idea is to evaluate a phenotypical trait against genotypes one locus at a time, and then summarize findings after adjusting for multiple hypothesis testing using, most commonly, false discovery rates (FDR) or Bonferroni correction [1, 2]. These methods have been effective in discovering associations between individual genotypes and phenotypes, leading to numerous discoveries and remarks [3, 4]. In epigenetics, some of the more data-driven multivariate models used in recent years include least absolute shrinkage and selection operator (LASSO) or elastic net for DNA methylation data [5, 6, 7]. This proposal will focus on DNA methylation data analyses. In later sections, I will address the statistical methods that have been applied to DNA methylation data and different phenotypical traits and outline future work that will compare and potentially improve these algorithms.

1.1 DNA methylation data

DNA methylation (DNAm) is an epigenetic modification of DNA which regulates gene expression and can be influenced by lifestyle and environmental factors. Methylation is the process of attaching a methyl group (-CH_3) to the cytosine (C) nucleotide in DNA. Scientists focus on methylation at cytosine-phosphate-guanine (CpG) sites (or loci) because these methylation patterns are retained as cells divide. The methylation process regulates gene expression without modifying the DNA sequence by preventing transcription factors from binding to the DNA [8]. For example, your skin cells and kidney cells have the same DNA, but the way they express the DNA makes the cellular type and function vastly different.

Researchers collect DNA methylation data by using bisulfite sequencing or a methylation array. The methylation arrays are cheaper and efficiently cover a fraction of total CpG sites ($\sim$ 28 million) spanning the epigenome. For example, two of the most efficient and comprehensive methylation arrays for the human genome, Illumina 450K array and EPIC v1 array, measure around 450,000 and 860,000 CpG loci, respectively. At any one CpG site on any one strand of DNA, the cytosine can be methylated or not, and arrays use 50 base pair (bp) probes

complimentary to the target loci to detect methylation at each locus. These measurements are referred to as methylation beta values and are commonly used because they can approximately be interpreted as the percent methylated at each CpG site. Specifically, the beta value is the ratio of the methylated intensity to the sum of methylated, unmethylated intensities, and a constant α as an offset for stabilizing sites where both intensities are small [9],

$$\beta_i = \frac{\max(y_{i,methy}, 0)}{\max(y_{i,unmethy}, 0) + \max(y_{i,methy}, 0) + \alpha}. \tag{1.1}$$

$y_{i,methy}$ is the methylated probe intensity and $y_{i,unmethy}$ the unmethylated probe for the ith CpG site. The constant, α, is usually set to 100 [9].

By definition, beta values are always between 0 and 1. M-values are logit transformed betas ($\log_2(\beta/(1-\beta))$), which take values from negative to positive infinity and are more homoscedastic. However, this transformation to M-values does not make the distribution of methylation values more gaussian. Instead, DNAm tend to be concentrated near 0 or 1, skewed, and/or multimodal. The Beta distribution is commonly used to simulate methylation data because it has support on (0, 1) and has a lot of flexibility. Software that simulates methylation data commonly use this distribution and allow for bimodality [10]. Recently, the beta-binomial distribution has also been proposed for use with DNAm data [11]. Therefore, it is commonly accepted that methylation is not gaussian in nature and requires flexible frameworks for modeling. Additional context around DNAm data distributions will be discussed in later sections.

1.2 DNAm Biomarkers

Hundreds of thousands of CpG sites across the genome change methylation states as organisms grow older. This change means aging is reflected in DNAm, which has enabled the construction of high-precision algorithms that predict age. These are collectively known as epigenetic clocks, and a large body of literature demonstrates these clocks are associated with human mortality risk [7, 12] and various age-related conditions [7, 13, 14]. The first DNAm-based age predictor was built in 2011 using LASSO (least absolute shrinkage and selection operator) regression from saliva [15]. It was later demonstrated that aging clocks could be built in almost all human tissues. Horvath developed a multi-tissue human DNA methylation epigenetic clock (DNAmAge) in 2012 which is a linear combination of 353 CpG sites and later become one of the most widely recognized human epigenetic aging clocks [6]. Since then, epigenetic clocks have

expanded into broader DNAm-based biomarkers which can estimate mortality risk [7], smoking [7], and blood cell count [16], which also appear to reflect one's biological age.

1.3 Biological Age

Aging is often manifested through the gradual and progressive deterioration across a variety of organ systems- like decline in cardiovascular, metabolic, mental, immune, and pulmonary systems [17]. Chronological age (CA) is a strong predictor of deterioration across systems, however, it does not explain why two people of the same age and sex can have such different aging phenotypes. Some people appear to age faster than others; they have earlier onset of diseases, frailty, or die at a younger chronological age. Conversely, lifestyle choices, such as a healthy diet and exercise are known to increase healthspan and slow the decline in some organ systems [18, 19]. Chronological age fails as an aging biomarker because it merely reflects the passage of time. Biological age (BA), however, reflects the underlying biological processes of aging captured through molecular and cellular changes. Therefore, measures of BA should be stronger predictors of age-related phenotypes than CA and distinguish people of the same chronological age but with different aging phenotypes.

Biological age does not have a gold standard metric for measuring it. Instead, BA is a *latent variable*, and we derive estimates of BA using observable variables that (we believe) relate to the biological process of aging. Some biological age indicators use DNAm, some use physiological tests, and others use plasma proteins to provide insight to the biological aging process.

2 DNAm Fitness Biomarkers and DNAmFitAge, a Biological Age Indicator Incorporating Physical Fitness

In this section I describe the first arm of my research, which entails developing DNAm biomarkers of fitness parameters and incorporating them into an estimate of biological age. This research corresponds to a manuscript published in Aging Albany titled "DNAmFitAge: Biological Age Indicator Incorporating Physical Fitness" [20].

2.1 Motivation

Physical fitness declines with aging and is well known to correlate to health [21]. This decline is evident in reduced function in specific organs, like lungs [22], and in performance tests of strength [23] or aerobic capacity [24]. The rate of this decline varies between individuals [18, 25], and those who preserve physical fitness as they age are at lower risk for a range of diseases and tend to live longer lives [18, 19, 26]. At the molecular level, changes in fitness and related indices of functional capacity correlate with changes in molecular signs of decline thought to reflect underlying biological processes of aging [27]. Measures of fitness may therefore provide a new window into biological aging [28]. However, fitness measurements are not possible for studies with remote data collection or those conducted with stored biospecimens. To enable such studies to quantify fitness, I developed blood based DNAm biomarkers of fitness parameters spanning mobility, strength, lung function, and cardiovascular fitness and use these to construct a novel indicator of fitness-based biological age, DNAmFitAge.

Multiple lines of evidence support a focus on DNAm to develop biomarkers of fitness. First, prediction of aging-related morbidity, disability, and mortality by DNAm biomarkers is enhanced by the incorporation of physiological data into prediction algorithms [7, 12, 17]. This suggests utility in including physical fitness in DNAm biomarkers, however, current DNAm biomarkers do not use fitness parameters in their construction. Second, there is emerging evidence that epigenetic clocks are sensitive to lifestyle factors [29], individual differences in fitness parameters are reflected in DNAm data [30, 31], and blood DNAm differs between athletes and controls [32]. Therefore, a growing body of evidence suggests blood DNAm carries information related to physical fitness, but it was unknown if fitness parameters could be estimated using

blood DNAm levels.

I develop blood DNAm biomarkers of four fitness parameters: gait speed (walking speed), maximum handgrip strength, forced expiratory volume in 1 second (FEV1; an index of lung function), and maximal oxygen uptake (VO2max; a measure of cardiorespiratory fitness). These parameters were chosen because handgrip strength and VO2max provide insight into the two main categories of fitness: strength and endurance [33], and gait speed and FEV1 provide insight into fitness-related organ function: mobility and lung function [26, 34]. Furthermore, each parameter is known to be associated with aging, mortality, and disease [26, 34]. The newly constructed DNAm biomarkers provide researchers a new method to incorporate physical fitness into epigenetic clocks and emphasizes the effect lifestyle has on the aging methylome.

2.2 Methods

2.2.1 LASSO Penalized Regression for DNAm Fitness Biomarkers

I used blood DNAm data from three datasets, Framingham Heart Study Offspring cohort (FHS, n=1830), Baltimore Longitudinal Study on Aging (BLSA, n=820), and novel data (Budapest, n=307) to develop the DNAm biomarkers of fitness parameters. In short, the FHS cohort is a cardiovascular study which followed adults from Massachusetts starting in 1948 [35]. The BLSA cohort began in 1958 studying healthy adults and the aging process [36]. Finally, Budapest is a smaller study (n=307) measuring physical fitness and DNA methylation in middle to older aged adults, some of whom are current or former rowing athletes. Dataset harmonization was performed to join multiple datasets when variables were on different scales following previously developed methods [37]. In brief, datasets were rescaled to have the same mean and standard deviation for each fitness parameter by recentering and multiplying by the ratio of standard deviations.

I developed DNAm biomarkers for four fitness parameters: gait speed, maximum handgrip strength (Gripmax), forced expiratory volume in 1 second (FEV1), and maximal oxygen uptake (VO2max). Gait speed, also known as walking speed, is measured in meters per second. Maximum hand grip strength is a measurement of force taken in kg. FEV1 measures lung function; it is the amount of air forced from the lungs in one second, measured in liters. VO2max is a measure of cardiovascular health and aerobic endurance [24]. It measures the volume of oxygen the body processes during incremental exercise in milliliters used in one minute of exercise per

kilogram of body weight (mL/kg/min). VO2max has been regarded as the best indicator of an athlete's physical capacity and is the international standard of physical capacity [38].

Each fitness DNAm biomarker was developed using least absolute shrinkage and selection operator (LASSO) penalized regression with 10-fold cross validation in which the fitness parameters were dependent variables and independent variables were either (1) DNAm levels at cytosine-phosphate-guanines (CpG) sites and chronological age or (2) DNAm levels at CpG sites only. Removing age as a potential variable for selection in LASSO was performed to remedy high collinearity discovered among these DNAm biomarkers when constructing DNAmFitAge. The LASSO-regression method uses an l_1 penalty on a standard multiple linear regression that shrinks each coefficient towards zero [39]. Specifically, LASSO minimizes with respect to β,

$$\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda\|\beta\|_1 \tag{2.1}$$

where $\mathbf{y}$ is the $n \times 1$ vector of a fitness parameter, $\mathbf{X}$ is the $n \times p$ matrix of methylation values at p CpG loci, β is the coefficient vector of all p features, and λ is a tuning parameter for the shrinkage. An optimal λ was chosen using the 10-fold cross validation process.

LASSO is more effective than Ridge (l_2 penalty) [40] and elastic net (mixture of l_1 and l_2 penalty) [41] when handling many irrelevant predictors and yields a smaller number of predictors in the final model because coefficients are shrunk to 0. Models were fit separately for men and women in the case of gait speed, gripmax, and FEV1 to select for sex specific CpG loci that reflect gender variation in fitness. When it came to building the biomarker for VO2max, stratifying by sex was not feasible due to the smaller sample size. This forced us to choose between using sex as a covariate or omitting sex and trusting LASSO to select X chromosome markers that best signify differences between males and females. We chose the latter, and it did. The selected covariates and estimated coefficients were then used as a prediction algorithm for each fitness parameter. I refer to the predicted fitness parameters generated by these algorithms as DNAmGaitspeed, DNAmGripmax, DNAmFEV1, and DNAmVO2max.

2.2.2 DNAmFitAge Construction via Klemera Doubal Method

DNAmFitAge is an indicator of biological age which is constructed separately for males and females using four DNAm variables: three of the DNAm fitness biomarkers: DNAmGripmax, DNAmGaitSpeed, and DNAmVO2max, and DNAmGrimAge, a biomarker of mortality risk [7].

DNAmGrimAge is formed from multiple DNAm biomarkers, including seven plasma proteins and smoking pack years. Including DNAmGrimAge allows DNAmFitAge to be a well-rounded biological age indicator that captures multiple facets of aging, not just facets related to physical fitness. Models including DNAmFEV1 as a fifth variable were explored, however no improvement in association to physical activity or age-related outcomes were observed; the parsimonious DNAmFitAge model using a subset of the DNAm fitness biomarkers was therefore chosen.

I constructed DNAmFitAge following the methods proposed by Klemera and Doubal [42] for constructing estimates of biological age. The Klemera-Doubal model framework stipulates there exists an underlying trait which is unobserved (biological age) which relates to an observable trait (chronological age) and a set of additional variables (DNAm biomarkers). This framework posits biological age is centered on chronological age with additional noise. The linear relationship between chronological age and each DNAm biomarker is used to estimate the relationship between biological age and the DNAm biomarker. Weighted least squares combines the standardized variables into a biological age estimate where the weights are formed from correlations of each variable with chronological age.

Let b_j be jth biomarker of interest from $j = 1, \ldots, 4$ (DNAmGaitSpeed, DNAmGripmax, DNAmVO2max, DNAmGrimAge). The correlation between b_j and chronological age is r_j and the linear relationship b_j has with chronological age is

$$b_j = \alpha_{0j} + \alpha_{1j} \times \text{Age} \tag{2.2}$$

where α_{0j} is the intercept and α_{1j} is the slope. For the ith observation, DNAmFitAge is calculated as a weighted sum:

$$\text{DNAmFitAge}_i = \sum_{j=1}^{4} w_j \frac{b_{ij} - \alpha_{0j}}{\alpha_{1j}} \tag{2.3}$$

where w_j is the weight of each DNAm biomarker based on its strength of relationship with chronological age. Specifically,

$$w_j = \frac{\frac{r_j^2}{1-r_j^2}}{\sum_{k=1}^{4} \frac{r_k^2}{1-r_k^2}}. \tag{2.4}$$

I estimate all parameters (α_0, α_1, r) using the harmonized FHS and BLSA training dataset.

Therefore, the only component that is different for each person's estimated DNAmFitAge is the DNAm biomarker values.

Finally, I created FitAgeAcceleration, the age-adjusted estimate of DNAmFitAge formed from taking the residuals after regressing DNAmFitAge onto chronological age. As such, FitAgeAcceleration is uncorrelated with chronological age. FitAgeAcceleration provides an estimate of epigenetic age acceleration, ie how much older or younger a person's estimated biological age is from expected chronological age. A positive FitAgeAcceleration means biological age is estimated to be older than chronological age. A negative FitAgeAcceleration means biological age is estimated to be younger than chronological age, which is the preferred outcome for a person.

DNAmFitAge and FitAgeAcceleration are novel DNAm biomarkers incorporating mortality risk with strength, mobility, and cardiovascular fitness.

2.3 Validation

Validation of the DNAm fitness biomarkers and DNAmFitAge consisted of multiple steps across five independent datasets. First, I correlated DNAm biomarker values with direct measurements of the fitness parameters and correlated DNAmFitAge with chronological age. In cases where direct measurement of a fitness parameter was not included in a validation dataset, substitutions were selected. Strong correlation between DNAm fitness biomarkers and true fitness parameters indicate the strength of a surrogate DNAm marker in estimating the fitness value. The Klemera Doubal modeling framework posits biological age is centered on chronological age, therefore validation datasets should demonstrate good correlation and general centeredness between DNAmFitAge and chronological age. Second, I test the association of DNAm biomarkers to aging-related variables. The DNAm biomarkers provide insight to the aging process through a fitness paradigm, therefore they should relate to aging-related phenotypes. Third, I test the relationship of DNAm biomarkers to physical activity and athlete status to demonstrate these biomarkers do capture fitness.

I combine results across validation studies using fixed effect models or Stouffer's meta analysis method. Fixed effect models use the inverse variance to weight estimates, and Stouffer's method uses the square root of the sample size to weight estimates. The latter is used when harmonization across cohorts was challenging; such as with physical activity variables, the number of age-related conditions, disease free status, and age at menopause. Forest plots evaluating

FitAgeAcceleration hazard ratios or coefficients in models adjusted for age and sex are displayed in Figure 2.3 and Supplemental Table B.4. We perform a test of heterogeneity for coefficients across datasets using Cochran Q test for fixed effect models; p-values are displayed as Het. P. Validation relationships evaluating error are presented in Supplemental Table B.1.

Detailed descriptions of each validation dataset are provided in the appendix A.2.

2.3.1 Correlations

I calculate Pearson's correlation between the DNAm fitness biomarkers and fitness parameters in validation datasets to understand how well the DNAm fitness surrogates estimate the true fitness parameters. The DNAm fitness biomarkers had modest correlation with direct fitness parameters. Average correlations across validation datasets ranged from 0.16-0.48 (Figure 2.1, Table 2.1). Correlation of DNAmVO2max to VO2max in CALERIE, the one validation dataset with the same direct fitness parameter, has good correlation overall and within sex (overall R=0.55, female R=0.19, male R=0.47).

DNAmFitAge had strong correlation to chronological age in validation datasets. The average Pearson r between DNAmFitAge and chronological age across validation datasets was 0.77 (Figure 2.2), and the lower correlation in LBC1921 (r=0.38) and LBC1936 (r=0.68) can be attributed to the small age range they cover. LBC1921 ages ranged from 77 to 90 and LBC1936 ages ranged from 67 to 80. The average r excluding LBC cohorts was 0.92. In addition, each validation dataset had low median absolute deviation (median of the absolute difference from chronological age to biological age) ranging from 2.3 to 4.9 years (Supplemental Table 3). Reproducibility across a wide span of ages (21 in CALERIE to 100 in InChianti) demonstrate DNAmFitAge's calibration across a wide adult age range.

2.3.2 Age-related Phenotypes

I tested DNAm fitness biomarker and DNAmFitAge associations to multiple aging-related variables in validation datasets. Specifically, I conducted regression analysis for time-to-death, time-to-coronary-heart-disease (CHD), the count of age-related conditions (arthritis, cataract, cancer, CHD, CHF, emphysema, glaucoma, lipid condition, osteoporosis, and type 2 diabetes), age at menopause, cancer, hypertension, type-2 diabetes, and disease-free status. Time-to-event outcomes were analyzed using Cox regression to estimate hazard ratios (HR); continuous outcomes were analyzed using linear regression to estimate slopes; dichotomous outcomes were

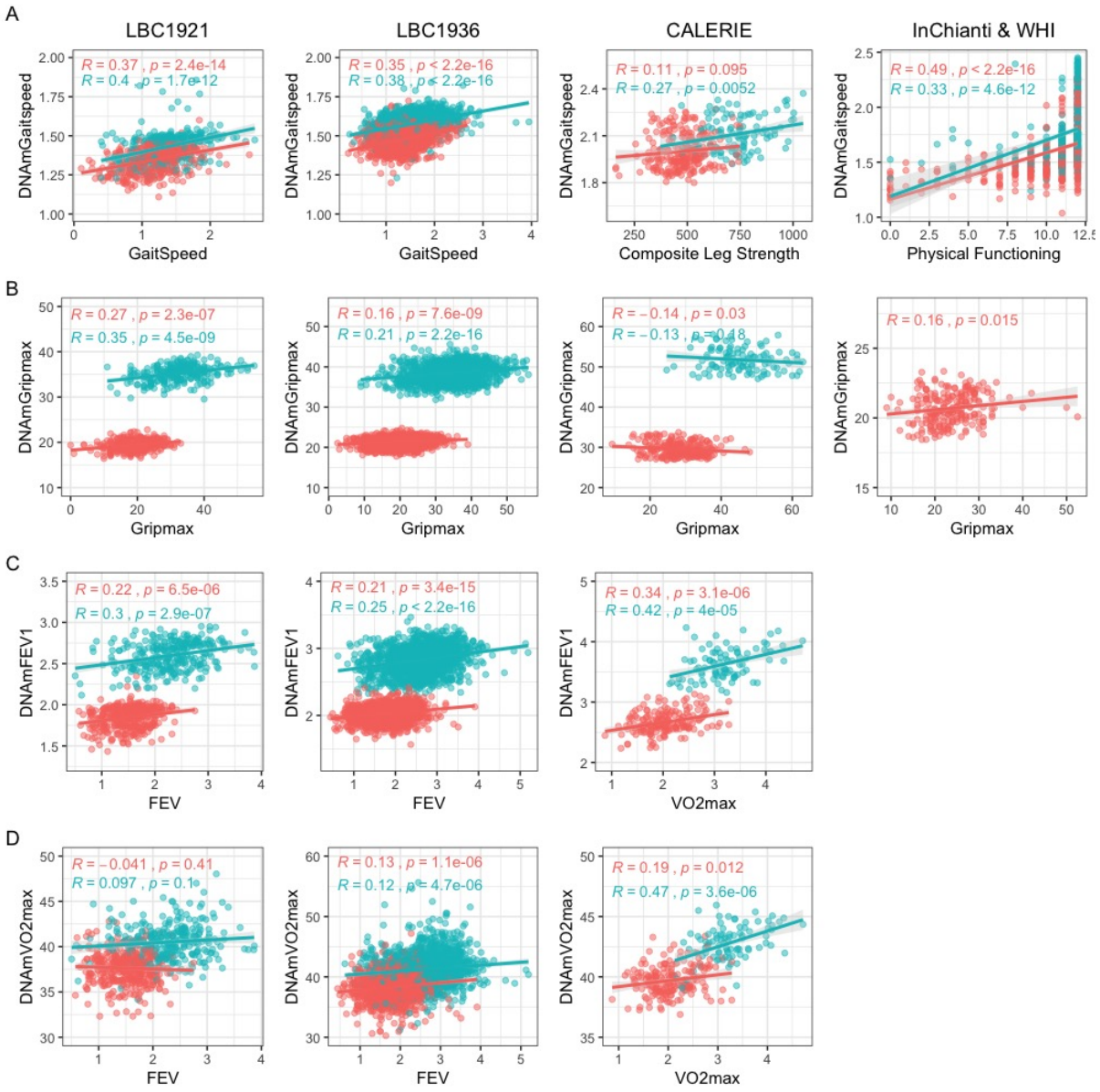


Figure 2.1: Scatterplots of DNAm fitness biomarker models versus true values in test datasets. Pink indicates females, and blue indicates males. When original variables were unavailable, best alternative variables are plotted against the DNAm fitness estimates. Each panel corresponds to the performance of one DNAm-based model built with chronological age across test datasets displayed with Pearson correlation and p-values. (A) DNAmGaitspeed with performance in InChianti dataset displayed, (B) DNAmGripmax with performance in WHI dataset, (C) DNAmFEV1, (D) DNAmVO2max. (A-C) (DNAmGaitspeed, DNAmGripmax, and DNAmFEV1) were built in each sex separately while (D) (DNAmVO2max) was built in both sexes jointly.

Table 2.1: DNAm Fitness Parameter Biomarker Pearson Correlation

DNAm Biomarker	CpGs	Age in Model	Sex	FHS + BLSA	Budapest	LBC1921	LBC1936	CALERIE	InChianti	WHI	Average Test R
Gaitspeed	42	Y	Females	0.61	0.61	0.37	0.34	0.11*	0.49^{+}	0.15^{+}	0.34
	26	Y	Males	0.43	0.59	0.40	0.38	0.27*	0.33^{+}		0.39
	53	N	Females	0.56	0.56'	0.17	0.17	0.095*	0.43^{+}	0.12^{+}	0.26
	59	N	Males	0.60	0.53'	0.23	0.21	0.26*	0.34^{+}		0.31
Gripmax	52	Y	Females	0.66	0.54	0.27	0.16	-0.14		0.16	0.20
	52	Y	Males	0.68	0.50	0.35	0.19	-0.089			0.24
	91	N	Females	0.66	0.52	0.22	0.10	-0.16		0.12	0.16
	93	N	Males	0.66	0.43	0.21	0.14	-0.078			0.18
FEV1	77	Y	Females	0.59	0.50^{v}	0.21ˆ	0.20ˆ	0.34			0.31
	73	Y	Males	0.63	0.30^{v}	0.30ˆ	0.25ˆ	0.42			0.32
VO2max	40	Y	Both	$0.52^{\$}$	0.70	0.43ˆ	0.40ˆ	0.55			0.48

Note: superscripts indicate correlation is with the closest fitness parameter available: ' Jumpmax, * Composite Leg Strength, $^{+}$ Physical Functioning, ˆFEV, $^{\$}$ FEV1, ^{v} VO2max

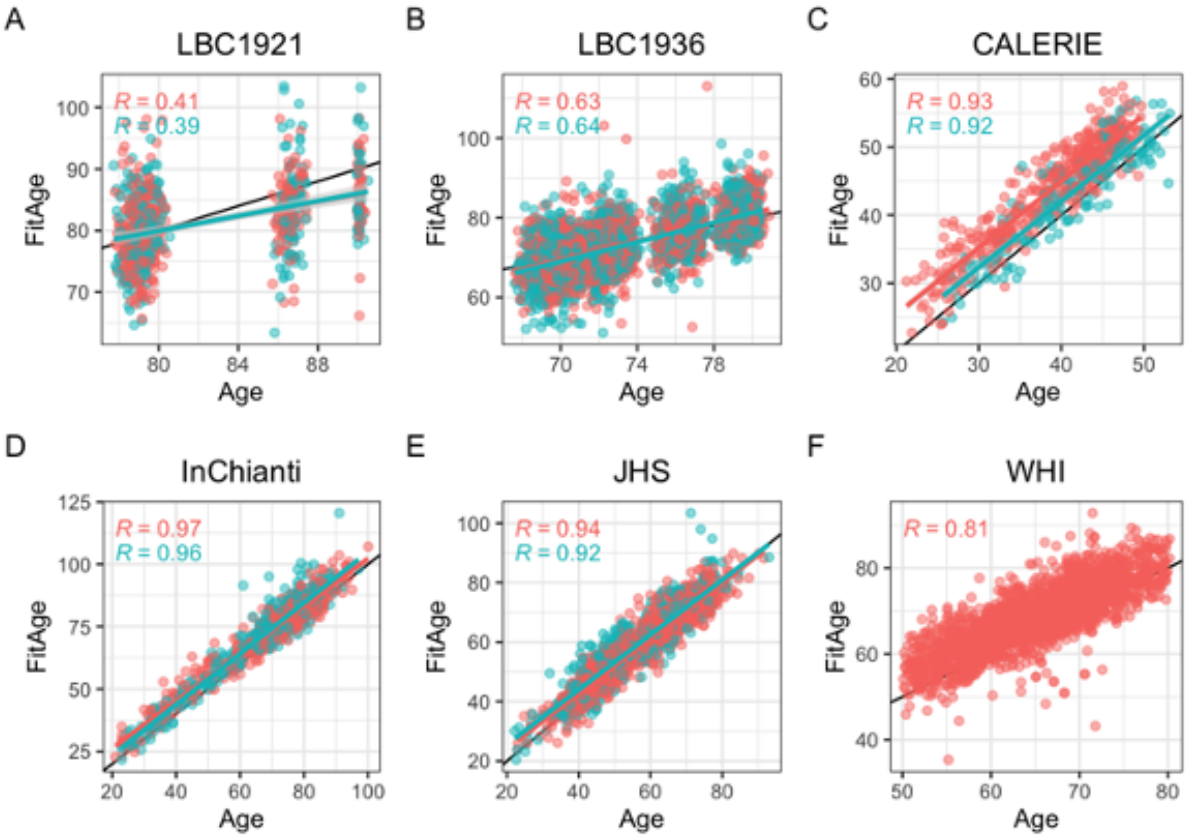


Figure 2.2: Scatterplots of DNAmFitAge versus age separated by sex. Pink indicates females, and blue indicates males. (A-F) Each panel corresponds to the performance of DNAmFitAge in one validation dataset displayed with Pearson correlation to chronological age and corresponding p-values. DNAmFitAge models applied to the same sex it was built in (ie DNAmFitAge built for females tested in females and DNAmFitAge built for males tested in males). DNAmFitAge is centered on chronological age with high correlation across all test sets.

analyzed using logistic regression to estimate odds ratios (OR); and ordinal outcomes were analyzed using multinomial regression to estimate OR. Some of our cohorts (InChianti, LBC1921, and LBC1936) involved longitudinal measures. In these cases, linear regression models with person-level random intercepts were implemented in R using the lmer function to adjust for correlation within the same individual.

All DNAm fitness biomarkers are individually predictive of mortality and disease-free status, and some are predictive of type 2 diabetes status and number of comorbidities in the validation datasets. After controlling for age and sex, higher (or more fit) values of DNAmGaitspeed (p = 1.1E-10), DNAmGripmax (p = 2.6E-9), DNAmFEV1 (p = 2.2E-20), and DNAmVO2max (p = 0.003) are associated with decreasing mortality risk (Supplementary Figure B.1). For example, on average, every 1 kg stronger DNAmGripmax is has an associated 5% decrease in mortality risk compared to a person of the same age and sex (hazard ratio = 0.95, confidence interval = [0.93, 0.96]). DNAmGaitspeed and DNAmFEV1 are both predictive of type 2 diabetes status (p=0.0013, p=0.0032) and number of comorbidities (p=0.0004, p=4E-12). Stronger values of any DNAm fitness biomarkers are associated with disease-free status.

I find that the age-adjusted version of FitAge, FitAgeAcceleration, is a significant predictor of mortality risk (all cause mortality), coronary heart disease, and other age-related conditions. Cox Proportional Hazard models demonstrated FitAgeAcceleration is a strong predictor for time-to-death (p=7.2E-51) and time-to-coronary heart disease (p=2.6E-8). FitAgeAcceleration had an overall hazard ratio of 1.07 (1.06, 1.08) (Figure 2.3). Thus, a FitAgeAcceleration value of 10 years was associated with almost doubling the mortality risk compared to the average person of the same age and sex ($1.07^10 = 1.97$ risk). Similarly, an increase in FitAgeAcceleration corresponds to more comorbidities (p=9.0E-9), hypertension (p=8.7E-5), and earlier age at menopause (p=6.6E-9) (Figure 2.3, Supplemental Table 4). A lower FitAgeAcceleration was associated with disease free status (p= 1.1E-7) and lower cholesterol (p=0.0005) (Supplemental Table 4). FitAgeAcceleration is additionally informative for mortality risk beyond the information captured with AgeAccelGrim (Age adjusted DNAmGrimAge) in JHS females and in InChianti males and females when comparing LRT p-values (Supplemental Table B.5). Our results indicate FitAgeAcceleration is informative for mortality risk and may act as a supplement (not replacement) to AgeAccelGrim.

Each of these associations were in the expected direction, as someone who had a low FitAgeAcceleration had a biological age estimate that was younger than their chronological

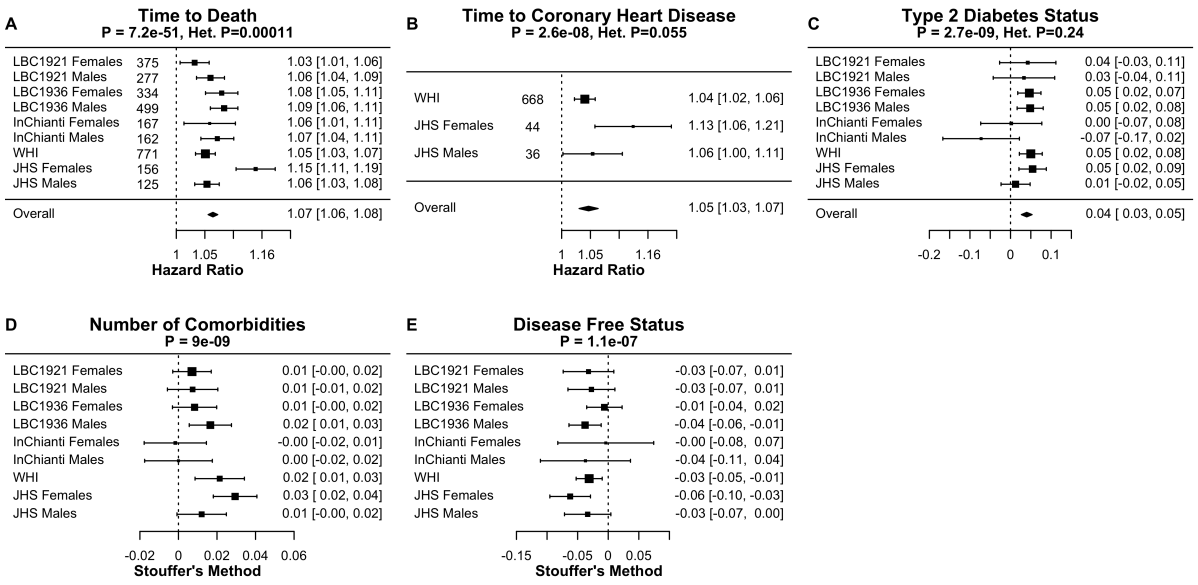


Figure 2.3: Meta-analysis forest plots for FitAgeAcceleration to age-related conditions adjusted for age and sex. Each panel reports a meta analysis forest plot for combining hazard ratios or regression coefficients across dataset cohorts. (A) Time-to-death with number of events, (B) time-to-coronary heart disease with number of events, (C) type 2 diabetes, (D) comorbidity count, and (E) disease free status. Meta-analysis p-values are displayed in the header of each panel, and test of heterogeneity Cochran Q test p-value (Het. P) are displayed for fixed effect models. Fixed effects models were used for (A-C) and Stouffer's method was used for (D, E).

age. Hence, people whose DNAm predicted them to be more 'physically fit' than their chronological age would suggest had better age-related outcomes. These relationships demonstrate epigenetic age acceleration can be well explained through DNAm fitness parameter biomarkers, and that FitAgeAcceleration provides a practical tool for relating fitness to the aging process.

2.3.3 Physical Activity and Athlete Status

The DNAm fitness biomarkers and DNAmFitAge incorporate physical activity into epigenetic clocks, therefore, they should relate to physical activity and athlete status. I test for associations across a range of fitness levels including people with low to intermediate physical activity, long-term rowing athletes (current and former), and body builders.

Low to Intermediate Physical Activity

I used linear regression to test for associations between physical activity or physical functioning in low to intermediate physically fit individuals with DNAm fitness parameter biomarkers and FitAgeAcceleration in the validation datasets. I restricted our analysis to people of low to intermediate fitness to determine if FitAgeAcceleration is sensitive to small improvements

Table 2.2: Reference DNAm Fitness Parameter Values for Fit (FitAge Acceleration ≤ -5 yrs) and Unfit (FitAge Acceleration $\geq +5$ yrs) Individuals

Females								
	DNAmGaitspeed		DNAmGripmax		DNAmVO2max		DNAmGrimAge	
Age	Fit	Unfit	Fit	Unfit	Fit	Unfit	Fit	Unfit
<40	2.1	2.0	34.6	30.5	42.8	40.1	37.1	40.9
40-59	1.9	1.7	31.3	26.9	39.2	37.9	49.2	60.5
60-79	1.7	1.5	28.8	22.4	37.6	36.1	63.2	72.4
80+	1.6	1.3	23.9	19.1	37.0	35.4	74.7	81.8
Males								
	DNAmGaitspeed		DNAmGripmax		DNAmVO2max		DNAmGrimAge	
Age	Fit	Unfit	Fit	Unfit	Fit	Unfit	Fit	Unfit
<40	2.1	1.8	49.3	43.8	45.1	44.9	34.8	52.9
40-59	1.9	1.7	46.6	42.5	43.9	42.3	47.5	60.1
60-79	1.7	1.5	41.3	36.8	43.1	39.5	68.0	77.9
80+	1.6	1.3	39.3	32.0	41.3	37.7	78.0	86.7

in fitness. In addition, this separation captures low to average physically active individuals in each dataset. LBC1921, LBC1936, and JHS measure physical activity, and WHI and InChianti measure physical functioning. Higher values of any variable indicate more activity or better physical functioning.

FitAgeAcceleration, DNAmGaitspeed, DNAmGripmax, and DNAmFEV1 have associations in the expected direction with physical activity in low to intermediate physically active individuals. Coefficients indicate the effect on physical activity for a one unit increase in each DNAm fitness biomarker after adjusting for chronological age within each sex (Table 2.3, Figure 2.4). The relationship to DNAmFitAge is as expected; someone with a higher FitAgeAcceleration has an estimated biological age that is older than expected, which corresponds to lower physical activity or physical functioning (Table 2.2). Similarly, men and women with a faster DNAmGaitspeed, stronger DNAmGripmax, and larger DNAmFEV1 are more physically active when holding age constant. In conclusion, men and women who were more active showed correspondingly 'fitter' values of FitAgeAcceleration and the DNAm fitness biomarkers.

Rowing Athletes

I evaluated whether DNAm fitness biomarkers and DNAmFitAge were able to distinguish athletes from controls across two independent studies using Kruskal Wallis tests. The Budapest study is a small, novel study (n=307) measuring physical fitness and DNA methylation in middle to older aged adults, some of whom are current or former athletes. The athletes (n=83 females, n=110 males) previously participated in the World Rowing Masters Regatta in Velence,

Table 2.3: Association of DNAm Biomarkers to Physical Activity and Physical Functioning in People with Low to Intermediate Activity Levels

Outcome		Females					Males				Meta Analysis p-value
		LBC 1921	LBC 1936	InChianti	JHS	WHI	LBC 1921	LBC 1936	InChianti	JHS	
DNAmFitAge	coefficient	-0.024	-0.031	-0.095	-0.033	-0.237	0.008	-0.024	-0.041	-0.040	6.37E-13
	p-value	2.3E-04	3.7E-06	0.042	0.046	0.014	0.199	2.0E-05	0.272	0.044	
DNAmGaitSpeed w/ Age	coefficient	-0.51	2.82	8.76	3.07	26.67	-3.31	1.99	4.97	1.78	1.82E-03
	p-value	0.567	0.002	0.084	0.165	0.025	2.4E-04	0.022	0.429	0.672	
DNAmGaitSpeed w/o Age	coefficient	0.87	0.90	1.32	0.40	8.77	-0.10	0.99	2.36	1.49	1.60E-06
	p-value	0.001	0.004	0.536	0.627	0.099	0.725	0.001	0.255	0.293	
DNAmGripmax w/ Age	coefficient	0.10	-0.06	0.14	-0.03	0.24	0.00	0.03	0.17	0.02	0.029
	p-value	0.036	0.201	0.635	0.821	0.035	0.943	0.256	0.291	0.801	
DNAmGripmax w/o Age	coefficient	0.076	0.035	0.10	0.002	0.30	-0.02	0.02	0.06	0.02	1.85E-04
	p-value	4.3E-06	0.043	0.379	0.953	0.181	0.125	0.058	0.364	0.617	
DNAmFEV1	coefficient	1.07	0.60	0.33	0.99	4.98	-0.17	0.37	0.37	-0.58	0.0062
	p-value	0.026	0.197	0.898	0.114	0.005	0.585	0.173	0.791	0.337	
DNAmVO2max	coefficient	0.06	0.03	0.26	-0.05	-0.47	-0.06	0.02	0.05	-0.03	0.113
	p-value	0.003	0.090	0.019	0.423	0.215	0.002	0.281	0.667	0.654	
DNAmGrimAge	coefficient	-0.01	-0.03	-0.08	-0.05	-0.30	-0.01	-0.03	-0.01	-0.05	1.25E-12
	p-value	0.524	5.7E-06	0.157	0.002	0.027	0.390	2.3E-05	0.794	0.007	
DNAmPhenoAge	coefficient	0.00	-0.01	-0.12	-0.02	-0.07	7.1E-05	-0.01	-0.01	-0.01	1.26E-06
	p-value	0.568	0.012	4.2E-04	0.063	0.354	0.989	0.004	0.764	0.425	
DNAmPAI1	coefficient	-3.4E-05	-4.4E-05	-7.5E-05	-1.1E-04	-3.4E-04	-1.8E-06	-2.8E-05	7.5E-05	-6.2E-05	6.36E-10
	p-value	0.021	0.002	0.358	2.5E-08	0.076	0.908	0.032	0.382	0.009	
DNAmGDF15	coefficient	-8.5E-05	-1.3E-03	-2.9E-03	-8.5E-04	-1.0E-02	-1.2E-03	-5.9E-04	-3.5E-03	-6.7E-04	6.16E-08
	p-value	0.802	0.0005	0.082	0.147	0.047	5.4E-04	0.054	0.117	0.373	

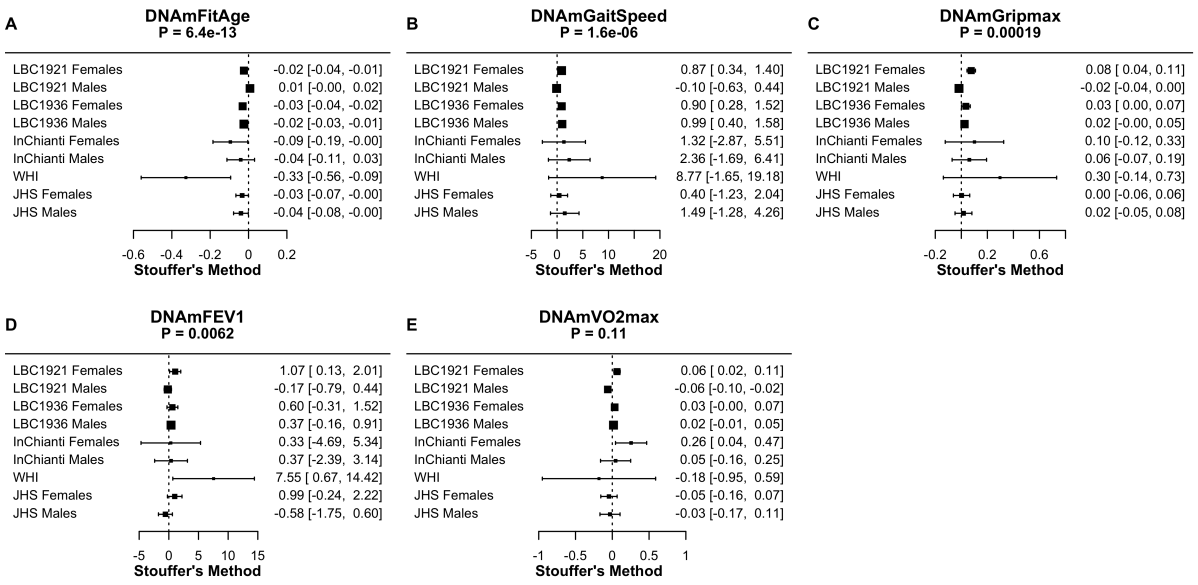


Figure 2.4: Meta-analysis forest plots for DNAmFitAge and DNAm fitness parameters relationship to physical activity or physical functioning in people with low to intermediate physical activity. Each panel reports the Stouffer's meta-analysis p-value for combining coefficients across dataset cohorts after adjusting for chronological age. (A) DNAmFitAge, (B) DNAmGaitspeed, (C) DNAmGripmax, (D) DNAmFEV1, and (E) DNAmVO2max. DNAmFitAge, DNAmGaitSpeed, DNAmGripmax, and DNAmFEV1 are predictive of physical activity in low to intermediate physically active individuals.

Hungary. We use the age-adjusted DNAm variables (such as FitAgeAcceleration) to remove the age effect seen between groups. The Polish Study is a small, novel study (n=416) measuring blood DNA methylation and lifestyle behaviors in Polish body builders and similar aged healthy controls ranging from 17 to 56 years of age (Kruskal wallis p-value ¿ 0.05). There were a total of 66 male body builders and 30 female body builders. Because of the small sample size in females, we restricted the analysis to males only, which decreases the sample size to 215 individuals total, 149 male controls and 66 male body builders. Both groups reported the number of years they regularly trained, the average number of intensity trainings they participated in per week, and 88 total participants reported supplements or drugs they are taking. Kruskal Wallis tests are reported for physical fitness parameters to provide a reference for the DNAm fitness biomarkers.

All the DNAm fitness biomarkers can distinguish trained from untrained females in the Budapest study but only DNAmVO2max distinguishes fitness grouping in males. We hypothesize the DNAm fitness biomarkers could not detect differences in the male fitness groups because the true physical fitness parameters are not very distinct between the male groups (relative gripmax p=0.252 and jumpmax p=0.015). FitAgeAcceleration distinguishes male and female athletes from controls in the Budapest study better than the other epigenetic clocks. FitAge

Acceleration is 2.7 years younger on average in trained females compared to untrained females (p=1.7E-7), and FitAgeAcceleration is 2.3 years younger on average in trained males compared to untrained males (p=0.0002). AgeAccelGrim and AgeAccelPheno estimate younger values in the trained groups of males and females, but only GrimAge is significant in females (p = 0.019). Furthermore, the differences observed between female fitness groups with GrimAge and PhenoAge Acceleration have smaller magnitudes than FitAgeAcceleration. Therefore, trained females and trained males are estimated to be 2.7 and 2.3 years biologically younger on average than their untrained counterparts, suggesting regular physical exercise is protective to biological age in males and females.

Male Body Builders

In the Polish study, male body builders are estimated to be biologically younger and more physically fit compared to male controls of the same age. On average, DNAmFitAge is 2.74 years younger in male body builders compared to controls (p=0.041), and DNAmVO2max is 0.4 mL/kg/sec better in male body builders (p=0.023) (Table 2.4). FitAge Acceleration (p=0.080), DNAmGaitspeed (p=0.055), and DNAmGripmax (p=0.075) are suggestive of having improvement in male body builders, however they were not significant at the 0.05 level. Boxplots displaying the spread of DNAmFitAge, DNAmVO2max, FitAge Acceleration, and DNAmGaitspeed between body builders and controls are presented in Figure 2.5. Male body builders have 5.4 more years of regular training (p=2.6E-6) and 1.1 more training sessions per week (p=9.4E-7) compared to male controls on average, and the DNAmFitAge and DNAmVO2max results correspond to male body builders being estimated as more physically fit, as expected. Our promising results in male body builders show a physically fit lifestyle corresponds to biological aging benefits that can be captured with our new DNAm fitness biomarkers and DNAmFitAge.

I evaluate whether dietary supplement use can explain the improvement in DNAmFitAge or DNAmVO2max in male body builders in the appendix A.3, accompanying Supplemental Table B.2.

2.4 Discussion

DNAm biomarkers have been constructed for blood cell count [16], age [6, 43], smoking [7], and more, however, there were not yet DNAm biomarkers for fitness parameters. Our work introduces new DNAm biomarkers for the fitness parameters of maximum handgrip strength, gait speed, FEV1, and VO2max. These DNAm biomarkers represent new tools for researchers

Table 2.4: Comparison between Male Controls and Body Builders in Polish Study

	Mean Control (n=149)	Mean Body Builder (n=66)	Control - Body Builder	Kruskal Wallis p-value
Intensity trainings per week	3.0	4.1	-1.1	9.43E-07
Years regular training	6.6	12.0	-5.4	2.61E-06
DNAmFitAge	41.1	38.4	2.74	0.041
DNAmVO2max	44.0	44.4	-0.40	0.023
FitAgeAcceleration	0.15	-0.56	0.72	0.080
DNAmGaitspeed w/o Age	1.99	2.02	-0.03	0.055
DNAmGripmax w/o Age	46.5	47.2	-0.69	0.075
DNAmFEV1	3.82	3.87	-0.05	0.199
DNAmGrimAge	44.1	41.8	2.24	0.063
DNAmPhenoAge	26.7	24.7	2.01	0.181
DNAmPAI1	19033	18238	795	0.009
DNAmGDF15	701.8	680.4	21.4	0.447

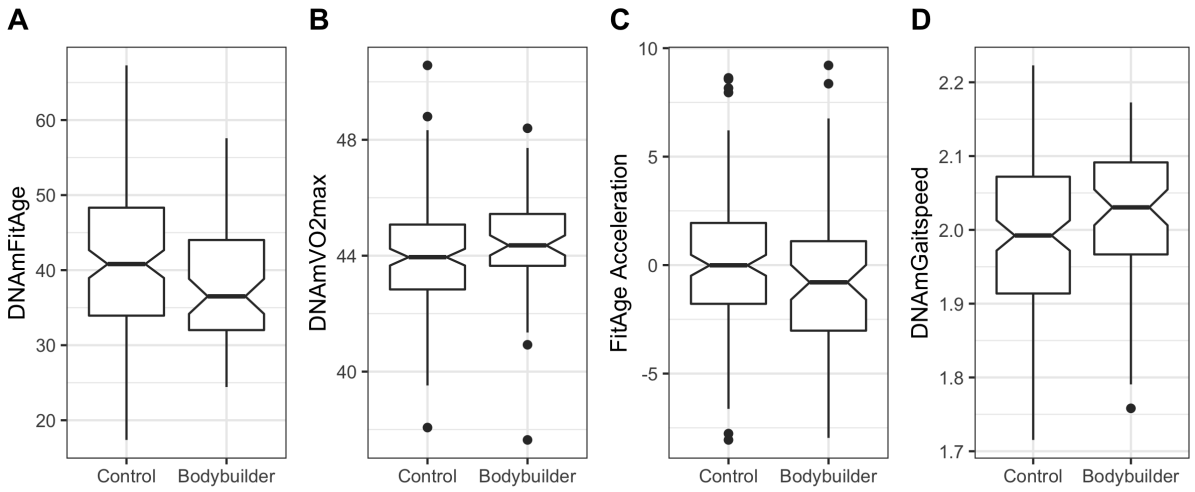


Figure 2.5: Boxplots showing spread of DNAm biomarkers between male controls (n=149) and male body builders (n=66) in the Polish Study. (A) DNAmFitAge is younger on average in the male body builders, (B) DNAmVO2max is fitter on average in the male body builders, (C) FitAge Acceleration and (D) DNAmGaitspeed are suggestively improved in body builders but not significantly different at 0.05.

interested in studying the epigenetic components to physical fitness.

DNAm biomarkers have been improved by incorporating phenotypic information [7, 12], however, DNAm biomarkers had not yet incorporated physical fitness. DNAmFitAge provides researchers a novel indicator of biological age which combines physical fitness and epigenetic health. This biomarker integrates the established DNAm prediction of mortality risk (DNAmGrimAge) with the newly developed DNAm predictions of fitness. Higher values of DNAmGaitspeed, DNAmGripmax, DNAmFEV1, and DNAmVO2max, which reflect greater physical fitness, correspond to younger estimated biological ages in men and women. We demonstrate physically fit lifestyles have younger biological ages and fitter DNAm fitness biomarkers, which we observe in people of low to intermediate physical activity levels across five large-scale validation datasets and in male body builders who have intense, athletic exercise regimes. Furthermore, FitAgeAcceleration is strongly associated with a host of age-related conditions and predicts time-to-death and time-to-CHD across validation datasets. FitAgeAcceleration provides a novel measure of epigenetic age acceleration that is expected to be particularly sensitive to exercise interventions.

We acknowledge the following limitations. First, the DNAm fitness parameter biomarkers lead to only modest improvement to estimate fitness parameters after including age and sex as covariates in validation datasets. This reflects the relatively weak signal present in blood for fitness parameters. Because of the biomarkers' limited correlation, DNAm fitness biomarkers should not replace true fitness parameters. Instead, the main benefit of our biomarkers is that they show blood epigenetic changes accompany physical fitness. These biomarkers advance the molecular understanding of exercise benefits, which we hypothesize to be most pronounced in athletic populations as illustrated in our analysis of body builders. The male body builders had a mean 2.7 year reduction in DNAmFitAge compared to controls, whereas the intermediate physically active people had at most a mean 0.33 year reduction in DNAmFitAge (WHI). Second, our DNAmVO2max biomarker was only validated in one dataset with VO2max; more research is needed to evaluate how our DNAmVO2max biomarker performs across a range of independent datasets.

Overall, DNAmGaitspeed, DNAmGripmax, DNAmFEV1, DNAmVO2max, and DNAmFitAge provide epigenetic components to evaluating a person's physical fitness. Physically fit people have a younger DNAmFitAge and younger FitAgeAcceleration, and younger values are associated with more physical activity and better age-related outcomes. Our research suggests

exercise and stronger fitness parameters are protective to DNAmFitAge in both sexes. We expect DNAmFitAge will be a useful biomarker for quantifying fitness benefits at an epigenetic level and can be used to evaluate exercise-based interventions.

3 Pseudo Copula Transformations for Large Scale Missing Data Imputation of Methylation Data

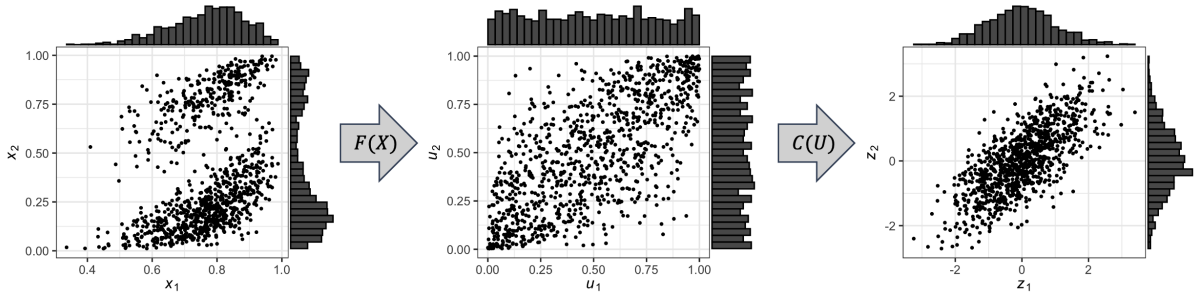
3.1 Motivation

As described in more detail in Section 1.1, DNA methylation data is measured using arrays and beta values can summarize the percent methylated at each CpG locus [9]. DNAm tend to arise from heterogeneous marginal distributions and both the beta distribution [10] and beta-binomial distribution [11] are commonly used for simulating DNAm data. For example, they are often concentrated near 0 or 1, skewed, and/or multimodal. Therefore, it is commonly accepted that methylation is not Gaussian in nature and requires flexible frameworks for modeling, however many statistical tools are built on the basis of Gaussian residuals which has limited the proper application of statistical tools to DNAm data. Furthermore, the transformation from beta to M-values may not result in a transformed margin that is close to Gaussian.

Missing DNAm & Imputation

Missing or poorly measured DNAm values commonly occur in samples from multiple factors. One reason is due to the presence of single nucleotide polymorphisms (SNPs), which can affect the hybridization of the probe to the DNA and cause inaccurate measurement of the DNAm value. Another factor is the large volume of measurements that need to be processed, which can result in technical errors, inconsistencies, and poor probe detection. Finally, different DNAm arrays measure different sets of CpG sites, which can cause missing values when comparing samples across arrays. Ignoring poor reads or missing values will increase the noise in the study, so it is important to fill in missing DNAm values (imputation) before downstream analyses.

For example, methylation-based biomarkers use beta values from a limited number of CpG sites across the human genome to construct their estimates. Hannum developed an age predictor based on 71 CpG markers from the whole blood tissue [43] and Horvath's model is a multi-tissue predictor relying on 353 CpGs markers [6]. Given the limited set of markers considered by these and other epigenetic clocks [20], the estimation accuracy is crucially dependent on the availability of the methylation levels for all the selected CpGs. Furthermore, it has been shown that imputing probes that are absent from the EPIC array but present in the 450K or 27K



Beta Scale Empirical CDF Transform Gaussian Copula Transform

Figure 3.1: Transforming bimodal CpG loci into Gaussian variables using empirical CDFs and a Gaussian copula.

arrays lead to underestimation of published ageing measures [44].

Most imputation tools are not explicitly built for DNAm data; they require error normality and DNAm data is not Gaussian in nature. Imputation tools that are designed specifically for DNAm data, such as the R package `methyLImp`, similarly assume error normality. Therefore, there is an unmet need to align the imputation tools with the inherent structure of DNAm data.

3.1.1 Copulas

Copulas are flexible multivariate models used to represent the dependence between random variables. The name copula comes from the latin word "link" introduced by Abe Sklar who developed the theoretical foundation for copulas [45]. This technique allows one to model the multivariate joint distribution among d variables using a well-defined distribution through a link function which doesn't require specifying the marginal distributions of the d variables. Today, Copula models are commonly used in quantitative finance [46] and high-dimensional statistical applications to model relationships between random variables. However, copula applications to bioinformatics have largely been ignored.

Copulas allow one to easily model and estimate the distribution of random vectors by estimating marginals and copulae separately and linking them through a transformation. This transformation is based on Sklar's theorem [45], where every continuous multivariate cumulative distribution function (CDF), $H(x_1, \ldots, x_d) = Pr(X_1 \leq x_1, \ldots, X_d \leq x_d)$ can be expressed by its marginals, $F_i(x_i) = Pr(X_i \leq x_i)$ and a Copula link function, C such that

$$H(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d)) \tag{3.1}$$

and when the density h exists,

$$h(x_1, \ldots, x_d) = c(F_1(x_1), \ldots, F_d(x_d)). \tag{3.2}$$

The Gaussian copula specifies an elliptical dependence structure through the variance-covariance matrix, which is commonly assumed in statistical procedures. There are different copula families and many parametric copula families which vary parameters to control the strength of dependence.

We leverage the separation of distribution margins and joint distributions through copulas to transform DNAm to Gaussian variables for use in statistical procedures. For each CpG site, we estimate the smoothed empirical CDF and inverse CDF. Then, we use these to transform each CpG into a normally distributed variable and impute missing values in this transformed space using popular imputation tools. Once values are imputed, we backtransform imputed values to the original beta scale. An example of a transformation converting a bimodal CpG locus into a Gaussian variable using the empircal CDF (F) and Gaussian copula (C) is provided in Figure 3.1. On the original Beta scale, bimodality may lead to complex patterns of dependence, which are potentially difficult to exploit for missing data imputation. Using a data-adaptive empirical CDF transform combined with a Gaussian Copula, we obtain patterns of dependence that follow a more familiar elliptically symmetric distribution.

Here, we develop an algorithm that transforms DNAm data into a Gaussian space using copulas to meet the statistical assumptions required for imputation and common statistical procedures and returns DNAm values on the original data scale for ease of interpretation. We demonstrate the use of our algorithm by imputing missing DNAm values.

In the following sections, we delineate the methodology and the multifarious aspects inherent to our algorithm. In the "Methods" section, we discuss the studies involved, articulate how we learn DNAm missingness patterns, and describe the process of mimicking this missingness for imputation analysis. This section will also detail the various methods we employ for imputation, share the specifics of our copula transformation algorithm, and explore the evaluations made to ascertain the efficacy of the imputation methods used. The subsequent "Results" section will focus on how our algorithm maintains the original distribution of the DNAm data and will shed light on the computational demands required, along with a comparative discussion on preferred imputation tools based on our findings. We will provide insights into the practicality and reli-

ability of the established methods, laying a comprehensive groundwork for understanding the operational intricacies and outcomes of our algorithm. Finally, the "Discussion" section encapsulates the benefits of our proposed method, along with an honest discourse on its limitations. We discuss how researchers in the field can integrate this method into their work, emphasizing the accessibility and adaptability of our open-source pipeline. We aim to provide a holistic view of our contributions, positioning them within the broader context of DNAm data analysis and highlighting the potential implications and applications in various scientific domains.

3.2 Methods

3.2.1 Datasets

CALERIE

Comprehensive Assessment of Long-term Effects of Reducing Intake of Energy (CALERIE) was a Phase 2 clinical trial studying young and middle-aged non-obese healthy adults [47]. CALERIE is the first clinical trial to focus on the effects of sustained caloric restriction in humans. Participants were randomized in a 2:1 fashion to 25% caloric restriction (CR) or ad libitum control group (diet is available at all times) in 2007 (N = 220). The intervention ran for two years across three sites. All participants needed to have a baseline body mass index (BMI) of 22-27.9 kg/m2 (lean to slightly overweight). An average of 12% caloric reduction was achieved in the CR group throughout the study. CALERIE data are available at https://calerie.duke.edu/samples-data-access-and-analysis [48].

CALERIE DNA methylation datasets were generated from banked whole blood DNA and adipose tissue samples by NIH grants R01AG061378 and 1U01AG060908 using Illumina EPIC 850k Arrays (Illumina Inc., San Diego, CA) as per the manufacturer's protocol. Blood DNA methylation assays were conducted by the Kobor Lab at the University of British Columbia. Adipose tissue DNA extraction and methylation assays were conducted by UCLA Technology Center for Genomics and Bioinformatics (TCGB). Adipose tissue methylation assays were ran by UCLA Neuroscience Genomics Core (UNGC). Quality control of sample handling included comparison of clinically reported sex versus sex of the same samples determined by analysis of methylation levels of CpG sites on the X chromosome. Methylation beta values were generated using the Bioconductor `minfi` package with Noob background correction.

Probes in either dataset were classified as poorly detected and therefore set to missing if they

had poor detection (p-value above 0.05) using the `minfi` package in `R`. For the blood dataset, a total of 608 observations and 845,645 probes were fully observed, and 20,446 probes had at least 1 poor detection whose missing pattern was analyzed. Details on analyzing missing patterns are provided below. The CALERIE blood dataset was used for analyzing missingness whereas the CALERIE fat dataset was used for imputation analysis.

For imputation analysis, the fat dataset was used and probes were excluded if any observation had a poor detection p-value; 91 observations and 856,697 probes remained for imputation analysis. We classified all 856,697 probes in the CALERIE fat dataset based on their relative distribution (Normal or Not Normal if Shapiro Wilks p-value was < 0.05 or ≥ 0.05, respectively), having a SNP in the probe or not, and median distance to SNP (either within 5 bp, within 20 bp, or over 20 bp away). The EPIC hg38 common SNP genomic annotation file was used to determine the presence and location of CpG loci to single nucleotide polymorphisms (SNPs).

FHS

Framingham Heart Study Offspring Cohort (FHS) is a longitudinal study that began in 1971 following the initial FHS study that began in 1948 in the town of Framingham, Massachusetts. The FHS cohort was initially composed of men and women aged 30 to 62 years who were free of cardiovascular disease (CVD) and were followed over time to investigate the causes of CVD [35]. All participants provided written informed consent at the time of each examination visit. The study protocol was approved by the Institutional Review Board at Boston University Medical Center (Boston, MA, USA).

Bisulphite converted DNA samples were hybridised to the 12 sample Illumina HumanMethylation450BeadChips [38] using the Infinium HD Methylation protocol and Tecan robotics (Illumina, San Diego, CA, USA). Peripheral blood samples were collected at the eighth examination samples (2005 to 2008). Genomic DNA was extracted from buffy coat using the Gentra Puregene DNA extraction kit (Qiagen) and bisulfite converted using EZ DNA Methylation kit (Zymo Research Corporation). DNA methylation quantification was conducted in two laboratory batches. Methylation beta values were generated using the Bioconductor minfi package with background correction. Sample exclusion criteria included poor SNP matching of control positions, missing rate >1%, outliers from multi-dimensional scaling (MDS), and sex mismatch. Probes were excluded if any observation had a poor detection p-value (p-value above 0.05) using the `minfi` package in `R`. In total, 2,544 observations and 455,200 probes remained for analysis. The 450K

common SNP and EPIC hg38 common SNP genomic annotation files were used to determine the presence and location of CpG loci to single nucleotide polymorphisms (SNPs). We observed more probes were mapped to the EPIC file (326,515) than to the 450K file alone (273,661), which is why we chose to use both. We classified all 455,200 CpG probes in the FHS dataset based on their relative distribution (Normal or Not Normal if Shapiro Wilks p-value was < 0.05 or ≥ 0.05, respectively), having a SNP in the probe or not, and median distance to SNP (either within 5 bp, within 20 bp, or over 20 bp away).

3.2.2 Examining True DNAm Missingness

Imputation tools commonly assume missingness is 'at random', and we hypothesized that DNAm missingness is non-random and related to known factors. If this was true, missing at random could not be assumed, and our imputation analysis would induce missingness based on the discovered non-random missing relationship instead of randomly throughout the dataset.

We explored the missingness pattern (determined by poor detection) within each CpG probe in CALERIE blood by examining probe distribution and presence and proximity to SNPs. To summarize findings, probes were binned into one of six categories based on missingness: All Measured (meaning no missing values), 1 poor (meaning 1 value missing in all 608 samples), $\leq 1\%$, $\leq 5\%$, $\leq 10\%$, and $> 10\%$. CpG loci were classified as normally distributed if the Shapiro Wilks p-value was > 0.05 and non-normal if it was ≤ 0.05. The EPIC hg38 common SNP genomic annotation file was used to determine the presence and location of CpG loci to single nucleotide polymorphisms (SNPs). If multiple SNPs were present in a single probe, the median distance to the SNP was taken.

3.2.3 Inserting Missingness

We use the observed missing patterns in CALERIE blood to design the missing pattern in the FHS dataset and CALERIE fat dataset for the imputation analysis. Global missingness (probes with any missingness) determined selecting 3% of probes to induce missingness which corresponded to 13,656 probes (3% of $455{,}200 = 13{,}656$). We chose to induce 1%, 5%, and 10% missing rates at 8,876, 3,414, and 1,366 loci, respectively, based on the relative frequency of missingness at those rates (Table 3.2). CpG loci were categorized based on normality, presence of SNP, and distance to SNP. Then CpG loci were randomly sampled within each category (without replacement) to match the relative frequency observed in CALERIE blood. The

Table 3.1: Missing Strategy in FHS and CALERIE Fat

Missing	Total CpGs	No SNP in Probe		SNP in 0-5 BP		SNP in 6-20 BP		SNP in 21-50 BP	
Rate	to Sample	Normal	Not	Normal	Not	Normal	Not	Normal	Not
1%	8,876	186	2,920	135	2,115	114	1,790	97	1,519
5%	3,414	61	963	60	943	47	741	36	562
10%	1,366	20	20	28	433	20	318	14	212

breakdown of number of CpGs sampled for each breakdown are provided in Table 3.1. For simplicity, 13,656 probes were also selected for inducing missingness for imputation in CALERIE fat, which corresponds to a global missing rate of 1.6%.

We randomly remove 1%, 5%, and 10% of the CpG values in the selected probes 5 separate times (different seeds for each loci) in the FHS dataset. This corresponds to removing 25, 128, and 255 samples. Because the CALERIE fat data is smaller (n=91), we chose to instead randomly remove approximately 5%, 10%, and 20% of the CpG values 5 separate times. This process results in having 5 different datasets to impute values in for FHS and CALERIE fat, but the loci needing imputation stays the same across the 5 iterations. By having repeated imputations at one locus, we improve precision in estimating imputation accuracy at a single locus which improves power to examine accuracy at a probe-specific basis.

3.2.4 Imputation Methods

Because many imputation procedures assume error normality, only the outcome of interest needs to follow a Gaussian distribution. Therefore, only probes with missing values need to be transformed into the Gaussian space for adequately meeting common imputation procedural assumptions. We devised 4 different scenarios to understand the operating characteristics associated with the use of our transformation method for DNAm imputation. Each method results in a different number of CpG loci being transformed. Our first method is referred to as "Missing Normal", where all probes with missing values are transformed into Gaussian variables. The dataframe to be imputed has 13,656 columns that are transformed and is fused with the non-missing untransformed CpG loci. The other three methods transform CpG loci if they are non-normally distributed based on a p-value threshold. Specifically, we chose common p-value thresholds for determining significance: 0.05, 0.01, and 0.001 and classified CpG's as normal or non-normally distributed based on those thresholds using the Shapiro Wilks p-value. CpG loci with missing values and Shapiro Wilks p-values that fall below the threshold are classified as non-normal and are transformed. We refer to each of these thresholding methods as

"Normal 05", "Normal 01", and "Normal 001" indicating CpGs were transformed if they were classified as non-normal at their corresponding p-value significance threshold and fused with the remaining untransformed missing CpG loci and untransformed non-missing CpG loci. A table summarizing the number of CpG loci transformed for each method and dataset are provided in Table 3.3.

These four methods are compared to the untransformed imputation approach, where the dataframe to be imputed does not undergo any transformation. This is how imputation is recommended to be performed [49] or compared [**?**], and we refer to this method as "Untransformed".

3.2.5 Imputation Tools

We explore imputing missing values in our datasets using three imputation tools in `R` that cover broad and different methods in their underlying imputation procedure. Specifically, we use `imputePCA`, `impute.knn`, and `methyLImp` from the `missMDA`, `impute`, and `methyLImp R` packages, respectively. We present only results using the first two tools in the body of this paper because of complications using `methyLImp` (explained below), however, a smaller analysis using `methyLImp` is provided in Appendix A.6. It is important to point out that our comparisons are not meant to exhaustively assess all imputation methods, but rather to evaluate the value of our transformation pipeline as we explore imputation methods with differing levels of model flexibility.

imputePCA

`imputePCA` imputes values using a regularized iterative principal components analysis [50]. `imputePCA` assumes error normality in the fixed effect model framework [50]. Initial values are drawn from a Normal distribution with mean and standard deviation calculated from the data. Then, PCA is performed on the completed dataframe, and values are imputed using the new principal components and loadings. This imputation process is repeated (PCA on dataframe, impute using new PC's) until the convergence threshold is met. Further explanation of this method is provided in Appendix A.4.1. We chose the number of principal components that explained 95% of the variance in the non-missing CpG loci, which corresponded to 1935 PC's in FHS and 81 PC's in CALERIE fat.

impute.knn

`impute.knn` was built for gene expression data and uses k-nearest neighbors to impute missing values [51]. This procedure finds k other non-missing CpG loci which have values similar to the CpG loci to be imputed (evaluated using Euclidean distance). The imputed value is a weighted average of the k loci with weights formed from similarity. Further explanation of this method is provided in the Appendix A.4.2. We chose k to be 50 neighbors based on the commonality and recommendations used in DNAm imputation. This procedure does not have distributional requirements and can safely be used for imputation regardless of the original distribution.

methyLImp

`methyLImp` was developed specifically for methylation data and builds a linear regression model via Singular Value Decomposition using completely observed data. It uses other samples as predictors (not CpG loci) and ignores the assumption of residual normality in the linear model specification. In addition, this method is known to have poor performance for methylation levels near 0.5 [49]. However, our methods prevented use of this function. `methyLImp` requires complete cases for observations, and our imputation procedure resulted in every observation having at least 1 missing value. We tried to circumvent this limitation by imputing on a smaller subset of the data at a time, however the computation time and resources required made using `methyLImp` infeasible. SVD is known to be computationally expensive, and we concluded other researchers would be more likely to use a different imputation tool than write code loops to similarly circumvent issues arising from using the method. We have included a small imputation analysis in the supplement.

3.2.6 Copula Transformation Algorithm

The proposed copula transformation pipeline takes place in 4 steps. First, we calculate the empirically smoothed CDF ($\widetilde{F}$) and inverse CDF ($\widetilde{F}^{-1}$) of select CpG loci. Second, we transform the select CpG loci to psuedo-Gaussian variables and return a fused dataframe (select CpG loci in the transformed space and the rest of the CpG loci in the untransformed, original scale). Third, we use one of the imputation tools described above to perform imputation. Fourth, we backtransform the select CpG loci that were imputed to the original beta value scale to return an imputed dataset on the desirable scale.

Let $\mathbf{X}$ be a $n \times p$ matrix of methylation data with n observations and p total columns where the first $1, \ldots, k$ columns have missing values and columns $k+1, \ldots, p$ have completely observed values. In the case of the "Missing Normal" tranformation method, $1, \ldots, k$ columns will be transformed. In the case of "Normal 05" or one of the other p-value thresholding methods, a subset of the k columns will be transformed, call these columns $1, \ldots, d$ for any one thresholding approach.

$$\mathbf{X} = \Big[\underbrace{X_1, \ldots, X_d}_{\substack{\text{missing CpG loci} \\ \text{to be transformed}}} , \underbrace{X_{d+1}, \ldots, X_k}_{\substack{\text{missing CpG loci} \\ \text{to stay on original scale}}} , \underbrace{X_{k+1}, \ldots, X_p}_{\text{non-missing CpG loci}} \Big]$$

The algorithm proceeds as follows:

For each j in $1, \ldots, d$ {:

1. Calculate the empirical smooth CDF, $\widetilde{F}_j(\cdot)$, and inverse CDF function, $\widetilde{F}_j^{-1}(\cdot)$.

2. Transform X_j to a pseudo copula variable, $\widetilde{Z}_j$

   Transform X_j to pseudo uniform variable, $\widetilde{U}_j$:

   $\widetilde{U}_j \sim \widetilde{F}_j(X_j)$

   Transform $\widetilde{U}_j$ to a pseudo normal variable, $\widetilde{Z}_j$:

   $\widetilde{Z}_j \sim \Phi^{-1}(\widetilde{U}_j)$

   }

   Collect $\{\widetilde{Z}_1, \ldots, \widetilde{Z}_d\}$ noting that each vector still has missing values.

   Return dataset for imputation,

   $\widetilde{\mathbf{Z}} = \left[\widetilde{Z}_1, \ldots, \widetilde{Z}_d, X_{d+1}, \ldots, X_k, X_{k+1}, \ldots, X_p\right]$

3. Impute missing values using preferred tool such as `imputePCA` or `impute.knn`

   $\widetilde{\mathbf{Z}}^*$ <- `impute.knn(`$\widetilde{\mathbf{Z}}$`)`

   $\widetilde{\mathbf{Z}}^* = \left[\widetilde{Z}_1^*, \ldots, \widetilde{Z}_d^*, X_{d+1}^*, \ldots, X_k^*, X_{k+1}, \ldots, X_p\right]$

4. Backtransform to obtain imputed dataset on original scale

   For each j in $1, \ldots, d$ {:

   Transform $\widetilde{Z}_j^*$ to a pseudo uniform variable, $\widetilde{U}_j^*$

   $\widetilde{U}_j^* \sim \Phi(\widetilde{Z}_j^*)$

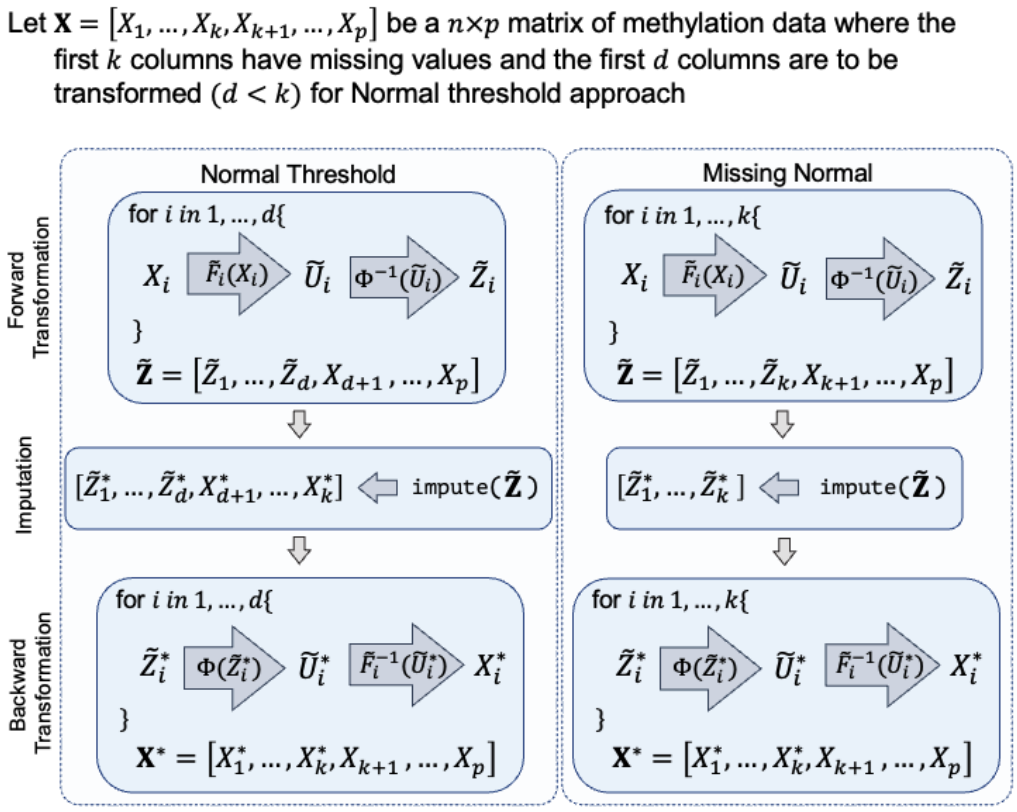


Figure 3.2: Overview of Transformation Methodology

Transform $\widetilde{U}_j^*$ to original scale of X_j

$$X_j^* \sim \widetilde{F}_j^{-1}(\widetilde{U}_j^*)$$

}

Collect imputed and backtransformed variables $\{X_1^*, \ldots, X_d^*\}$, imputed variables $\{X_{d+1}^*, \ldots, X_k^*\}$ and original non-missing variables $\{X_{(k+1)}, \ldots, X_p\}$ into $\mathbf{X}^*$, a $n \times p$ completely filled matrix.

$\mathbf{X}^* = [X_1^*, \ldots, X_k^*, X_{(k+1)}, \ldots, X_p]$ can now be used as the non-missing version of $\mathbf{X}$ that would be used for regression or other statistical procedures. This process is summarized in a diagram in Figure 3.2.

Step 1: Calculating $\widetilde{F}_i$ and $\widetilde{F}_i^{-1}$

The process to calculate functions needed to transform CpG values is a data driven process and is done by calling our function, `TransformDataset` (Appendix A.5.1) on the methylation dataframe with missing values set to `NA`. For each CpG loci i, we calculate the smooth density of the methylation beta values X, using the `density` function with a gaussian smoothing

kernel and parameters `n=1000` and `adjust=0.5`. The gaussian smoothing kernel uses 0.5 times Silverman's 'rule of thumb' for bandwidth [52], and using a smaller value of `adjust` would result in a less smooth density estimation. We chose the Gaussian kernel for convenience and efficiency. Additional details on the smoothing process are provided below in section 3.2.6. We convert the density to a function by interpolating between the 1000 points using cubic splines with `splinefun` function with the Forsythe, Malcolm and Moler (FMM) spline. Then, we integrate the density function using the `integrate` function to obtain vectorized values, Y_i. The empirically smoothed CDF, $\widetilde{F}_i$, is the interpolation between the set of input values (X_i) and vectorized values (Y_i). To obtain the empirically smoothed inverse CDF, $\widetilde{F}_i^{-1}$, we interpolate between the mapping of Y_i to X_i.

Step 2: Transforming to Psuedo Gaussian Variables

After the esCDF is computed, we can continue with the forward transformation into the Gaussian space. We construct *pseudo copula observations* by first transforming to a pseudo uniform variable and then to a pseudo normal variable. Pseudo uniform variables are computed one at a time by applying the esCDF: $\widetilde{U}_i = \widetilde{F}_i(X_i)$ for $i = 1, \ldots, d$. Each pseudo uniform variable is then transformed into a pseudo univariate normal variable using the normal inverse CDF: $\widetilde{Z}_i = \Phi^{-1}(\widetilde{U}_i)$ for $i = 1, \ldots, d$ (`pnorm` function). The output from this step corresponds to d pseudo-copula transformed CpG loci, which can be merged with the $p - d$ untransformed CpG loci into a single data frame for imputation.

The process to transform CpG values is done alongside calculating the forward transformation functions in Step 1 by calling our single function, `TransformDataset`. The code and corresponding R function are provided in the supplemental file (Appendix A.5.1) and our Github repository github.com/kristenmcgreevy/CONCORDANT. To use the function, the user specifies the dataset for transformation, the columns for transforming, and a name for the row names in order to merge the dataframe. The function outputs the inverse functions of each CpG loci needed to backtransform the d CpG loci and the transformed columns of the dataset ready for imputation (ie $\widetilde{Z}_1, \ldots, \widetilde{Z}_d$).

Step 3: DNAm Imputation

Once CpG loci are transformed into Gaussian variables, the transformed columns are combined with the untransformed columns for imputation. Imputation can be performed using

any imputation tool. We have selected to focus on two main imputation tools for comparison: `impute.knn` and `imputePCA` with limited results (for computational reasons) using `methyLImp` in the supplement. After imputation is performed on the dataset, we refer to the imputed DNAm loci with an $*$, which includes columns $\tilde{Z}_1^*, \ldots, \tilde{Z}_d^*, X_{d+1}^*, \ldots, X_k^*$.

Step 4: Back Transformation to Original β Scale

After imputation is complete, we back transform $\widetilde{Z}_1^*, \ldots, \widetilde{Z}_d^*$ variables to their original scaled methylation beta values. One locus at a time, we transform to a pseudo uniform variable: $\widetilde{U}_i^* = \Phi(\widetilde{Z}_i^*)$ using the standard normal CDF. Then we transform to the original scale using the esCDF from Step 1: $X_i^* = \widetilde{F}_i^{-1}(\widetilde{U}_i^*)$. Collect $X_1^*, \ldots, X_d^*$ as the CpG loci with missing values imputed on the original scale that can be used alongside $X_{d+1}^*, \ldots, X_k^*, X_{k+1}, \ldots, X_p$ for regression or other statistical procedures.

This process can be implemented calling our `BackTransformDataset` (Appendix A.5.2) function which requires the imputed dataframe and inverse CDF functions as inputs. The output is the completed dataframe referred to as $\mathbf{X}^*$.

Rationale and Details for Step 1

If we know the true CDF, F_i for random variable x_i, then $F_i(x_i) = u_i$, a uniform distributed variable and we can proceed. However, in practice we often do not know the true CDF. Instead, we can estimate the true F_i using the empirical CDF (eCDF), $\widehat{F}_i$, which is formed from the data ranks: $\widehat{F}_i(x) = \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}(X_j \leq x)$. However, because the eCDF measures the fraction of data observed that are less than or equal to a value, the function generally takes on a ragged staircase appearance, which makes the eCDF not differentiable. Smoothing the eCDF removes discontinuities, improves the function's mathematical properties, and captures the generalizable trend for variable x_i. We refer to the empirically smoothed CDF as esCDF and $\widetilde{F}$. Because $\widetilde{F}$ is a smooth, continuous, non-decreasing function, $\widetilde{F}_i^{-1}$ exists and can be estimated similarly [53]. We construct *pseudo copula observations* using the esCDF, $\widetilde{F}_i$: $\widetilde{u}_i \sim \widetilde{F}_i(x_i)$ and $\widetilde{z}_i = \Phi^{-1}(\widehat{F}_i(x_i))$.

In calculating the smooth density of x_i, the density is calculated within a window size of datapoints using a weighted sum. The weights are determined by the Gaussian kernel function. The result of this calculation is then used as an estimate of the probability density of the data points within the window. With the specifications laid out above, we estimate the density of the CpG locus by dispersing the mass of the empirical distribution over a regular grid of 1,000 points. The fast Fourier transformation is used to convolve this approximation with a

discretized version of the kernel and then uses linear approximation to evaluate the density at the specified points. This computes the kernel density estimate using the gaussian smoothing kernel with 0.5 times Silverman's 'rule of thumb' for bandwidth [52]. The bandwidth used is therefore $h = 0.5 \times 0.9 \times min(\hat{\sigma}, \frac{\text{IQR}}{1.34}) \times n^{-1/5}$, where $\hat{\sigma}$ is the standard deviation of the CpG loci, IQR is the interquartile range ($Q_3 - Q_1$), and n is the number of observed data points. Specifically, the probability density function f is estimated by the smooth kernel function f_n:

$$f_n(X) = n^{-1}h^{-1}\sum_{j=1}^{n} K\{h^{-1}(X - x_j)\} \tag{3.3}$$

where h is the bandwidth or smoothing parameter, K is the kernel function, and $x_1, \ldots, x_n$ are the DNAm methylation values for CpG locus i [52]. Because h is defined as above, this means the resulting smooth density is calculated as:

$$f_n(X) = \frac{1}{n \times 0.45 \times min(\hat{\sigma}, \frac{\text{IQR}}{1.34}) \times n^{-1/5}} \sum_{j=1}^{n} K\left\{\frac{(X - x_j)}{0.45 \times min(\hat{\sigma}, \frac{\text{IQR}}{1.34}) \times n^{-1/5}}\right\} \tag{3.4}$$

$$f_n(X) = \frac{1}{n^{4/5} \times 0.45 \times min(\hat{\sigma}, \frac{\text{IQR}}{1.34})} \sum_{j=1}^{n} K\left\{\frac{n^{1/5}(X - x_j)}{0.45 \times min(\hat{\sigma}, \frac{\text{IQR}}{1.34})}\right\}. \tag{3.5}$$

The choice of kernel function is not an important factor for kernel density estimation because it acts as the weight in the weighted sum of input points [54]. Therefore, the weight will be the same for all data points regardless of the kernel function used. The weighted sum is used to estimate the probability density of the data points within each window. As long as the same weights are applied to all input points regardless of their kernel function, then they can be used to estimate the same probability density. We chose the Gaussian kernel for convenience and efficiency.

We convert the density to a function by interpolating between the 1000 points using cubic splines, a method used to connect a set of data points that lie on a curve. The interpolated values are smooth and continuous. The interpolant is constructed as a piecewise cubic polynomial that passes through each given data point and its neighbors. We perform this using `splinefun` function with the Forsythe, Malcolm and Moler (FMM) spline. FMM splines fit an exact cubvic spline through the four points at each end of the data which is used to determine the end conditions. These have several advantages over other interpolation methods, such as being easier to compute, having fewer parameters to adjust, and providing more accurate results [55]. This method connects x_{i1} to x_{i2} with a spline where $x_{i1} < x_{i2} < \cdots < x_{in}$. Then, we integrate

the density function to obtain the empirically smoothed CDF using the `integrate` function. This integration results in vectorized values, y_i. To obtain the empirically smoothed CDF, $\widetilde{F}_i$, we interpolate between the set of input values and vectorized values, (x_i, y_i). To obtain the empirically smoothed inverse CDF, $\widetilde{F}_i^{-1}$, we interpolate between the mapping of y_i to x_i.

3.2.7 Evaluating Imputation Performance

We evaluate imputation performance between our transformed methods and the untransformed method using RMSE, Pearson's r, and Kolmogorov-Smirnov test. All evaluations are calculated in the original scale of the data (beta values), therefore imputed values using one of our four methods were back transformed prior to calculating these metrics.

Root Mean Square Error (RMSE) is one of the most popular metrics for missing data imputation performance assessment. It estimates total error between imputed (P_i) and true values (T_i) for CpG locus i with lower values indicate better performance.

$$RMSE(P_i, T_i) = \sqrt{\frac{\sum_{j=1}^{n_i}(P_{ij} - T_{ij})^2}{n_i}}. \tag{3.6}$$

Pearson's r or Pearson Correlation Coefficient measures the amount of linear correlation between the predicted and true values with values closer to 1 being better.

$$r_i = \frac{\sum_{j=1}^{n_i}(P_{ij} - \bar{P}_i)(T_{ij} - \bar{T}_i)}{\sqrt{\sum_{j=1}^{n_i}(P_{ij} - \bar{P}_i)^2}\sqrt{\sum_{j=1}^{n_i}(T_{ij} - \bar{T}_i)^2}}. \tag{3.7}$$

Kolmogorov-Smirnov test is a non-parametric assessment of how similar a sample of values are to a reference distribution [56]. We are interested in this metric to understand if the transformed method better preserves the true distribution of the imputed CpG loci. The reference distribution will be the empirical distribution of the ith CpG loci before missingness was inserted, $\hat{F}_i(X_i)$, and the distribution to compare will be the empirical distribution of the n_i imputed CpG values, $\hat{F}_{n_i}(X_i^*)$. The Kolmogorov-Smirnov (KS) statistic for CpG loci i is:

$$D_{n_i} = \sup_{X_i} |\hat{F}_{n_i}(X_i^*) - \hat{F}_i(X_i)|. \tag{3.8}$$

We conclude that the imputed sample is not from the reference distribution when $\sqrt{n_i}D_{n_i} > K_\alpha$, where K_α is the critical value of the Kolmogorov distribution. Imputed values are classified as adhering to the original CpG locus distribution if the KS statistic p-value is > 0.01 and otherwise

classified as not adhering to the original distribution.

RMSE and Pearson Correlation are calculated using the `gen_stat` function from the `methyLImp` package, and the KS test is performed using the `ks.test` function. We summarize imputation performance stratified by various features including median methylation levels, loci distribution (normal or non-normal), and presence of SNPs.

3.3 Results

3.3.1 DNAm Missingness Patterns

Poor probe detection and therefore missingness is not random across CpG loci in CALERIE blood; percent missingness of a probe increases if the CpG loci is not normally distributed, has more SNPs per probe, and is closer in proximity to SNPs. There were a total of 866,091 CpG loci measured on 608 samples via the EPIC v1 array, and 845,645 probes were fully observed with good detection across all samples (Table 3.2). This corresponds to 2.4% of all probes having some poor detection, which is similar to previously reported research which observed 3% probe missingness on average in DNAm data [49]. In total, there are only 0.1% of data points with poor reads because most (54%) of these CpG loci have only 1 bad read. 21% of fully observed probes are relatively normally distributed, whereas CpG loci with missing values are less likely to be normally distributed (5-8%), which is 13% less than probes with complete values. Fully observed probes have on average 1.5 SNPs per probe and are 20 base pairs (bp) away from the CpG loci to be measured. Probes with high levels of missing ($> 10\%$) have over 2 SNPs per probe and are less than 10 bp away on average.

These results informed our procedure for selecting CpG loci to insert missingness in and the missing rate to induce. While specifics of how missingness was inserted is provided in the Methods section, probes were randomly selected for our imputation analysis based on their normal distribution and distance to SNPs. More non-normal probes were selected for our imputation analysis, and missingess at probes were induced at 1, 5, and 10% missing rates to best match what is truly observed in real DNAm data.

3.3.2 Overall Imputation Performance

Our transformation methods improved or maintained imputation accuracy while significantly improving imputed DNAm value adherence to the original methylation distribution in both the

Table 3.2: Observed Missingness of CpG values in CALERIE Blood Study (n=608)

Probe Missingness	Number of CpGs	Poorly Detected CpGs	Relatively Normally Distributed		Average Number of SNPs per Probe	Median Distance to SNP
All Measured	845,645	-	179,071	21%	1.5	20
1 poor	11,029	54%	924	8%	1.6	18
$\leq 1\%$	5,612	27%	331	6%	1.8	14
$\leq 5\%$	2,120	10%	172	8%	2.1	11
$\leq 10\%$	603	3%	49	8%	2.2	10
$> 10\%$	1,082	5%	52	5%	2.2	9

FHS and CALERIE datasets.

Transformation Preserves Original Probe Distribution

Kolmorgov-Smirnov statistics demonstrate that our transformation method better preserves the original data distribution than untransformed methods using `impute.knn` and `imputePCA` for imputation. Interestingly, there is not one transformation method that outperforms the rest globally. In the FHS dataset, 11.7% more of imputed probe values adhere to the original distribution using the Normal 05 threshold with `impute.knn`. The untranformed approach, however, has only 3.6% of all imputed probes following the original distribution. For `imputePCA`, the Normal 001 transformation method in the FHS dataset performs the best with 61% of imputed probes adhering to the original distribution which corresponds to a 4.1% improvement over the untransformed imputation. Both the Normal 01 and Normal 05 methods also perform comparably. In CALERIE, using `imputePCA` with any of our transformed methods corresponds to 5.5-8.2% more probes adhering to the original distribution (Table 3.3). The Missing Normal transformation method corresponds with 86.8% of probes adhering to the original distribution.

The effectiveness of the transformation methods lies in their ability to counteract the forced symmetrization of the CDF curve near boundaries (methylation levels near 0 or 1) induced by untransformed imputation procedures, evident in Figure 3.3. Both `impute.knn` and `imputePCA` appear to be under and over-estimating the CDF curves at extremes. In contrast, the transformed methods' empirically smooth CDF overlay the original distribution (green) better than the untransformed method (purple). Our transformation approach effectively integrates probe-specific information and minimizes this forced symmetrization in the back transformation step, providing a more accurate representation of the original data distribution.

It is evident from these findings that the proposed transformation pipeline provide substantial improvement in CDF adherence in both the FHS and CALERIE datasets. Relying on empirical transformation methods, particularly the Normal 05 and Normal 001 proved effective

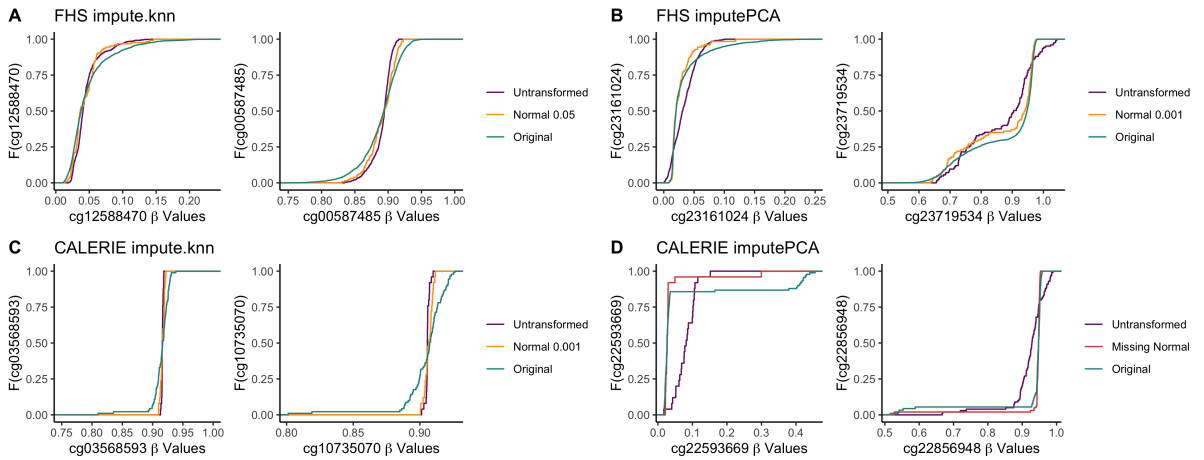


Figure 3.3: Comparison of imputed methylation eCDFs to original methylation eCDF

in better preserving the original data distribution while maintaining or improving imputation accuracy.

The proposed approach provides a more robust imputation that closely mirrors the original data distribution. Consequently, practitioners can expect more accurate and reliable imputations that maintains a stronger adherence to the original data distribution, as demonstrated in the FHS and CALERIE datasets using both `impute.knn` and `imputePCA`.

Impact of Transformation Methods on Imputation Accuracy

The median RMSE was maintained within each imputation tool regardless of the transformation condition, with RMSE deviating by 0.001. Such a marginal change in accuracy indicates each imputation tool has similar errors regardless if the data are transformed or not prior to imputation. Regarding median correlation, using any of the transformation methods with `impute.knn` has better performance in both datasets. The best performance uses the Normal 05 transformation in FHS (r = 0.691, +0.055 improvement) and the Missing Normal transformation in CALERIE (r = 0.508, +.17 improvement) compared to the untransformed imputation (Table 3.3). Median correlation using `imputePCA` is similar for any of the normal thresholding methods in FHS (r = 0.775) and CALERIE (r = 0.594) compared to the untransformed imputation, with less than 0.01 difference in correlation. However, the Missing Normal method performed worse than the untransformed method in the FHS dataset with a 0.043 decrease in correlation.

The distribution of imputed probes' correlations can be seen by dataset and imputation tool in Figure 3.4. The best normal thresholding method is plotted in yellow alongside the Untransformed and Missing Normal methods in red and purple, respectively. A substantial

Table 3.3: Imputation Performance Summary

Dataset	Imputation Tool	Loci Transformed	Dataset Form	Median RMSE	Median Correlation	Median Kolmogorov Smirnov (KS) Statistic	Adhering to Distribution (KS p>0.01)	CDF Adherence Improvement
FHS	impute.knn	0	Untransformed	0.020	0.636	0.255	3.6%	-
		13655	Missing Normal	0.019	0.681	0.191	13.6%	10.0%
		12535	Normal 05	0.019	0.691	0.221	15.3%	11.7%
		7025	Normal 01	0.019	0.691	0.222	14.6%	11.0%
		330	Normal 001	0.0195	0.643	0.261	5.5%	1.9%
	imputePCA	0	Untransformed	0.015	0.781	0.110	56.9%	-
		13655	Missing Normal	0.016	0.738	0.125	44.1%	-12.8%
		12535	Normal 05	0.016	0.774	0.129	60.3%	3.4%
		7025	Normal 01	0.016	0.774	0.129	60.3%	3.4%
		330	Normal 001	0.015	0.776	0.116	61.0%	4.1%
CALERIE	impute.knn	0	Untransformed	0.016	0.338	0.451	2.1%	-
		13655	Missing Normal	0.016	0.508	0.418	3.1%	1.0%
		12535	Normal 05	0.016	0.505	0.418	3.6%	1.4%
		10317	Normal 01	0.016	0.493	0.419	5.0%	2.9%
		8150	Normal 001	0.016	0.481	0.429	6.6%	4.4%
	imputePCA	0	Untransformed	0.014	0.592	0.250	78.6%	-
		13655	Missing Normal	0.013	0.593	0.246	86.8%	8.2%
		12535	Normal 05	0.013	0.593	0.246	86.6%	7.9%
		10317	Normal 01	0.013	0.594	0.247	85.4%	6.8%
		8150	Normal 001	0.013	0.595	0.250	84.1%	5.5%

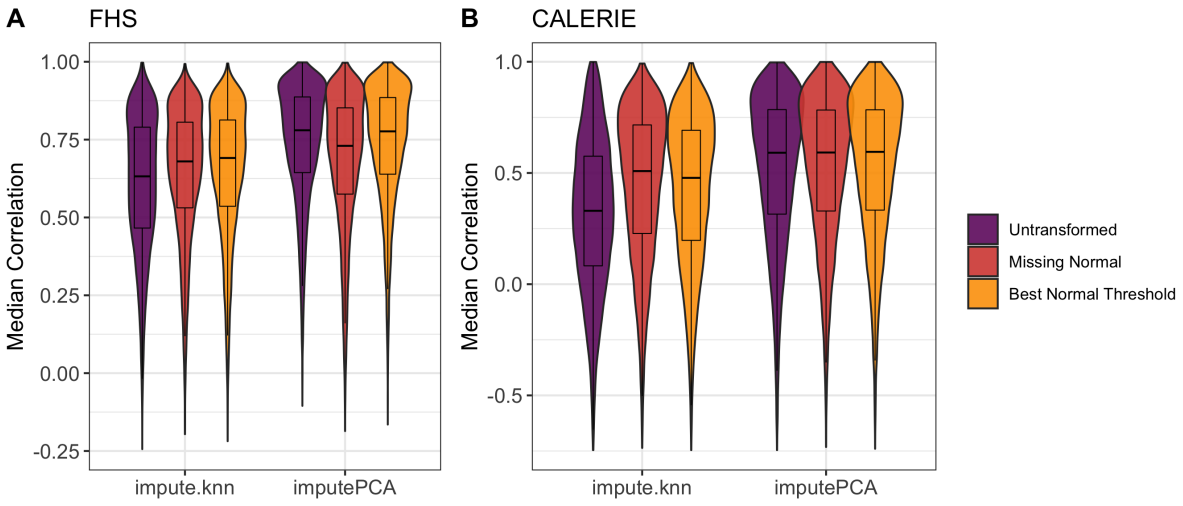


Figure 3.4: Comparison of Median Imputation Correlation by dataset (A) FHS or (B) CALERIE and imputation tool. Purple refers to imputating on the original DNAm scale, red to imputing with all missing probes converted to gaussian variables, and orange to imputing probes that have been transformed using a p-value threshold.

improvement in correlation is observed for either transformation method in `impute.knn` with comparative correlations between the Normal Thresholding and Untransformed methods with `imputePCA`. A better global correlation is observed using `imputePCA` over `impute.knn`.

Comparing Imputation Tools

The `imputePCA` tool demonstrated a substantially stronger correlation, lower RMSE, and better adherence to the original DNAm distribution than `impute.knn`. `imputePCA` consistently has lower RMSE values across all transformation methods, ranging from 0.013 to 0.016. In contrast, `impute.knn` ranges from 0.016 to 0.020 (Table 3.3).

`imputePCA` has a 0.145 and 0.254 better median correlation than `impute.knn` in the FHS and CALERIE data, respectively when comparing untransformed imputation procedures. The difference in median correlation between imputation tools is approximately halved when using one of our transformation methods. For example, the improvement seen using `imputePCA` over `impute.knn` is only 0.083 and 0.088 when using the Normal 05 approach in the FHS and CALERIE data, respectively. As such, our transformation methods appear to make imputation accuracy more robust and less sensitive to the imputation tool chosen to perform imputation.

Interestingly, the median KS statistic is much lower using `imputePCA` than `impute.knn` resulting in a larger percentage of imputed values having better adherence to the original CpG distribution. This is surprising because `impute.knn` does not have distributional assumptions

Table 3.4: Imputation Performance by Missing Probe Normality

Dataset	Imputation Tool	Dataset Form	Non-Normal Probes (p <0.001)			Normal Probes (p ≥ 0.001)		
			Median RMSE	Median Correlation	Correlation Difference	Median RMSE	Median Correlation	Correlation Difference
FHS	impute.knn	Untransformed	0.02	0.636	-	0.035	0.673	-
		Missing Normal	0.019	0.681	0.045	0.037	0.689	0.016
		Normal 05	0.02	0.697	0.061	0.0375	0.695	0.022
	imputePCA	Untransformed	0.015	0.781	-	0.025	0.818	-
		Missing Normal	0.017	0.717	-0.064	0.03	0.748	-0.07
		Normal 001	0.015	0.781	0.0	0.026	0.812	-0.006
CALERIE	impute.knn	Untransformed	0.016	0.344	-	0.017	0.327	-
		Missing Normal	0.015	0.551	0.207	0.017	0.444	0.117
		Normal 001	0.015	0.535	0.191	0.017	0.398	0.071
	imputePCA	Untransformed	0.013	0.611	-	0.015	0.566	-
		Missing Normal	0.012	0.615	0.004	0.015	0.56	-0.006
		Normal 001	0.012	0.616	0.005	0.015	0.567	0.001

in its imputation process, however `imputePCA` assumes normally distributed errors. Across both datasets, `impute.knn` corresponds to roughly 2%-16% of the imputed probes adhering to the original distribution whereas `imputePCA` has 44%-87% adherence.

In general, it can be noted that the `imputePCA` method shows a greater percentage of probes adhering to the original distribution in both datasets compared to the `impute.knn` method, indicating a better match to the expected distribution. These results, in addition to better accuracy compared to `impute.knn`, demonstrate potential superiority of `imputePCA` in handling missing DNAm data imputation.

3.3.3 Imputation Performance by Probe Properties

Probe Normality

Imputation accuracy using our transformation is generally improved more for non-normal probes than normal probes (Pearson Correlation in Table 3.4). Probes are classified as 'normal' if their Shapiro Wilks p-value is > 0.001 on the original beta value scale and 'non-normal' otherwise. For `impute.knn`, there is about 0.05 better correlation in FHS and 0.20 better imputation correlation in CALERIE when using our transformation on non-normal probes. This improvement is approximately twice the improvement seen when using our transformation on normal probes prior to imputing with `impute.knn` (0.02 in the FHS and 0.09 in CALERIE). There is almost no difference in imputation accuracy via RMSE or correlation when using `imputePCA`, except for worse correlation using the missing normal transformation method in FHS. As mentioned earlier, the accuracy of missing normal transform method for `imputePCA` in FHS performed worse, and both normal and non-normal probes have similar performance.

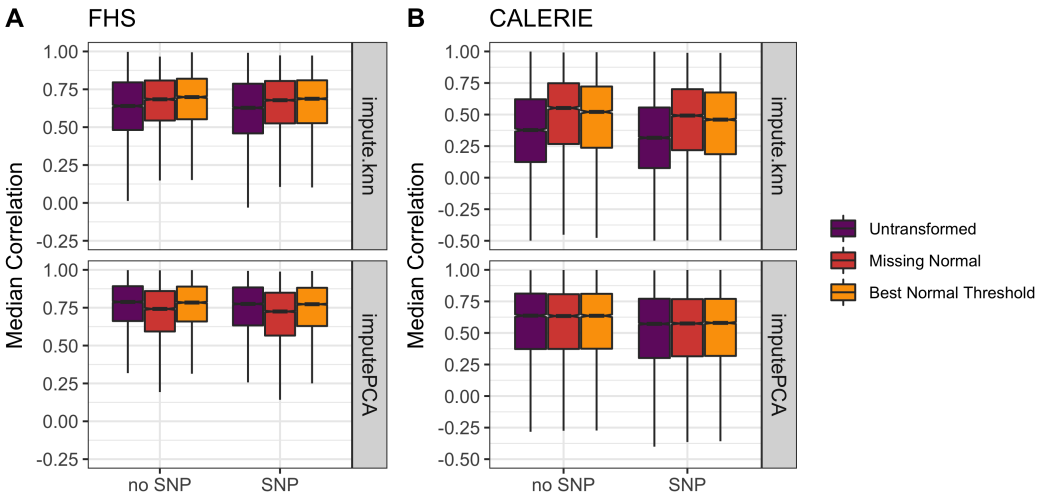


Figure 3.5: Correlation Accuracy by Presence or Absence of SNP in CpG probe

SNPs

Probes without SNPs have better imputation performance via RMSE and correlation, with approximately a 0.01 decrease in median correlation accuracy in FHS and 0.06 in CALERIE between probes without SNPs and with SNPs (Figure 3.5, Supplemental Table B.7). The larger difference in median correlation between SNP and non-SNP probes observed in CALERIE compared to FHS is likely due to the smaller dataset size. Distance to SNP appears to slightly decrease imputation accuracy if its within 5 bp away from the CpG locus, otherwise distance to SNP does not appear to change median imputation accuracy (Supplemental Figure B.2).

3.3.4 Computational Demand

All computation was performed using R version 4.1.0 and the Hoffman2 Cluster, which is a Linux compute cluster. The current peak CPU performance of the cluster is approximately 150 Trillion Floating Point, double precision, operations per second (TFLOPS). Some nodes have 36 cores and 192 GB of memory with computing capability between 4-7.5.

Transformation Functions

The forward transformation function, `TransformDataset` takes approximately 2 seconds to process each CpG locus in the FHS dataset and less than 0.01 seconds in the CALERIE dataset (Table 3.5). When employing the Normal 001 thresholding approach, this equates to a processing time of roughly 11-12 minutes for either dataset. However, with the Missing Normal

Table 3.5: Computational Demand for Forward Transformation

Dataset	Overall		Normal 001		Missing Normal	
	Time per CpG (seconds)	Storage per datapoint (bytes)	Time (minutes)	Storage (GB)	Time (hours)	Storage (GB)
FHS	1.97	64	10.8	0.053	7.46	2.2
CALERIE	0.09	80	11.6	0.059	0.32	0.10

approach, which involves transforming all missing CpG loci (totaling 13,656), the computation time extends to 7.5 hours for the FHS dataset and 20 minutes for CALERIE. The scalability of our forward transformation function is directly tied to the number of observations in the dataset, a result of the data-driven methodologies employed in crafting the transformation.

Considering that the function outputs both the transformed data and the inverse functions for the back transformation, it is essential to ascertain the storage requirements of this function. For each datapoint needing transformed, this function requires roughly 70 bytes of space in its output. In the context of the FHS data, this equates to approximately 0.17 megabytes per CpG locus and a significantly smaller 0.01 megabytes for the CALERIE dataset. Consequently, when applying the Normal 001 threshold, either dataset requires less than 0.1 gigabytes of memory. However, when employing the Missing Normal approach for the FHS dataset, the memory demand escalates to 2.2 gigabytes. Researchers can anticipate similar memory requirements in their dataset by calculating $70 \times$ (number observations) $\times$ (columns to transform)$/10^9$ = GB required.

The back transformation of imputed data (`BackTransformDataset`), on the other hand, is comparatively uncomplicated and economical in terms of computational power. In this phase, no further development of functions or algorithms is necessary. On average, the back transformation requires 2.6 milliseconds per CpG locus for the FHS dataset and less than 1 millisecond per CpG locus for CALERIE. This translates into approximately 1 second for FHS and 7.3 seconds for CALERIE when the Normal 001 threshold is used.

Imputation Tools Computational Demand

We observed considerable differences in the computational demands for `impute.knn` and `imputePCA`, with `impute.knn` requiring substantially less computational resources. For the FHS dataset, `impute.knn` required about 1.6 hours of computation time and used 64.3 GB of memory. In contrast, `imputePCA` showed significantly higher computational demand, necessitating 28.7 hours of time and 99.6 GB of memory. Using `impute.knn` corresponds to over 17 times re-

duction in time and 1/3 the memory requirements compared to `imputePCA`. On a per data point basis, this translates to roughly 3 and 50 microseconds needed for `impute.knn` and `imputePCA`, respectively. As expected, there is a higher computational demand (time and memory) on the larger FHS dataset compared to the smaller CALERIE dataset. With CALERIE, `impute.knn` required approximately 0.05 hours and 16 GB of memory, whereas `imputePCA` required 1.3 hours and slightly less memory (15 GB). These results are summarized in Supplemental Table B.8.

In summary, `impute.knn` exhibited less computational demand across both datasets. `imputePCA`, on the other hand, demanded more computational resources, especially in terms of time.

Our results underline the importance of carefully considering the balance between computational demand and imputation accuracy in DNAm studies. It was evident that the choice of the imputation tool, the dataset size, and the use of additional transformation functions significantly impact the computational demand. However, the incorporation of our transformation functions, although computationally expensive, can lead to substantial improvements in imputation quality, justifying the increased computational load.

3.3.5 Open Source Pipeline

In our endeavor to facilitate and streamline the handling of missing data in DNA methylation (DNAm) studies, we have designed an accessible and user-friendly coding pipeline. This pipeline incorporates our novel transformation and backtransformation functions, effectively assisting researchers in transforming and imputing DNAm data for increased accuracy. Functions and sample code can be found freely available at https://github.com/kristenmcgreevy/CONCORDANT and in the Appendix Section A.5.

To begin with, researchers need to process the raw DNAm data using detection p-values. In this initial step, probes that are poorly detected, indicated by a high detection p-value, should be set to NA. Researchers should decide which transformation method they want to incorporate, such as Normal 001, etc. Researchers can use our `TestNormalityofMissingCols` function (Appendix A.5.3) on their dataset to determine which CpG loci have missing values and are non-normal based on the preferred p-value threshold. Once this preprocessing is done, the data is ready for transformation.

The first function in our transformation pipeline is `TransformDataset` (Appendix A.5.1). This function is data-driven and performs a forward transformation on the specified columns,

which in our case are each CpG locus. Specifically, it leverages the inherent data distributions to convert the DNAm data into a standard normal scale, making it more conducive for common statistical procedures. While the performance of this function scales with the number of observations in the dataset, it typically executes within two seconds per CpG locus for larger datasets.

Following the transformation process and subsequent imputation, the imputed DNAm data can be backtransformed to its original scale using our `BackTransformDataset` function (Appendix A.5.2). This function is relatively efficient, and doesn't require the development of new functions or algorithms, making the backtransformation process both simple and computationally inexpensive.

By implementing this pipeline, researchers can effectively manage and analyze DNAm data. The primary advantage of this approach is the improved imputation accuracy, which, in turn, enhances the reliability of subsequent analyses and conclusions. We encourage researchers to use this open-source pipeline in their DNAm studies and contribute to the collective advancements in this field.

3.4 Discussion

Our research delves into the efficacy of applying transformation methods prior to imputing DNA methylation values with imputation tools like `impute.knn` or `imputePCA`. We find that our copula-based transformation strategies, especially Normal 05 and Normal 001, dramatically improved preservation of original data distribution in DNAm imputed values. By acknowledging the inherent non-symmetry in the methylation distribution and transforming the data accordingly before imputation, our method curtails underestimation at the edges, a pitfall observed in these imputation methods. The backtransformation process helps to mitigate the issue of forced CDF symmetrization near extremes (methylation levels near 0 or 1). Thus, any forced symmetrization arising from the imputation procedures in the transformed distribution is effectively removed in the back transformation phase. Consequently, our methodology, unlike its untransformed counterparts, leads to a more representative and accurate depiction of the methylation values.

Our study encompassed various tissues and arrays, using differing dataset sizes to establish robustness. Unlike current research, which often assumes data is missing at random, we conducted a thorough exploration of actual missingness patterns in DNAm data to inform our

analysis. Moreover, we opted for real datasets over simulated DNAm to better reflect conditions that researchers may encounter in actual analyses. Our discovery that the missingness of CpG sites can be partially explained by probe normality suggests potential bias in imputation results or analyses that employ standard statistical methods across randomly imputed loci. Consequently, our results, derived from realistic conditions and considering true missingness patterns, offer a more robust perspective.

While our study did not definitively identify an "optimal" p-value threshold for transformation, it provides evidence that applying transformations at common thresholds—such as 0.05 or 0.001—enhances correlation accuracy and adherence to the original probe distribution. With our data-driven approach, we integrate probe-specific information into the imputation process, a step that is overlooked when solely relying on neighboring values in untransformed methods, such as in `impute.knn`. By transforming non-symmetric, extreme methylation values into a symmetric normal distribution, we ensure that the initial sampling draw from `imputePCA` aligns with the correct distribution, thereby negating any forced symmetrization during imputation.

This pioneering integration of copula models into bioinformatics—although in our case, limited to Gaussian copulas—suggests further scope for improvement if other copulas are explored. While the Gaussian copula allows for elliptical dependence structure needed by many imputation tools, it fails to capture dynamic changes over time. Adopting alternative copulas that accommodate evolving CpG dependencies could further refine the model.

While our research was conducted within the realm of DNAm, it's crucial to highlight that the transformation functions we tested are generalizable to any form of continuous data that requires a Gaussian distribution. This universality expands their potential applicability well beyond the confines of epigenetic studies, to virtually any field that grapples with the challenges of handling continuous data. To facilitate their broader use, we have made the code and functions freely accessible to researchers in the supplement and on GitHub. Our findings not only shed light on the challenges of DNAm imputation but also open a pathway for these transformation methods to be applied and evaluated in other bioinformatic fields. Future research should extend these transformation methods across a broader spectrum of continuous data types and imputation scenarios, thereby further enhancing their potential impact.

Our transformation approach provides a more robust imputation that closely mirrors the original data distribution. Consequently, our transformation method provides a more accurate and reliable imputation that maintains a stronger adherence to the original data distribution, as

demonstrated in the FHS and CALERIE datasets using both `impute.knn` and `imputePCA`. Given these outcomes, we strongly recommend researchers consider implementing our transformation methods prior to imputing DNAm values. This approach will likely enhance the accuracy of their imputed values and the validity of subsequent analyses, leading to more robust and reliable results in epigenetic studies.

4 Cross Tissue DNAm Biomarker Prediction using Transfer Learning

4.1 Motivation

DNA methylation (DNAm) is an epigenetic mechanism that varies across tissues and contributes to cell type, regulates gene expression, and influences disease states [8]. DNAm continues to be studied offering a window into the biological processes and aging within cells [27]. Traditionally, blood has been the tissue of choice for DNAm biomarker development, serving as a versatile medium that interacts with and carries information from an array of organs. However, this choice is not without its limitations. The performance of DNAm based biomarkers is inherently tied to the relevance of the tissue that DNAm is measured in [57, 58], and biomarkers built with tissues more related to the condition or trait of interest are likely to be more accurate and informative.

Yet, herein lies a significant challenge—the tissues most pertinent to certain diseases or conditions, such as the brain, adipose, and bone, are often the hardest to access. Their collection is typically invasive, painful, and expensive, leading to small sample sizes. This scarcity significantly limits the development of tissue-specific DNAm biomarkers. While some research has looked at estimating methylation across tissues [59] or predicting species' average methylation across tissues [60], algorithms are not available for individual level prediction or without needing access to multiple tissue data. Moreover, simply measuring these tissues and applying current DNAm biomarkers may not yield meaningful insights, a point underscored by researchers who advise caution when using methylation markers from surrogate tissues [61].

This landscape delineates two explicit needs in the field of DNAm research. First, there is an urgent requirement for using more accessible tissues to accurately measure biomarkers of interest [57, 62]. Saliva emerges as a promising candidate, offering a non-invasive and easily accessible alternative to blood [63]. Its collection is patient-friendly and uncomplicated, allowing for larger sample sizes and broader applications. Second, there is a pressing need for novel methodologies that can develop accurate biomarkers in tissues traditionally challenged by inadequate sample sizes [64, 62].

Moreover, this context highlights an area for innovation: the use and understanding of Trans-

fer Learning (TL) in the realm of DNAm research and biomarker development. TL, a powerful technique in machine learning, is adept at leveraging small data sizes and applying knowledge from one context to enhance precision in another [65]. TL has been used in bioinformatics for tasks like imputing the methylome in low-coverage cases [66] and augmenting gene expression data [67], but its application in DNAm biomarkers, particularly for cross-tissue prediction, has yet to be explored.

Our study aims to address these needs from three angles. We demonstrate how transfer learning can improve DNAm biomarker accuracy, especially with limited data sources. We introduce cross tissue prediction algorithms for estimating common blood DNAm biomarkers using saliva DNAm. Additionally, our study offers practical tools and guidelines for researchers, empowering them to implement TL methods and develop algorithms tailored to their specific biomarker interests. Through these contributions, we aspire to provide easily usable methods for researchers to better estimate their favorite biomarkers across different tissues by leveraging shared information from other tissues, as well as methods to develop more accurate biomarkers in typically inaccessible tissues by combining data from similar sources.

4.1.1 Common Transfer Learning Terminology and Definitions

The following section lists notation and definitions commonly used in transfer learning, and the notation is consistent with other papers outlining methodology [68, 69].

Domain, D: A domain is defined by two components: a feature space X and a marginal probability distribution $P(X)$. In the context of our study, the feature space X represents DNA methylation levels, with $X = \{x_1, \ldots, x_p\} \in X$ indicating the set of methylation loci.

Task, T: A task is characterized by an outcome space Y and a predictive function $f(\cdot)$ or $P(Y|X)$, which is derived from the relationship between feature vectors X and outcomes Y. In our research, Y signifies DNAm biomarkers, making our task the prediction of these biomarkers in tissue m based on methylation levels in tissue l ($P(Y_{mi}|X_{li})$).

Source Domain, D_S, and Target Domain, D_T: The source domain D_S comprises data $\{(x_{Si}, y_{Si})\}$ from which knowledge is transferred, with $x_{Si} \in X_S$ and $y_{Si} \in Y_S$. The target domain D_T, where this knowledge is applied, consists of data $\{(x_{Tj}, y_{Tj})\}$.

Heterogeneous transfer learning occurs when $X_S \neq X_T$, indicating differences in covariates between source and target domains. Homogeneous transfer learning occurs when $X_S = X_T$ and either $P(X_S) = P(X_T)$ or $P(X_S) \neq P(X_T)$. In the case where $P(X_S) \neq P(X_T)$, the source and target domains differ in their marginal probability distributions. Our setting involves the same methylation sites across different tissues ($X_S = X_T$), but with distinct covariance structures in X, representing tissue specific methylation, resulting in $P(X_S) \neq P(X_T)$.

Finally, classical transfer learning is single source, single task, referring to one source domain informing a single task [70, 71]. In contrast, multi-source TL uses multiple sources or datasets to improve a task. This approach introduces additional heterogeneity among the data, amplifying the complexity of the learning process [68, 72]. A critical consideration in multi-source TL is the avoidance of 'negative transfer', the situation where incorporating source information actually worsens the target model performance. Various methodologies have been developed to assess the informativeness of source samples and effectively integrate them with the target to mitigate negative transfer [72, 73, 74, 67, 75]. We implement and test methodologies for our unique context, employing multiple cross tissue datasets (sources) to enhance the saliva to blood biomarker prediction accuracy. Our implementation explores a nuanced approach to gauge the relevance of source data, recognizing that its informativeness may vary across different datasets, tissue combinations, and specific DNAm biomarkers. This careful examination ensures the transfer learning techniques are tailored to the intricacies of DNA methylation.

4.1.2 Data Distinctions to Classical Transfer Learning

Our study falls within a unique category of 'multi-source, multi-target, high-dimensional, homogeneous transfer learning', however our data has novel characteristics not typically observed or studied in TL.

Simultaneous Use of Similar and Dissimilar Data

While conventional transfer learning might focus on domains with significant overlap or similarity, our approach blends data across various tissue types, some of which might share more characteristics than others. This heterogeneity is distinct in that it involves not only different domains (tissues) and different relationships within domains (tissue specific methylation patterns), but also different relationships across two different, distal tissues. For example, how adipose can predict muscle methylation can be different from how buccal can predict brain

methylation in an individual. Despite having common CpG loci across tissues, the differing covariance structure across tissues complicates predicting DNAm biomarkers from one tissue to another as the biological contexts are no longer held constant. By predicting across tissues and incorporating data from different cross tissue predictions, we are seeking to capture universal within-person shared tissue signals. This set-up recognizes that inter-tissue signaling exists and biological processes can be shared in different tissue environments. We trust that shared patterns across tissues and biomarkers exist, and our algorithm is designed to discern these patterns, even in distally related tissues.

Multiple Source and Target Data

Our target datasets, focusing on saliva DNAm predicting blood DNAm biomarkers, are aggregated from six different studies. This multi-study integration is uncommon in transfer learning, where target data is usually drawn from a single or more uniform source [75, 73]. The diversity in our target datasets adds complexity due to variations in study design, data collection methods, and participant characteristics.

As we highlight later, what is typically called "source" datasets, we refer to as "*auxiliary*" datasets instead, alluding to the potential of the data to be superfluous and uninformative similar to auxiliary statistics.

Adaptation in High-Dimensional Settings

Addressing the high-dimensional nature of DNAm data, where the number of predictors p exceeds the number of observations n, adds complexity. In this situation, the sparsity of X needs learned, demanding robust model optimization [67, 75]. Our penalized regression framework is designed to adapt to these settings, focusing on discerning shared patterns across tissues, despite their diverse biological and environmental contexts.

Final TL remarks

Our approach uniquely embraces the heterogeneity of both domains and tasks. We do not limit our analysis to similar biological contexts and instead structure our TL framework to capture shared information across multiple distally related tissues. By developing algorithms that predict DNAm biomarkers across different tissues and incorporating data from multiple cross-tissue predictions, we aim to capture universal within-person shared tissue signals. This methodol-

ogy requires sophisticated data handling and model optimization to manage the heterogeneity across sources and domains. We explore the optimization of TL techniques for complex epigenetic problems with methods simple enough that other researchers can readily adopt. This approach offers an opportunity to unveil correlations and patterns in DNAm across various tissues and advance methodologies in epigenetic research.

4.2 Methods

The core aim of our study was to assess the utility of Transfer Learning (TL) in the context of DNA methylation (DNAm) and DNAm-based biomarkers. Our analytical strategy is predominantly built upon the translasso framework [67]. In our research, we prioritized the this class of algorithms due to their compatibility with the widely utilized Lasso regression techniques [39]. This choice ensures that our methods can be seamlessly integrated by researchers accustomed to Lasso regression, thereby promoting wider adoption within the field. We employ a transfer learning paradigm for high-dimensional linear regression, utilizing prediction and estimation procedures outlined in their study to investigate whether transfer learning can be utilized in high dimensional epigenetics to improve DNAm biomarker prediction across tissues.

Our methods section has the following format. First, we describe the basic model set-up and datasets used. Then, we describe the transfer learning methodology that we employ. After that, we develop several methodological variations aimed at optimizing parameterization and TL application to our cross tissue DNAm setting. We then describe how we classify optimizations and develop the final cross tissue prediction algorithms. Finally, we describe the validation process for testing our algorithms in outside datasets.

4.2.1 Subsetting Potential Covariates: C+S and C Method

In our study, we developed two distinct algorithms for predicting DNA methylation (DNAm) biomarkers across different tissues. The first algorithm, referred to as the C+S method, combines both the DNAm biomarkers derived from saliva and the raw methylation values from saliva samples. This approach allows for the saliva DNAm biomarkers to be 'updated' via methylation loci, enhancing their predictive accuracy. The terminology "C+S" references **C**pGs and **S**aliva DNAm Biomarkers being included as covariates. The second algorithm, known as the "C" method, exclusively uses saliva methylation values to predict blood DNAm biomarkers. This CpGs-only strategy is especially relevant when creating new biomarkers or when faced with datasets that lack a significant number of CpGs necessary for computing DNAm biomarkers, often due to variations in array platforms. By focusing solely on saliva methylation values, the C method provides a valuable tool in situations where comprehensive CpG data is unavailable or when developing novel biomarkers.

We restrict the potential covariates to predict each blood DNAm biomarker to CpG loci

known to be informative to epigenetic clocks and conserved across commonly used methylation arrays. These loci are those published in public DNAm epigenetic clocks, specifically DNAmAge [6], DNAmAgeSkinBloodClock [76], DNAmHannumAge [43], DNAmPhenoAge [12], DNAmFitAge [20], Zhang [64], MethylDetectR [77], EpiTOC [78], and the PanMammalianClock [79]. Because some loci are not available across different array platforms, we included only CpG loci conserved across the 450K and EPIC array. This resulted in 6662 unique CpG loci to be included. For our C+S method, our potential covariates included 6663 variables: 6662 methylation sites and the saliva DNAm biomarker. For the C method, we further reduced the methylation sites included to only those conserved across the 450K, EPIC, and Mammalian40K array. This resulted in 1307 unique CpG loci to be potential covariates in our C method. This is not only conducive to the development of human biomarkers but also holds potential for direct application to animal studies because the loci are conserved across mammalian species. For example, animal models measuring DNAm under calorie restriction (CR) could be integrated into our research to better inform the conserved epigenetic changes in humans from CR. Overall, developing and comparing algorithms in these two cases informs us of information loss or gain when implementing TL in DNAm contexts.

4.2.2 Datasets

Our "target" data includes the datasets that have DNAm values in saliva and blood. These datasets include the tissues of the targeted research objective: predicting blood DNAm biomarkers from saliva DNAm. We refer to "auxiliary" data as datasets that include DNAm in two tissues that are not saliva and blood. Finally, we include validation datasets, which are datasets not used in the TL algorithm development stage, but instead are for testing our developed algorithms in. All the datasets used have been previously described elsewhere; we provide brief summaries here.

Target Data

Our target data consisted of 6 independent datasets that included DNAm in saliva and blood tissues for the same individuals. GSE111165 (n=33), GSE214901 (n=19), GSE159899 (n=19), GSE130153 (n=22), GSE59507 (n=4), and GSE73745 (n=12) for a total of 109 samples in the target datasets. In developing our methods, we employ a Leave-One-Data-Out (LODO) methodology that builds the TL model in 5 of the target datasets and tests in the 1 held out

target dataset. This process is repeated for each target dataset, resulting in 6 LODO iterations per method, and the weighted average by target dataset size is used to calculate the final LODO estimate.

GSE111165 collected samples from blood, saliva, buccal, and brain tissue in epilepsy patients undergoing brain resection. DNAm was measured with both the 450K and EPIC arrays. GSE214901 measured brain, blood, saliva, and buccal in Japanese individuals undergoing neurosurgery, aged 13-73. DNA methylation was measured using the EPIC array. GSE159899 collected and measured methylation in whole blood, saliva, and T-cells using the EPIC array. GSE130153 measured saliva and blood sample methylation with the 450K array. GSE59507 measured multiple tissues from male crime scene samples aged 20-59 including blood, saliva, and semen. Methylation was measured using the 450K array. GSE73745 measured methylation in people with respiratory allergies and healthy controls in saliva and mononuclear blood cells. Methylation was measured using the 450K array.

Auxiliary Data

Our auxiliary datasets consisted of 5 independent datasets that included DNAm measured in two different tissues for the same individuals. Comprehensive Assessment of Long-term Effects of Reducing Intake of Energy (CALERIE) included muscle and adipose (n=130), GSE111165 included buccal and blood (n=27), GSE214901 included buccal and brain (n=19), TwinsUK included adipose and skin (n=136), and GSE48472 included fat and blood (n=6) for a total of 318 samples in the auxiliary datasets. For each of these datasets, they are presented where the first tissue listed is the tissue used as the potential covariates, and the second tissue is the tissue used for the outcome. For example, in CALERIE, muscle DNAm is used in the X and adipose DNAm biomarkers are used in the Y. In the TwinsUK data, original person ID's were not available for each tissue dataset, so individuals were matched across tissue based on SNPs, age, BMI, twin zygosity, and matching family IDs.

Validation Datasets

We use three datasets that measure phenotypic variables known to relate to DNAm biomarkers and DNA methylation. This included GSE119078, GSE148000, and GSE149747. The first has 59 saliva samples measured in males (n=25) and females (n=34) with and without Celiac's Disease on the 450K array. No differences were observed by disease group to the saliva DNAm

biomarkers, and to our best knowledge, this has not been observed elsewhere. This dataset is used to validate the relationship between sex and telomere length. The second dataset includes 26 asthma, COPD, and healthy patients where DNAm was measured in sputum using the 450K array, which is similar to, but not identically saliva. Age was observed to be different between the three disease classifications, so all models include age as a covariate. This dataset was used to look at differences in predicted DNAm biomarkers to disease status, relate predicted blood cell count models to measured lymphocyte percentage, and lung health DNAm biomarkers to reported cumulative smoking pack years. The third dataset is a exercise, diet, and sleep intervention study with saliva DNAm measured at baseline, 4 weeks, and 8 weeks after intervention on the EPIC v1 array. The HorvathHIV data measured methylation with a custom array (HorvathMammalMethylChip) from 661 samples across 11 human tissues (adipose, blood, bone marrow, heart, kidney, liver, lung, lymph node, muscle, spleen and pituitary gland) [58]. This sample included 133 clinically characterized, deceased individuals, including 75 infected with HIV. Based on the results of the initial study, we restricted our analysis to tissues with substantial sample sizes, relevant to saliva, and accessible- being lymph, muscle, and adipose.

When available, DNAm was processed using bioconductor package in R with Noob normalization. This was not possible for GSE130153 which provides only processed DNAm data and not the original idat files. In addition, 3 datasets (GSE73745, 159899, and 130153) did not provide chronological age, which is required for some DNAm clocks. In these cases, their predicted age from DNAm was used in place of their chronological age. After pre-processing, DNAm data was uploaded to the online DNAm clock calculator to calculate DNAm biomarker values (https://dnamage.clockfoundation.org). The resulting epigenetic clocks were used as end points in our prediction models.

4.2.3 Transfer Learning Methodology

We are interested in estimating β that maps DNAm from saliva to DNAm biomarkers in blood. The model of interest is

$$y_j^{(0)} = x_j^{(0)\top}\beta + \epsilon_j^{(0)}$$

where y_j is the jth blood DNAm Biomarker, x is the vector of saliva DNAm inputs, and (0) refers to the target data. Because we develop two different algorithms for each DNAm biomarker, for the C+S method, x includes the jth saliva DNAm biomarker and saliva methylation values

Table 4.1: Dataset Summary

Purpose	Dataset	N	Tissues
Target	GSE111165	33	saliva, blood
	GSE214901	19	
	GSE159899	19	
	GSE130153	22	
	GSE59507	4	
	GSE73745	12	
Auxiliary	CALERIE	130	muscle, adipose
	GSE111165	27	buccal, blood
	GSE214901	19	buccal, brain
	TwinsUK	136	adipose, skin
	GSE48472	6	adipose, blood
Validation	GSE119078	59	saliva
	GSE148000	26	sputum
	GSE149747	122	saliva
	HorvathHIV	28-58	lymph, adipose, muscle, blood
	TwinsUK	53, 95	skin, blood
	MammConsortium	2069	skin

(1×6663 size vector), whereas for the C only method, x includes only saliva methylation values (1×1307 size vector).

In our study, we observe additional data from a collection of auxiliary studies, denoted as K, to enhance our estimation of the vector β. Each auxiliary study provides samples that may vary in their relevance to the target data. The incorporation of these auxiliary samples into our analysis can take various forms—they can be merged into a single dataset, kept separate, or grouped into distinct combined datasets. To accommodate these different integration strategies, we define L as the ensemble of auxiliary data configurations utilized. Specifically, L represents any subset or combination of the K studies that we use for informing β.

Mathematically, let us consider K auxiliary samples indexed by $k = 1, \ldots, K$. The set L then represents a collection of these samples in various configurations, such as $L = \{(4), (4, 1), (4, 1, 3), (4, 1, 3, 2)\}$, or as a single combined set $L = \{(1, 2, 3, 4)\}$, or as individual datasets $L = \{(1), (2), (3), (4)\}$ when $K = 4$. Here, the first instance illustrates L containing three groupings of l combining multiple auxiliary datasets (as is performed in the default settings of TransLasso), the second instance shows L as one grouping that combines all auxiliary datasets, and the third instance depicts L with each l representing an individual auxiliary dataset. For scenarios where auxiliary samples are considered individually, such as in the Oracle 1df method, we can substitute l with k in our notation. This substitution reflects the scenario where each auxiliary sample is assessed separately to inform the model.

The general methodology can be summarized into five primary steps below.

Step 1: Perform Lasso on the Target Data (saliva DNAm → blood DNAm biomarker):

$$\hat{\beta}_0 = \min_{\beta_0 \in \mathbb{R}^p} \|\mathbf{y}^{(0)} - \mathbf{X}^{(0)}\beta_0\|^2 + \lambda_0\|\beta_0\|_1 \tag{4.1}$$

Step 2A*: Determine the informative set, L, from auxiliary data, K, as detailed in Section 4.2.4

Step 2B: Perform Lasso on Informative Auxiliary Data for each $l \in L$ (any tissue DNAm → any other tissue DNAm biomarker):

$$\hat{w}_l = \min_{w_l \in \mathbb{R}^p} \|\mathbf{y}^{(l)} - \mathbf{X}^{(l)}w_l\|^2 + \lambda_l\|w_l\|_1 \tag{4.2}$$

Step 3: Perform Lasso on Target Data Residuals (saliva DNAm → Residual blood DNAm biomarker):

$$\hat{\delta}_l = \min_{\delta_l \in \mathbb{R}^p} \|(\mathbf{y}^{(0)} - \hat{\mathbf{y}}^{(0)}_{\hat{w}_l}) - \mathbf{X}^{(0)}\delta_l\|^2 + \lambda_{0l}\|\delta_l\|_1 \tag{4.3}$$

$$= \min_{\delta_l \in \mathbb{R}^p} \|(\mathbf{y}^{(0)} - \mathbf{X}^{(0)}\hat{w}_l) - \mathbf{X}^{(0)}\delta_l\|^2 + \lambda_{0l}\|\delta_l\|_1 \tag{4.4}$$

Step 4: Calculate coefficients estimated from the auxiliary samples with a constant threshold for combining:

$$\hat{\beta}_l = I_{cw}\hat{w}_l + I_{cd}\hat{\delta}_l \tag{4.5}$$

Step 5: Combine $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_L$ for $\forall m \in 0 : L$ using the squared prediction error on the target data.

$$\hat{\beta} = \theta_0\hat{\beta}_0 + \cdots + \theta_L\hat{\beta}_L \tag{4.6}$$

where

$$\theta_m = \frac{\frac{\text{Total squared error}}{m\text{th squared error}}}{\sum_{m=0}^{L} \frac{\text{Total squared error}}{m\text{th squared error}}} = \frac{1/\sum_i^{n_0}(y_i^{(0)} - x_i^{(0)\top}\hat{\beta}_m)^2}{\sum_{m=0}^{L}[1/\sum_i^{n_0}(y_i^{(0)} - x_i^{(0)\top}\hat{\beta}_m)^2]} \tag{4.7}$$

In Steps 1-3, the tuning parameters $\lambda_0, \lambda_l, \lambda_{0l}$ are selected through cross validation and is described in greater detail in the next Methods section. In Step 4, I_{cw} and I_{cd} are indicator

vectors that indicate which coefficients of $\hat{w}_l$ and $\hat{\delta}_l$ are of magnitude greater than or equal to the threshold value of c. This is also in the next Methods section, where we detail the different constants we use, including allowing any coefficient magnitudes to be added together. In the weighting scheme θ_l of step 5, the error is calculated in the training target data. The term, $x_i^{(0)\top}\hat{\beta}_m$ is the the prediction for individual i in the target data using model m, and the summation over i aggregates squared prediction errors over all individuals. The weights are based on the inverse of the squared error for each model's predictions on the target data, which gives more importance to models with lower errors. We note that this is *not* the error from the left out target data, because then this algorithm would not be applicable to researchers developing new algorithms without validation data. Error from LODO is used at a later step, however, to combine the coefficients from our various TL models in a Super Learner process to obtain the final coefficients.

This methodology is described with all 6 target datasets together in $\mathbf{y}^{(0)}$ and $\mathbf{X}^{(0)}$, which is the case for our final algorithms. However, in the process of testing parameterization and TL method specification, we employ a Leave-One-Data-Out (LODO) process, meaning 1 target dataset is held out and the remaining 5 are used to develop the TL coefficients in the steps described above. As such, for developing the TL algorithm for each biomarker and each tested TL method, the 5 steps were performed 6 times to capture $j = 6$ LODO folds. This results in $\hat{\beta}_{-0_1}, \ldots, \hat{\beta}_{-0_6}$ for each jth fold (total of 6 target datasets), which are the target data coefficients when the jth target data is left out ($\hat{\beta}_{-0_j}$). After each fold is run, the $\hat{\beta}_{-0_j}$ coefficients are applied in the jth dataset to calculate the LODO error and correlation for comparing the TL methods. The LODO error and correlation are a sum of the metrics weighted by the left out dataset size, n_j. For example, the LODO total squared error is calculated as:

$$\text{LODO Squared Error} = \sum_{j=1}^{6} \frac{n_j}{n_0} \sum_{i=1}^{n_j} \left(y_{ji}^{(0)} - \hat{y}_{ji}^{(0)} \right)^2 \tag{4.8}$$

$$= \sum_{j=1}^{6} \frac{n_j}{n_0} \sum_{i=1}^{n_j} \left(y_{ji}^{(0)} - x_{ji}^{(0)\top} \hat{\beta}_{-0_j} \right)^2 \tag{4.9}$$

where $y_{ji}^{(0)}$ is the blood DNAm biomarker of interest for the ith person in the jth left out target dataset and $x_{ji}^{(0)\top}$ is vector of saliva DNAm for the ith person in the jth left out target dataset.

4.2.4 Calculating Informative Auxiliary Sets (Step 2A):

In Step 2A, we use * to denote the possibility to skip this step, as is the case for the Oracle methods, where informativeness is a-priori known and therefore does not need calculated. We employ two oracle methods. One, which we refer to as Oracle A0, treats all auxiliary datasets as equally informative, and L is the combined set of all K auxiliary data. In the second oracle method, deemed Oracle 1df, each auxiliary dataset is considered informative, but not necessarily equal. In this case, L is exactly K, with each auxiliary dataset being treated individually.

When we are determining the informativeness of auxiliary datasets, we construct L, the set of informative auxiliary datasets. In summary, this algorithm computes differences in marginal correlations between auxiliary datasets and the target datasets. The most significant of these differences calculates an index $\widehat{R}^{(k)}$ that gives an indication of the informativeness of the kth auxiliary dataset. These are ranked and auxiliary datasets are taken in sequence to produce the informative set L.

Step 1: Calculate $\hat{\Delta}^{(k)}$:

$$\widehat{\Delta}^{(k)} = \frac{1}{n_k}\sum_{i=1}^{n_k} x_i^{(k)} y_i^{(k)} - \frac{1}{n_0}\sum_{i=1}^{n_0} x_i^{(0)} y_i^{(0)}. \tag{4.10}$$

$\hat{\Delta}^{(k)}$ holds the marginal statistics for the k^{th} auxiliary dataset, which is the average difference between the k^{th} auxiliary dataset CpG's and the target dataset CpG's correlation to the DNAm biomarker outcome. The term $X'y$ captures how each predictor variable individually correlates with the DNAm biomarker variable. If you think of y as a vector in a space, and each column of X as another vector in that space, then each component of $X'y$ is essentially the projection of y onto each of those predictor vectors. So we are taking the average difference of the projection between the auxiliary data and target data for each p. ($\hat{\Delta}^{(k)}$ is a $p \times 1$ vector)

Step 2: Pick out the most significant components of $\widehat{\Delta}^{(k)}$ by obtaining the top t^* $\hat{\Delta}^{(k)}$ (largest differences) between the auxiliary and target data, called $\widehat{T}_k$. As such, $\widehat{T}_k$ are indices of the largest marginal statistics for each auxiliary dataset. The default value for t^* is one third the target dataset size ($n_0/3$), and we also explore variations to include more $\hat{\Delta}^{(k)}$. This parameter changes how many of the largest differences are considered. Call this subset $\hat{\Delta}^{(k)}_{\widehat{T}_k}$.

$$\widehat{T}_k = \left\{ 1 \le j \le p : \left|\widehat{\Delta}_j^{(k)}\right| \text{ is among the first } t_* \text{ largest of all } \right\} \tag{4.11}$$

By taking the largest values, we are capturing the most unique differences between the auxiliary and target data. Capturing the largest differences allows us to understand how different the auxiliary data are from the target data.

Step 3: Calculate Sparse Index ($\widehat{R}^{(k)}$) for the k^{th} auxiliary dataset.

$$\widehat{R}^{(k)} = \left\| \widehat{\Delta}^{(k)}_{\widehat{T}_k} \right\|_2^2 \tag{4.12}$$

This estimated sparse index is the squared Euclidean norm of the vector $\hat{\Delta}^{(k)}_{\widehat{T}_k}$. Essentially, this is summing the total squared differences of the top t^* correlation differences. Auxiliary datasets with smaller $\widehat{R}^{(k)}$ are more informative because the total deviation between the auxiliary and target data is small. This metric will be larger for auxiliary datasets that have more significant differences from the target datasets. After this, we use these sparsity indices to make candidate sets / subsets of the auxiliary data. We rank each $\hat{R}^{(k)}$ and take the smallest values in sequence to make L subsets of size $1, \ldots, K$. Specifically, the l^{th} candidate set is:

$$l = 1 \leq k \leq K \colon \widehat{R}^{(k)} \text{is among the first } l \text{ smallest of all} \tag{4.13}$$

For illustration, we have 5 auxiliary datasets, and suppose the rank of $\widehat{R}^{(k)}$'s is 4,1,2,3,5. Then L would be the 5 element set $\{(4), (4,1), (4,1,2), (4,1,2,3), (4,1,2,3,5)\}$. Computation proceeds as described above with each subset of auxiliary datasets being run together in the Lasso algorithm. In total, there will be K subsets each with $1, \ldots, K$ auxiliary datasets included. Therefore, the most informative auxiliary dataset will be present in all L subsets.

4.2.5 Optimizing Transfer Learning Methodology for Cross-Tissue DNAm Prediction

The core aim of our study was to assess the utility of Transfer Learning (TL) in the context of DNA methylation (DNAm) and DNAm-based biomarkers. We made methodological modifications within the TransLasso framework to tailor the TL process for DNAm based biomarkers. The variations explored include lambda penalty optimization, coefficient thresholding, and the integration of auxiliary dataset information, which are delineated below.

Lambda Penalty

The λ penalty in penalized regression models is typically determined through cross-validation (CV), either by identifying the λ that minimizes CV error or by selecting a more conservative λ within one standard error above the minimum. In transfer learning contexts involving multiple auxiliary datasets, optimizing λ for each dataset can be computationally intensive. We expand upon the TransLasso algorithm, which uses a constant calculated from the target dataset, c_0, to define λ_k for each auxiliary set: $\lambda_k = c_0 \times \sqrt{2(log(p))/n_k}$. Our investigation included this approach and allowed for λ selection through CV in each auxiliary set, aspiring to harness dataset-specific nuances to inform λ choice.

In our CV λ selection process, λ_0 is the optimal λ for the target dataset from CV on $\mathbf{y}^{(0)}$ and $\mathbf{X}^{(0)}$, λ_k is the optimal tuning parameter for $\hat{w}_k$ from $\mathbf{y}^{(k)}$ and $\mathbf{X}^{(k)}$, and λ_{0k} is the optimal tuning parameter λ_k scaled by the target dataset size, n_0. We scrutinized the constant λ and individualized λ approaches under the min and 1se parameterizations to determine the most suitable for our DNAm biomarker prediction. This approach was geared towards determining the optimal λ for our prediction problem, with the intent to refine the model's predictive accuracy and reliability.

Coefficient Thresholding

In the transfer learning process, a two-step coefficient computation is used for estimating coefficients from auxiliary data: initially within the auxiliary dataset to predict the outcome of interest (weights, w_k), followed by an adjustment for biases between the target and auxiliary datasets (δ_k). Lasso regression shrinks coefficients toward zero, which can result in some coefficients being small and near zero. To mitigate the incorporation of negligible coefficients from auxiliary sources, thresholding can be applied prior to combining the auxiliary weights and bias coefficients. While the conventional TransLasso method maintains coefficients exceeding a threshold of λ (described above), our exploration incorporated more lenient thresholds, including halving the threshold ($0.5 \times \lambda$) and considering all coefficients, regardless of coefficient magnitude. We compare these three thresholding approaches and refer to them as 'lambda', 'half lambda', and 'all'. This approach acknowledges the inherent characteristics of DNA methylation (DNAm) where the effects can be minuscule. By lowering or removing the thresholding, we may account for the subtlety of DNAm effects and improve the application of TL to DNAm

data.

Auxiliary Dataset Information

The relevance of auxiliary datasets to your target data and objective can often be ambiguous. An 'oracle' scenario would entail using only known informative auxiliary datasets; however, this is not always practical. Consequently, we need a mechanism to gauge the informativeness of auxiliary datasets. We examined both the oracle and estimation approaches for auxiliary dataset incorporation. In the oracle scenario, we considered all auxiliary data as either a single collective sample or as separate individual datasets, referred to as Oracle A0' and Oracle 1df', respectively. Informativeness of auxiliary datasets were computed as described above using the differences in marginal correlations. We varied the number of considered correlations to calculate auxiliary data similarity starting from the default value of one-third of the target dataset size ($n_0/3$). For the C+S method, we considered the top 100, 500, 2000, and 6663 (all) correlation differences. For the C method, limited to the 40K array-conserved sites, we considered the top 100, 500, and 1307 (all) correlation differences. These methods are referred to as Rhat and the corresponding number, like 'Rhat100'.

4.2.6 Evaluating TL Methods and Developing Final Algorithms

Evaluating Transfer Learning (TL) Methodologies

To assess the efficacy of various TL algorithms, we conducted a comprehensive evaluation of their performance using Leave-One-Dataset-Out (LODO) correlation, mean squared error (MSE), and mean absolute percent error (MAPE). This approach provides a holistic view of how different parameters and methods of integrating auxiliary data affect the predictive accuracy for DNA methylation (DNAm) biomarkers. It should be noted, however, that while this evaluation offers a broad understanding of algorithmic performance, it does not specifically address variations across individual DNAm biomarkers.

Development of Optimized TL Algorithms

Our methodology for refining TL algorithms involved a detailed analysis of their performance across each DNAm biomarker. We calculated and then ranked both the correlation and prediction error for each TL method within individual DNAm biomarkers. This ranking was based on a weighted LODO MSE and the correlation between the predicted and actual blood DNAm

biomarkers, with weights assigned in proportion to the size of the dataset left out. The most effective methods were identified based on their mean performance in terms of correlation and MSE across all DNAm biomarkers. The four top-ranked methods were subsequently applied to the entire target dataset, with the resulting coefficients being recorded.

In the final stage of our analysis, we employed a Super Learner approach to combine coefficients from these four top performing methods, thereby deriving the final algorithms' coefficients for each DNAm biomarker. The weights assigned to each coefficient set were inversely proportional to the squared errors from the LODO analysis of the respective method. We opted for the Super Learner framework over the single best-performing TL method because of the former's proven superiority in enhancing predictive accuracy and providing more stable estimates in regression models, particularly when multiple a priori techniques are utilized [80]. We developed the optimal TL algorithm separately for the two algorithms desired: one incorporating saliva DNAm biomarkers with 6662 CpGs as potential covariates and the other using only the 1307 CpGs conserved on the 40K array. Consequently, for each DNAm biomarker, we developed two cross-tissue prediction algorithms — one incorporating saliva DNAm biomarkers and the other based solely on saliva methylation beta values. However, as described in the section below, we do not provide all DNAm biomarker algorithms to researchers because the TL method does not always adequately predict DNAm biomarkers.

4.2.7 Validation of Algorithms

In our study, we embarked on a comprehensive evaluation of Transfer Learning (TL) to determine its potential contributions in the realm of cross-tissue prediction, particularly in the context of predicting blood DNA methylation (DNAm) biomarkers. Our evaluation framework was multi-faceted, focusing on three key benchmarks deemed critical for any robust cross-tissue prediction algorithm.

Comparison of TL with Direct Saliva DNAm Estimates

Initially, we assessed whether TL offers an advantage in predicting blood DNAm biomarkers from saliva DNAm. This involved contrasting the efficacy of our TL algorithms with the baseline approach of directly computing biomarkers from saliva DNAm. The TL algorithms can be considered advantageous if they demonstrate superior performance in approximating blood DNAm biomarkers compared to using saliva DNAm as a direct surrogate.

Benchmarking TL against Lasso Regression

We then compared the TL algorithms against conventional Lasso regression to evaluate the benefits of adopting advanced computational techniques. Lasso regression was applied solely to the target data, serving as a comparative baseline. In contrast, our TL methods leveraged both the target data and additional auxiliary cross-tissue DNAm data. The key distinction between the TL and Lasso methods lies in the incorporation of this auxiliary cross-tissue DNAm data in TL as both TL and Lasso used the same target and held out datasets for development and comparison. We analyzed and compared the Leave-One-Dataset-Out (LODO) correlations and errors generated by both the TL and Lasso algorithms. The TL algorithms can be considered advantageous for DNAm biomarkers if they demonstrate better prediction accuracy compared to Lasso techniques.

Comparing TL Algorithms with Different Domains

Lastly, we scrutinized the differential predictive power and accuracy between our two distinct TL algorithms developed for each DNAm biomarker. One algorithm incorporated saliva DNAm biomarkers and beta values across 6662 CpG loci, while the other was confined to 1307 CpG sites alone. This comparative analysis aimed to determine whether the inclusion of saliva DNAm biomarkers as covariates significantly enhances predictive accuracy or if their inclusion is largely redundant in the context of these TL models.

4.2.8 Scope of Algorithms: Application to Validation Sets

The utility of these algorithms extends beyond mere accuracy; for them to resonate with and be adopted by the broader research community, they must demonstrate an ability to reflect biologically meaningful relationships. We explore this by examining if our predicted DNAm biomarkers have the same relationships that have been established between blood DNAm biomarkers and phenotypic variables across three distinct validation datasets, and further explore the relationship to age in two additional validation datasets.

We calculate the association between our predicted blood DNAmTL biomarker and sex to evaluate if our predictions align with known biological trends, where females tend to have longer telomere length compared to men. Next, we compare predicted blood DNAm biomarkers among people with COPD, asthma, and healthy controls, using sputum DNAm and adjusting for

age. Research has demonstrated older DNAm age estimates in people with COPD and asthma compared to controls. We also investigate the congruence of predicted blood cell count DNAm biomarkers with lymphocyte percentages. The study also reports cumulative pack years, and we evaluate the association to predicted DNAm biomarkers surrounding lung health and fitness: DNAmPackYears, DNAmFEV1, and DNAmVO2max blood biomarkers. Finally, we explore if our predicted blood DNAm biomarkers change in the expected direction from a longitudinal exercise, diet, and sleep intervention study. Here, we employ a main effects mixed model, controlling for age and study duration, to determine if our fitness-related blood biomarkers show expected improvements in the intervention group. This analysis also offers the unique opportunity to assess whether the newly developed blood DNAm fitness biomarkers exhibit expected improvements from an exercise intervention, an evaluation that has not yet been undertaken. These comparisons validate the predictive potential of the TL algorithms and demonstrate their robustness to application in novel data sources.

Assessing Algorithmic Flexibility to Tissue Type

Because our TL approach incorporates information from multiple human tissues, we are interested in understanding if the algorithm is robust to tissue type. For example, instead of saliva DNAm, can we provide skin, lymph nodes, or other tissue DNAm and still have accurate blood DNAm biomarker predictions? We applied the C algorithms to three accessible tissues (lymph nodes, adipose, and muscle DNAm) in HIV-negative and HIV-positive individuals and compared the predicted DNAm biomarkers. This helps us assess if the predictions accurately reflect the accelerated aging and immune dysregulation associated with HIV infection. Furthermore, we examine the prediction accuracy using skin DNAm compared to true blood DNAm biomarkers in TwinsUK sample, and we relate the predicted blood DNAm biomarkers to chronological age to determine if the predictions correlate in the expected direction.

4.2.9 Applying Human Biomarkers Across Species

In our final novel exploration, we apply the algorithms designed for predicting human blood DNAm biomarkers to other mammals. Specifically, we use skin DNAm measured in mammals using the 40K Mammalian array and calculate predicted DNAm biomarkers using our C algorithms. This cross-species application aims to estimate 'human equivalent' aging biomarkers, offering a unique opportunity to translate and understand DNAm aging biomarkers beyond

chronological age in different species. This includes an assessment of whether the algorithms correlate in the expected directions to relative age (chronological age / maximum species age) and if the predicted values fall within reasonable ranges. This offers access for animal studies to have meaningful DNAm biomarkers that translate to human phenotypes and direct methods for animal studies to inform human studies. These endeavors strive to bridge cross-species gaps and contribute to a broader understanding of aging and disease processes.

Initially, we focus on mammalian species most commonly used in biomedical research, including primates, mice, and rats. The selection of these species is strategic, as their frequent use in laboratory studies presents an opportunity to leverage our biomarkers as surrogate aging indicators. We also studied bats as a counter example of distant mammalian species to humans that are also studied in aging research [81]. Subsequently, we broaden our scope to encompass a diverse array of mammalian species, analyzing skin DNA methylation (DNAm) samples across 91 species. To evaluate these algorithms, we calculate overall correlation across all species and samples, within-species weighted correlation, and compare the differences between the two to understand if the strength of relationship is driven by conserved signal among all species or conserved signal within species.

Rejuvenation Effects in Mice

Previous research has demonstrated epigenetic rejuvenation in kidney and skin samples of mice with long-term partial reprogramming [82]. While this research sampled multiple tissues, it did not include blood measurements. We wanted to determine if our C algorithms are predictive of rejuvenation in the blood by using mouse skin DNAm as input. This analysis not only provides insight to mice blood, an unmeasured tissue, but it also offers novel insight to potential blood rejuvenation of human DNAm biomarkers from partial reprogramming. In this study, they used both Black6 (B6) mice and 4 Factor (4F) mice, with the latter being genetically modified mice with 4 reprogramming factors from a lentiviral vector (Tet-O-FUW-OSKM). This vector is activated and the factors are expressed when the inducer, doxycycline (Dox) is administered. Mice were categorized into 1 of 3 groups: Control (n=7 B6, n=11 4F, n= 2 B6 Dox), Short Treatment (n=3 4F+1mDox), or Long Treatment (n=5 4F+7mDox, n=2 4F+10mDox). In total, there are 20 control mice, 3 short term mice, and 7 long term mice. To evaluate whether the C algorithms capture DNAm biomarker rejuvenation, we apply our C algorithms to mice skin DNAm, and then calculate the association of each DNAm biomarker to

the treatment group adjusting for relative age. We include relative age as a covariate because balanced ages were not observed in the treatment groups. For the DNAm fitness biomarkers, we additionally include sex as a covariate because the fitness biomarkers are sex specific. Therefore for DNAmGaitspeed, DNAmGripmax, DNAmVO2max, DNAmFEV1, and DNAmFitAge, we present results controlling for relative age and sex.

Summary of Terminology Used

- **LODO**: Leave-One-Data-Out, meaning leaving 1 target dataset out at a time.
- **TL**: Transfer Learning
- **C+S Method**: One algorithmic set up which includes 6662 saliva CpG loci and the saliva DNAm biomarker as covariates to predict the blood DNAm biomarker.
- **C Method**: The second algorithmic set up which includes 1307 CpG loci as covariates to predict the blood DNAm biomarker.
- **Target Data**: Datasets that include saliva and blood DNAm from the same individual. There are 6 of these and are used to develop the algorithms. Referred to with $^{(0)}$.
- **Auxiliary Data**: Datasets that include two tissues that are not saliva and blood from the same individual. There are $K = 5$ of these and are used to develop the algorithms. Referred to with $^{(k)}$.
- **Validation Data**: Four datasets that are not used to develop the algorithms, but are used to apply the algorithms to validate the signatures.
- **Oracle A0**: TL method where all auxiliary datasets are a-priori specified as equally informative and combined into 1 auxiliary dataset
- **Oracle 1df**: TL method where all auxiliary datasets are specified as informative and used as individual auxiliary datasets
- **Informative Auxiliary Sets**: When not using the Oracle TL methods, we estimate which auxiliary datasets are informative for our target data. Referred to with l and L.
- **Min / 1SE**: The λ penalty term in Lasso that either minimizes the CV error or is the value that is at most 1 standard error above the minimum CV error.
- **lambda, half lambda, all**: The three different coefficient thresholds used for combining weights, $\hat{w}_k$, and bias coefficients, $\hat{\delta}_k$, from the auxiliary data
- **Rhat**: The different number of correlation differences considered when determining auxiliary informative sets.

4.3 Results

4.3.1 Optimal TL Algorithm

Optimal Parameters for C+S Method

The best median LOOCV correlation was observed using the minimum lambda in the C+S methods (0.515) (Table 4.2). Interestingly, despite a relatively lower correlation of 0.475 under the constant lambda, determined based on CV in target data and auxiliary dataset size, the error metrics are slightly reduced. This suggests improved prediction accuracy despite a less strong linear relationship between the predicted and true blood DNAm biomarker. Settings without thresholding or partial thresholding results in comparable correlation with no thresholding having 1.3% lower error. Both of these methods outperform complete (lambda) thresholding with a difference of 0.05 correlation and 2.3% higher error, suggesting complete thresholding underfits the model by removing small, but informative coefficients.

The integration of auxiliary datasets through the Oracle 1df approach yielded the best correlation and the lowest Mean Squared Error (MSE) on average, with a median Mean Absolute Percentage Error (MAPE) similar to other methods incorporating auxiliary information. Contrary to expectations, the Oracle 1df method outperformed the standard Oracle approach, which treats all auxiliary datasets as a single combined source, improving correlation by 0.06 and maintaining comparable MAPE. Furthermore, the estimation of the informative set indicated a beneficial trend in incorporating more Rhats, as evidenced by a gradual improvement in correlation (R = 0.494, 0.520, 0.530) with negligible variation in error metrics. To understand variability in performance for the C+S method based on parameterization, we provide Supplemental Figure B.5 for each biomarker.

In summary, the C+S method's optimal generalization employs the Oracle 1df approach, does not perform coefficient thresholding, and uses the minimal lambda derived from cross-validation in each dataset.

Optimal Parameters for C Method

In contrast, the C method consistently demonstrates higher median R values across most parameters compared to C+S. This suggests a stronger correlation with the actual blood DNAm biomarker under the C setting. Echoing the C+S method's findings, the minimum lambda from

auxiliary data CV was preferred, achieving the highest average R of 0.604 and lowest MAPE of 24.8% (Table 4.2).

In a departure from C+S, the C method favored more rigorous coefficient thresholding. Specifically, the median R of 0.597 with complete thresholding was 0.016 higher than settings without thresholding. This divergence between methods suggests a greater number of CpGs are needed to estimate the blood biomarker in the absence of original tissue DNAm biomarkers (C method), however, this also propagates noise in the prediction, which can be corrected by removing smaller coefficients. Finally, auxiliary methods using either Oracle 1df or estimating the informative set with more Rhats perform comparably well based on R and MAPE. While Oracle and Oracle 1df have similar correlations, Oracle 1df decreases error by approximately 2%.

Thus, the C method's best practice includes the minimal CV lambda from each auxiliary dataset combined with complete (lambda) coefficient thresholding. Using either Oracle 1df or estimating the informative sets with more Rhats are similarly efficacious.

General Recommendations

Both C+S and C methods favor the Oracle 1df method and the minimal lambda. They diverge on coefficient thresholding — more thresholding benefits the C method, while less thresholding benefits the C+S method. These recommendations, however, do not account for individual biomarker variability, which may influence the optimal choice for a specific biomarker. Researchers with the resources to conduct an LOO procedure are advised to employ the TL tuning function to determine the best parameters for their specific scenario. In the absence of such a procedure, our broad recommendations provide a starting point.

4.3.2 Final Algorithms

Our previous section evaluated individual parameters, however, it's crucial to recognize that the best individual parameterizations might not synergize when combined. Therefore, we ranked each TL method within each biomarker to determine the most efficacious models. These results generally aligned with our recommendations, particularly for the C+S method where the Oracle 1 df, min lambda, and all coefficient setup emerged as the top-ranked for lowest Mean Squared Error (MSE) and highest correlation among all C+S TL methods. Despite its superior performance, it didn't achieve a universal top rank, highlighting the presence of multiple

Table 4.2: Optimal Parameters for Transfer Learning with DNAm

Parameter	C+S			C		
	median R	median MSE	median MAPE	median R	median MSE	median MAPE
Lambda						
Min	0.515	240	25.2	0.604	209	24.8
1SE	0.503	237	26.2	0.572	228	25.5
Constant	0.475	219	24.8	0.535	237	27.0
Coefficient Threshold						
Complete (lambda)	0.490	256	27.1	0.597	226	25.5
Partial (.5 lambda)	0.545	248	26.1	0.592	248	25.6
None	0.540	220	24.8	0.581	200	25.1
Auxiliary Informative						
Oracle	0.493	248	26.2	0.596	248	26.7
Oracle 1df	0.554	214	25.9	0.592	243	24.8
Estimate A0	0.513	244	26.1	0.586	214	25.5
Rhat nk/3	0.494	236	25.4	0.573	226	25.9
Rhat 500	0.520	248	26.2	0.592	208	25.4
Rhat All	0.530	242	26.2	0.592	213	25.4

algorithms with nearly identical performance across various methods. The top 4 C+S methods were Oracle A0 1df, min lambda, all coef; Oracle A0, 1se lambda, half coef; Estimate A0 Rhat nk/3, 1se, half coef; and Estimate A0 Rhat All, 1se lambda, half coef. For the C method, the top 4 algorithms were OracleA0 1df, 1se lambda, all coef; Estimate A0 Rhat 500, min lambda, all coef; Estimate A0 Rhat nk/3, min lambda, lamb coef; and Oracle A0, min lambda, lambda coef. The variation in top methods reaffirms that no single TL method suits all scenarios. Given the absence of one-size-fits-all solution, we used a Super Learner approach to generate the final coefficients for each biomarker. This strategy capitalizes on the strengths of each top algorithm, mitigating their individual weaknesses and catering to the specificities of each biomarker. The final algorithms for each biomarker are readily accessible on our GitHub repository at https://github.com/kristenmcgreevy/EpigenTL, and are also provided in the Appendix section A.8.

4.3.3 Algorithmic Comparison

Comparison to Saliva Surrogates

In the comparative analysis of Transfer Learning (TL) algorithms for estimating blood DNA methylation (DNAm) biomarkers from saliva DNAm, we found that 20 out of 26 biomarkers were more accurately predicted by TL methods than by their saliva DNAm surrogates, based

on Mean Squared Error (MSE) and correlation metrics. When separating the comparison between the C+S method and C method to saliva, 18 and 15 biomarkers were more accurately predicted with the TL methods, respectively. The amount of improvement using either of the Transfer Learning methods varied by biomarker. Some biomarkers saw drastic improvement, such as DNAmGrip_noAge and DNAmLeptin, which had 0.364 and 0.364 stronger correlation, respectively. DNAmLeptin also saw a 7 figure improvement in MSE when using our C+S algorithm. For our provided algorithms, the average LODO correlation improvement of our algorithms compared to the saliva surrogate itself is 0.120. This improvement is not universal across all DNAm biomarkers, however. One biomarker—Gran—showed notably poor correlations across TL, Lasso, and saliva surrogate methods, prompting its exclusion from any saliva to blood biomarker recommendations. In addition, CD8pCD28nCD45RAn did not have good performance with C+S method, C method, or Lasso, however the saliva surrogate had a decent correlation (0.33) (Supplemental Table B.9).

Comparison between TL and Lasso

When comparing TL algorithms directly against Lasso regression, TL algorithms outperformed Lasso for 23 out of 26 biomarkers. This underscores the significant potential of TL algorithms in the generation of new DNAm biomarkers. Additionally, the Lasso algorithm averaged the lowest/worst rank in MSE and Correlation across all TL and Saliva surrogate methods within each biomarker. Even when Lasso outperformed the TL methods, it did not outperform the saliva surrogate alone. With these results, we did not observe any instance where Lasso would be more beneficial than TL when saliva surrogates are available. As such, we strongly encourage researchers to adopt TL methods in place of Lasso when similar data are available, like when different tissue DNAm samples are available. Additional metrics evaluating median absolute percent error are presented in the Appendix Section B.4.

Comparison of C+S to C method

Unexpectedly, 9 biomarkers estimated with the C method surpassed the performance of the C+S method. This suggests that incorporating DNAm biomarkers from the original tissue can sometimes introduce noise to its prediction, with pure methylation loci offering clearer signals for certain biomarkers. DNAmCystatinC was one biomarker where the C method out performed the C+S method with a 0.259 improvement in correlation and 9 figure improvement in MSE

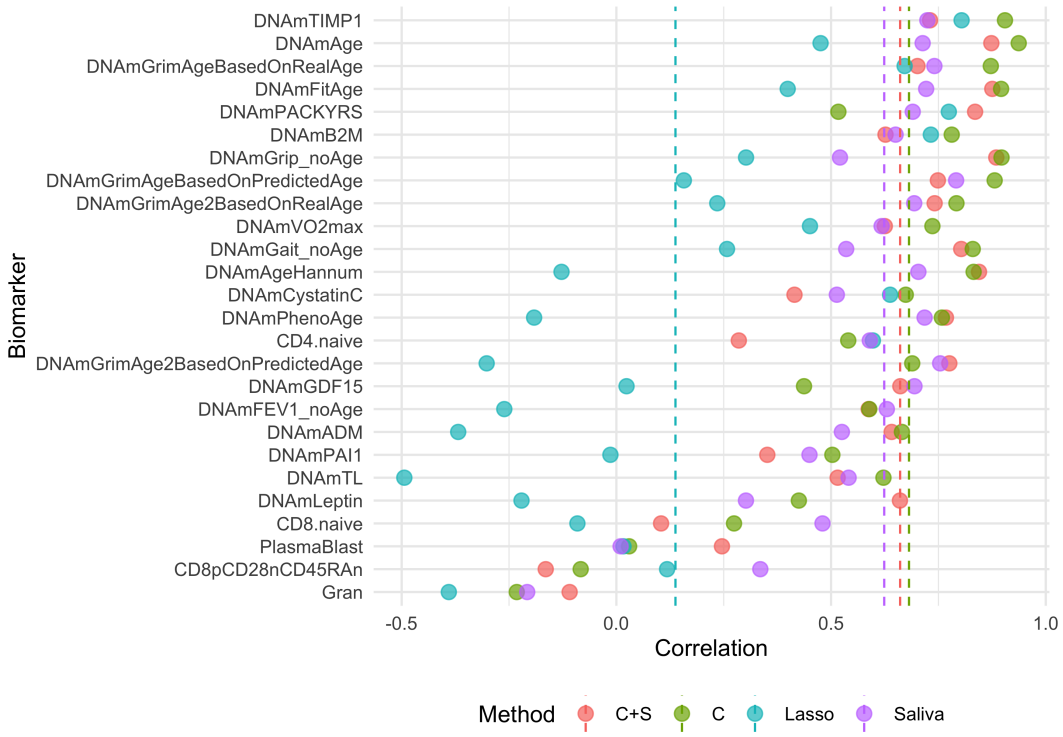


Figure 4.1: Correlation between True and Estimated Blood DNAm Biomarkers by top Performing TL Methods, Lasso, and Saliva Surrogates. Median correlation presented as dotted line. LODO Correlation presented for C+S, C, and Lasso methods.

compared to C+S method and saliva surrogate.

While not all TL algorithms outshined their corresponding saliva surrogates, they did exhibit substantial predictive power for 23 out of the 26 biomarkers. For instances where saliva DNAm biomarkers cannot be calculated without extensive imputation—as with the 40K array— our C method algorithms provide a valuable alternative. This resulted in 10 biomarkers having both a C+S algorithm and C algorithm. Three biomarkers have a C algorithm excluded due to inadequate predictive power: CD8pCD28nCD45RAn, Gran, and PlasmaBlast. In addition, we want to iterate that for CD4.naive, CD8.naive, DNAmB2M, and DNAmPAI1 biomarkers, the C algorithms are only recommended when saliva DNAm biomarkers are unavailable. A summary table detailing the available algorithms by biomarker can be found in Table 4.3.

In summary, 11 biomarkers were best predicted using the C+S method, 9 with the C method, 5 using saliva DNAm biomarkers directly, and 1 biomarker was not well estimated by any method. We provide algorithms for predicting 23 blood DNAm biomarkers from saliva DNAm along with guidelines for their use. For researchers in the field, this study offers a comprehensive suite of algorithms tailored to a variety of biomarkers, enhancing the predictability and applicability of DNAm studies. Our research underscores the efficacy of TL in biomarker pre-

Table 4.3: Summary of Cross Tissue DNAm Biomarker Algorithms Provided

Biomarker	Best Model	C Algorithm	Total Algorithms available
CD4.naive	Saliva	Yes	1
CD8.naive	Saliva	Yes	1
CD8pCD28nCD45RAn	Saliva	No	0
DNAmADM	C+S	Yes	2
DNAmAge	C	Yes	1
DNAmAgeHannum	C+S	Yes	2
DNAmB2M	Saliva	Yes	1
DNAmCystatinC	C	Yes	1
DNAmFEV1_noAge	C+S	Yes	2
DNAmFitAge	C+S	Yes	2
DNAmGait_noAge	C	Yes	1
DNAmGDF15	C+S	Yes	2
DNAmGrimAge2BasedOnPredictedAge	C+S	Yes	2
DNAmGrimAge2BasedOnRealAge	C+S	Yes	2
DNAmGrimAgeBasedOnPredictedAge	C	Yes	1
DNAmGrimAgeBasedOnRealAge	C	Yes	1
DNAmGrip_noAge	C	Yes	1
DNAmLeptin	C+S	Yes	2
DNAmPACKYRS	C+S	Yes	2
DNAmPAI1	Saliva	Yes	1
DNAmPhenoAge	C+S	Yes	2
DNAmTIMP1	C	Yes	1
DNAmTL	C	Yes	1
DNAmVO2max	C	Yes	1
Gran	None	No	0
PlasmaBlast	C+S	No	1

diction and development and encourages its adoption over Lasso regression when applicable, particularly when multi-tissue DNAm data are available.

4.3.4 Application to Validation Datasets

DNAmTL to Sex Relationship

We assessed the association between predicted blood DNA methylation Telomere Length (DNAmTL) biomarkers and sex, noting that females generally exhibit longer telomeres compared to males. Utilizing saliva DNA methylation (DNAm) as input, both C+S and C method blood DNAmTL predictions accurately reflected this trend. Specifically, females exhibited an average increase in telomere length of 0.29 and 0.3 for C+S and C methods, respectively, aligning closely with the observed 0.33 mean difference in the saliva surrogate. These disparities were statistically significant, as indicated by t-test and Kruskal Wallis test, with p-values ranging from 8.3E-6 to 3.5E-4 (Supplemental Table B.10).

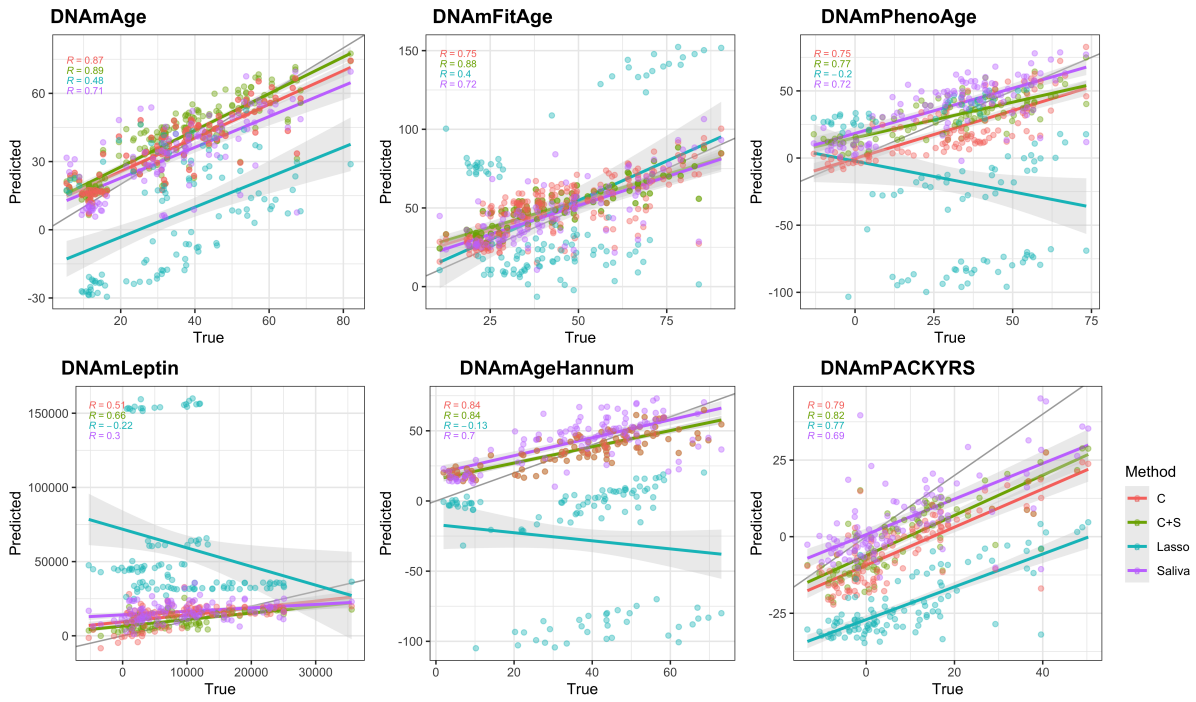


Figure 4.2: Scatterplots between True and Estimated Blood DNAm Biomarkers in Target Datasets. LODO Correlation presented for C+S, C, and Lasso methods.

COPD, Asthma, and Healthy Controls

Subsequently, we compared predicted blood DNAm biomarkers among individuals with Chronic Obstructive Pulmonary Disease (COPD), asthma, and healthy controls, adjusting for age using sputum DNAm as input. Consistent with previous studies showing advanced DNAm age in COPD and asthma patients, our analysis identified elevated sputum DNAm biomarkers in COPD individuals. Notably, a previously unreported association was discovered with DNAmFitAge, indicating that COPD patients have a 5.38 older sputum DNAmFitAge relative to healthy controls after age adjustment (p=0.036). For asthma, two C+S predicted biomarkers (DNAmPhenoAge and DNAmGDF15) aligned with expectations, whereas DNAmVO2max demonstrated an inverse relationship potentially influenced by inhaler use. For COPD patients, three predicted biomarkers corresponded with expected outcomes, revealing higher mean Smoking Pack Years (14.3, p=0.031) and an average 4.24 older DNAmAge (p=0.040) (Table 4.4A).

Our study further investigated the relationship between cumulative pack years and predicted DNAm biomarkers pertinent to lung health and fitness: DNAmPackYears, DNAmFEV1, and DNAmVO2max. A strong correlation was observed between sputum and blood predicted C+S DNAmPackYears: 0.793 (p=1.4E-6) and 0.764 (p=5.6E-6), respectively. However, the C DNAmPackYears did not significantly correlate with cumulative pack years (p=0.348), likely

due to it being an inferior algorithm as demonstrated in the Leave-One-Out-Dataset (LODO) samples. Both sputum and blood-predicted C+S DNAmPackYears and DNAmVO2max exhibited the anticipated directional correlation with cumulative pack years, with DNAmVO2max inversely associated, suggesting diminished DNAmVO2max in longer-term smokers. Furthermore, both sputum and blood-predicted C+S and C DNAmFitAge displayed significant correlations with self-reported cumulative pack years, moving in the expected direction, with both C+S and C methods exhibiting comparable correlations of 0.52. These findings on DNAmVO2max and DNAmFitAge are novel, illustrating that health phenotypes associated with physical fitness and lifestyle factors like smoking can be detected using predicted blood DNAm biomarkers derived from methylation profiles of alternative tissues. This insight underscores the broader systemic implications of fitness and health habits, which manifest beyond the typically studied tissues, highlighting the potential for a more holistic understanding of health and disease (Table 4.4B).

Exercise, Diet, and Sleep Intervention

Lastly, we evaluated the responsiveness of predicted blood DNAm biomarkers to a longitudinal 8-week intervention study focusing on an exercise, diet, and sleep. Employing a main effects mixed model, we controlled for age and study duration to ascertain if fitness-related blood biomarkers demonstrated anticipated improvements in the intervention group. This analysis provided a novel opportunity to assess the effectiveness of newly developed blood DNAm fitness biomarkers. We observed a significant increase in saliva DNAmVO2max by 0.781 in the intervention group (p=0.026), marking an unprecedented finding in exercise-induced DNAm fitness biomarker improvement in just 2 months time. Additionally, blood-predicted C DNAmCystatinC (a marker of inflammation), DNAmTL, DNAmVO2max, and DNAmFEV1_noAge exhibited significant enhancements in the intervention cohort. Notably, CystatinC decreased, while telomere length, VO2max, and FEV1 increased, with the latter two showing average improvements of 0.101 liters and 0.699 mL/kg/min, respectively. These changes paralleled the saliva DNAmVO2max improvement (0.781, p=0.026), reinforcing the credibility of our C algorithm for predicting DNAmVO2max (Table 4.5).

4.3.5 Alternative Tissues

Because our TL approach incorporates information from multiple human tissues, we were interested in understanding if the algorithm is robust to tissue type. We applied the C algo-

Table 4.4: COPD, Asthma, and Healthy Control Analysis

A. Comparison to Healthy Individuals After Controlling for Age

Biomarker	Asthma Effect	Asthma p-value	COPD Effect	COPD p-value
C+S DNAmAge Prediction	1.37	0.445	4.24	0.040
C+S DNAmPhenoAge Prediction	8.04	0.044	7.87	0.072
C+S CD8.naive Prediction	20.5	0.272	46.5	0.031
C+S DNAmPACKYRS Prediction	-0.80	0.831	14.3	0.0021
C+S DNAmVO2max Prediction	0.97	0.037	0.25	0.606
C+S DNAmGDF15 Prediction	181.0	0.047	58.7	0.543
Sputum DNAmFitAge	-2.37	0.288	5.38	0.036

B. Correlation with Person Reported Cumulative Pack Years

	Sputum Biomarker		C+S Predicted Biomarker		C Predicted Biomarker	
Biomarker	Pearson R	p-value	Pearson R	p-value	Pearson R	p-value
DNAmPACKYRS	0.793	1.4E-06	0.764	5.6E-06	-0.192	0.348
DNAmVO2max	-0.479	0.013	-0.377	0.058	-0.299	0.137
DNAmFEV1_noAge	-0.220	0.281	-0.199	0.330	-0.009	0.964
DNAmFitAge	0.615	8.2E-04	0.520	0.0065	0.521	0.0063

Table 4.5: Exercise, Diet, and Sleep Intervention Effects
Controlling for Age, Study Time, and Person-Specific Variation

Outcome	Treatment Effect	p-value
C DNAmCystatinC Prediction	-16580	0.030
C DNAmTL Prediction	0.112	0.031
C DNAmFEV1_noAge Prediction	0.101	0.033
C DNAmVO2max Prediction	0.699	0.033
Saliva DNAmVO2max	0.781	0.026

rithms to three accessible tissues (lymph nodes, adipose, and muscle DNAm) in HIV-negative and HIV-positive individuals and compared the predicted DNAm biomarkers. In lymph node DNAm samples, four predicted DNAm biomarkers demonstrated significant associations in the anticipated direction with HIV status after adjusting for age. Notably, DNAmAge and DNAmPhenoAge were, on average, 6.1 and 9.4 years older, respectively, in HIV-positive individuals when accounting for chronological age (p=0.043 and p=0.012), as shown in Supplemental Table B.11. These findings suggest that lymph nodes might serve as viable alternative tissue for our C algorithms as they can capture some biologically known differences. Nevertheless, the limited sample size (n=28) necessitates cautious interpretation, particularly concerning the performance of the remaining 19 predicted DNAm biomarkers using lymphatic tissue.

Conversely, adipose and muscle tissues exhibited markedly poor performance with these algorithms. Specifically, nine predicted biomarkers in adipose tissue and four in muscle tissue were significantly associated with HIV status after controlling for age. However, all 13 of these biomarkers displayed effects opposing the expected direction. Consequently, these outcomes imply that adipose and muscle tissues are unsuitable for application with our current C algorithms. The distinct responses across tissues underscore the necessity for a nuanced approach to selecting appropriate tissues for DNAm biomarker analysis in the context of HIV status and potentially other conditions (Supplemental Table B.11).

When evaluating the performance using skin DNAm as input, the correlation between predicted and true blood DNAm biomarkers is positive for 22/23 of the biomarkers, with 10 biomarkers having significance beyond the Bonferroni threshold of 0.002, and 13 having significance at the classical 0.05 threshold (Table 4.6). The best performance is with DNAmAge (R=0.57, p=9.4E-6), DNAmGrimAgeBasedOnRealAge (R=0.57, p=1.1E-5), DNAmAgeHannum (R=0.56, p=1.6E-5), and DNAmTIMP1 (R=0.55, 1.9E-5). Interestingly, 15 biomarkers have lower MSE compared to directly calculating the blood biomarker using skin DNAm, sug-

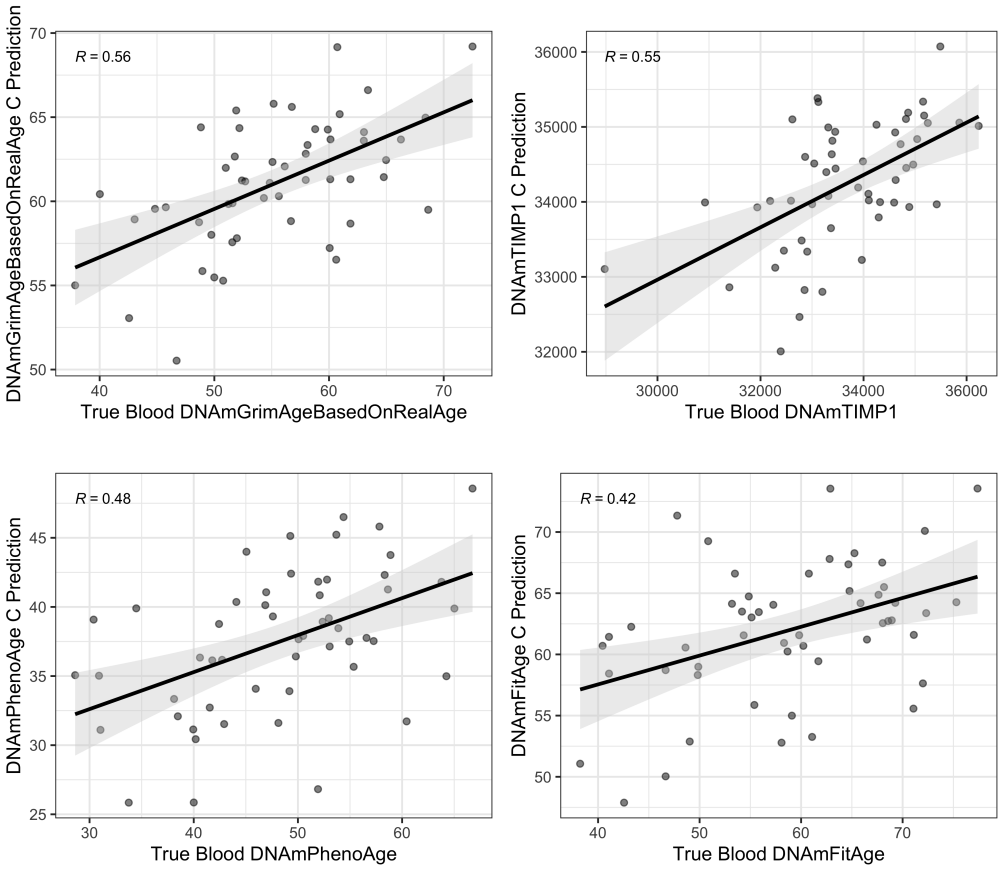


Figure 4.3: Scatterplots between True Blood DNAm Biomarkers and Skin Estimated DNAm Biomarkers using C algorithms in TwinsUK cohort (n=53).

gesting our C algorithms are preferable to direct biomarker calculation in skin. An additional 4 biomarkers have comparable MSE to their direct skin counterpart, being DNAmPACKYRS, DNAmGaitspeed, DNAmFEV1, and DNAmVO2max. The strength of correlation and low error from our predicted blood biomarkers via C algorithms demonstrate skin DNAm can be reliably used for estimating blood DNAm biomarkers.

Twenty-one out of twenty-two (95%) of the predicted biomarkers have correlations in the expected direction with chronological age (Supplemental Table B.12). Furthermore, 13 of these biomarkers have significant correlations at the Bonferroni threshold, and 15 have significance at the classical 0.05 threshold. Some of the best performing biomarkers are DNAmAge (R=0.78), DNAmFitAge (R=0.75), DNAmGrimAgeBasedOnRealAge (R=0.72), and DNAmGaitspeed (R=-0.68). These results demonstrate applying the C algorithms to skin DNAm samples preserves relationships of the predicted DNAm aging biomarkers to chronological age. As such, this further lends evidence that these algorithms can accurately reflect known aging patterns consistent with blood DNAm biomarkers and can serve as valuable surrogate alterna-

Table 4.6: Validation of Skin-Based Predicted Biomarkers Against Actual Blood and Direct Skin DNAm Biomarkers in TwinsUK Study using C Algorithms

Biomarker	Pearson R	p-value	MSE C Algorithm	MSE Skin DNAm Biomarker
DNAmAge	0.567	9.41E-06	6.29	3.81
DNAmGrimAgeBasedOnRealAge	0.565	1.06E-05	4.93	32.6
DNAmAgeHannum	0.555	1.63E-05	17.9	9.98
DNAmTIMP1	0.552	1.86E-05	820	2295
DNAmPhenoAge	0.482	2.54E-04	11.3	5.54
DNAmGDF15	0.453	6.67E-04	403.8	640.1
DNAmB2M	0.433	0.0012	259473	1524836
DNAmGrimAgeBasedOnPredictedAge	0.423	0.0016	8.61	31.4
DNAmFitAge	0.422	0.0016	5.99	19.2
DNAmGrimAge2BasedOnRealAge	0.413	0.0021	6.29	30.8
DNAmGrimAge2BasedOnPredictedAge	0.384	0.0045	8.46	29.9
DNAmCystatinC	0.376	0.005	108517	458159
DNAmTL	0.325	0.018	0.41	0.49
CD8.naive	0.258	0.062	41.5	93.8
DNAmPAI1	0.242	0.081	5927	26089
DNAmPACKYRS	0.198	0.155	16.3	14.2
DNAmGait_noAge	0.188	0.178	0.16	0.15
DNAmFEV1_noAge	0.121	0.390	0.26	0.23
DNAmADM	0.116	0.406	13.1	43.9
DNAmLeptin	0.067	0.635	4611	22475
DNAmGrip_noAge	-0.061	0.665	4.11	2.16
DNAmVO2max	0.026	0.852	1.53	1.38
CD4.naive	0.015	0.914	77.1	89.0

tives.

4.3.6 Human Biomarkers in Mammals

We apply the C algorithms to mammalian skin samples and correlate their predicted DNAm biomarkers to relative age (chronological age / maximum species age). Of the 23 applied algorithms, 6 are expected to decrease with age, 15 are expected to increase with age, and 1 doesn't have an expected direction (Leptin).

Human Biomarkers in Laboratory Animals

In primates (n=74), 16 predicted biomarkers have correlations in the expected directions with 14 of these having p-values < 0.05. Twelve of these biomarkers have p-values below the Bonferroni threshold ($p < 0.002$) (Table 4.8). Out of all DNAm biomarkers with significant signal (p-value < 0.05), 87.5% of them have strong correlations in the expected direction, performing far better than expected by our group. For example, the DNAmAge prediction has a correlation of 0.695 (p=6.1E-12) and DNAmGaitspeed has a -0.68 correlation ($p = 4.2E-11$) which are comparable correlations to those observed in the TwinsUK sample, ie humans (Supp Table B.12). Furthermore, 9 biomarkers have correlations above 0.5 including some of the widely renowned

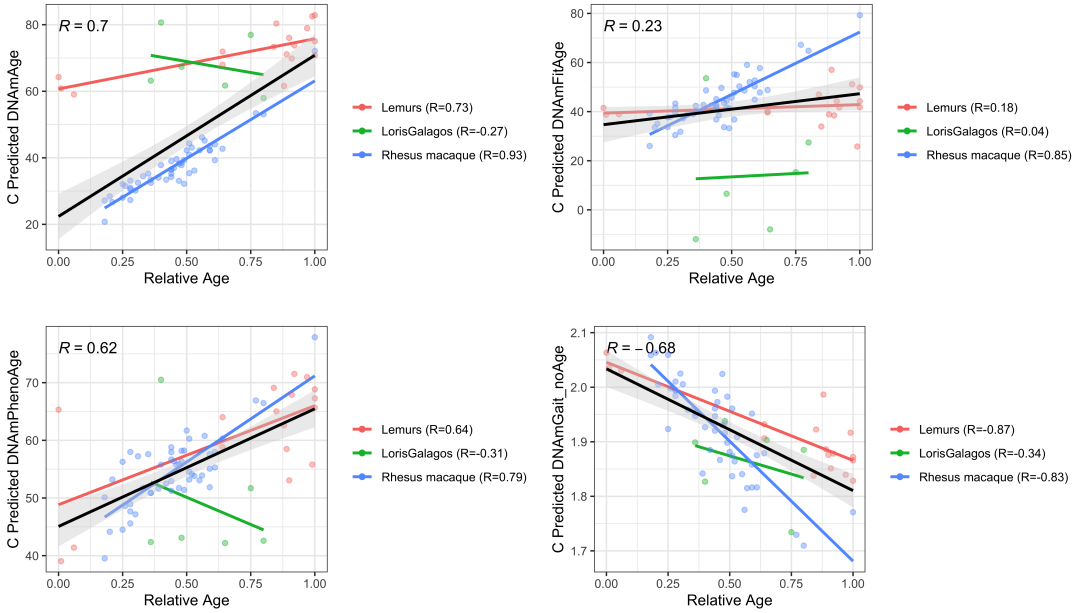


Figure 4.4: Scatterplots between Predicted Blood DNAm Biomarkers using Skin DNAm to Relative Age in Primates (n=74). Overall correlation (primates) presented on the plot and animal classification correlations presented in legend.

biomarkers like DNAmGrimAge, DNAmPhenoAge, and DNAmPAI1. Only 2 biomarkers had a significant correlation in the unexpected direction, CD8.naive and DNAmGDF15. These biomarkers consistently perform poorly across other mammalian species (strong negative correlations which are opposite of their expected direction), suggesting the C algorithms for them should not be used in mammalian skin DNAm samples.

However, the performance of the C algorithms in monkeys (Rhesus macque) have exceptionally strong performance, further suggesting that species most related to humans will have better DNAm biomarker performance. Specifically, 12 biomarkers have significant correlations at the Bonferroni threshold and an additional 5 have correlations above the 0.05 p-value threshold. All of the significant correlations are in the expected direction. The correlation magnitude between C Predicted blood DNAm biomarkers using skin methylation as input and relative age are over 0.7 for nine predicted biomarkers. For example, DNAmAge R=0.93 (p=2.5E-23), DNAmGrimAgeBasedOnRealAge R=0.85 (p=1.9E-15), DNAmFitAge R=0.85 (p=5.2E-15), and DNAmGait_noAge R=-0.83 (p=3.9E-14) (Table 4.7, Figure 4.4).

In bats (n=723), 77% (17/22) of the predicted DNAm biomarkers have significant correlations in the expected direction and only 4 biomarkers have significant correlations in the unexpected direction (Table 4.8). DNAmGrimAge, DNAmVO2max, and DNAmFitAge are some of the strongest correlations ($r > 0.3$) with p-values $<$ 1E-16. Interestingly, the strength

Table 4.7: Predicted Blood DNAm Biomarkers using Skin DNAm Correlation to Relative Age (Age / Maximum Species Age) in Rhesus Macque (n=51)

C Predicted Biomarker	Pearson R	p-value	In expected direction?
DNAmAge	0.933	2.5E-23	Yes
DNAmGrimAgeBasedOnRealAge	0.853	1.9E-15	Yes
DNAmFitAge	0.846	5.2E-15	Yes
DNAmGait_noAge	-0.832	3.9E-14	Yes
DNAmTIMP1	0.806	9.2E-13	Yes
DNAmPhenoAge	0.786	8.0E-12	Yes
DNAmGrimAgeBasedOnPredictedAge	0.756	1.4E-10	Yes
DNAmGrimAge2BasedOnRealAge	0.744	3.9E-10	Yes
DNAmAgeHannum	0.706	7.1E-09	Yes
DNAmGrimAge2BasedOnPredictedAge	0.668	8.7E-08	Yes
DNAmADM	0.516	1.1E-04	Yes
DNAmB2M	0.463	6.2E-04	Yes
DNAmPAI1	0.381	0.006	Yes
DNAmVO2max	-0.349	0.012	Yes
DNAmFEV1_noAge	-0.345	0.013	Yes
DNAmCystatinC	0.302	0.031	Yes
DNAmGrip_noAge	-0.286	0.042	Yes
DNAmGDF15	0.272	0.054	Yes
DNAmTL	-0.174	0.221	Yes
CD4.naive	-0.174	0.223	Yes
CD8.naive	-0.101	0.480	No
DNAmLeptin	-0.036	0.801	Yes
DNAmPACKYRS	-0.009	0.949	No

of correlation lowers dramatically in bat samples compared to primates and rats. Separating the correlation into fruit, micro, and vampire bats, patterns become clearer, with fruit bats having very different predicted DNAm biomarkers. For example, the correlation for DNAmFitAge is 0.25, 0.27, and -0.07 for micro, vampire, and fruit bats, respectively. Microbats also have much lower DNAmFitAge at birth compared to the fruit and vampire bats.

In rats (n=153), 16 biomarkers have correlations in the expected direction with 11 being significant at $p < 0.05$. Some of the best performing biomarkers include DNAmGrimAge-BasedonRealAge (r=0.68, p=1.8E-22), DNAmADM (r=0.66, p=3.0E-20), and DNAmTIMP1 (r=0.57, p=1.0E-14) (Table 4.8). DNAmAge seemingly has a significant correlation in the unexpected direction (r=-0.28, p=5.7E-4), but when the correlation is evaluated separately in true rats, naked mole rats, and other mole rats, the strength and directionality of DNAmAge correlation is as expected. Specifically, while the overall rat correlation was negative, r = 0.78, 0.56, and 0.36 in true rats, naked mole rats, and other mole rats, respectively. Similarly, DNAmGaitspeed's correlation is 0.39, but is -0.31 in naked mole rats. Both of these findings point to some DNAm biomarkers having strong within-species signal that are better averaged using within-species correlation.

In mice (n=48), only 4/22 biomarkers have significant correlations, and 9/22 biomarkers have correlations in the expected directions. Only 2/4 of the significant biomarkers are in the expected direction, being DNAmAge (r=0.67) and CD4.naive (r= -0.54) (Table 4.8). This lack of association is likely due to the small sample size (n=48) and concentration of samples collected around the same point in mice life (relative age = 0.5).

Across these four animal classifications, DNAmGrimAgeBasedonRealAge, DNAmTIMP1, and CD4.naive biomarkers have correlations in the expected direction in all four of these main research animal classifications. DNAmAge, DNAmPhenoAge, DNAmPAI1, DNAmPACKYRS, DNAmTL, DNAmFitAge, DNAmGripmax, DNAmADM, DNAmVO2max, and DNAmAgeHannum have correlations in the expected direction in 3/4 of the classifications. CD8.naive and DNAmGDF15 biomarkers consistently perform poorly and are not recommended for application in animal skin samples. The observed strength of these correlations, particularly in primates which are genetically closest to humans, provides compelling evidence of the C algorithms' ability to preserve the relationship to relative age. This finding is pivotal, as it supports the feasibility of using these algorithms as reliable biomarkers for aging studies in research animals. Such applications could offer invaluable insights into the human equivalents of these biomarkers

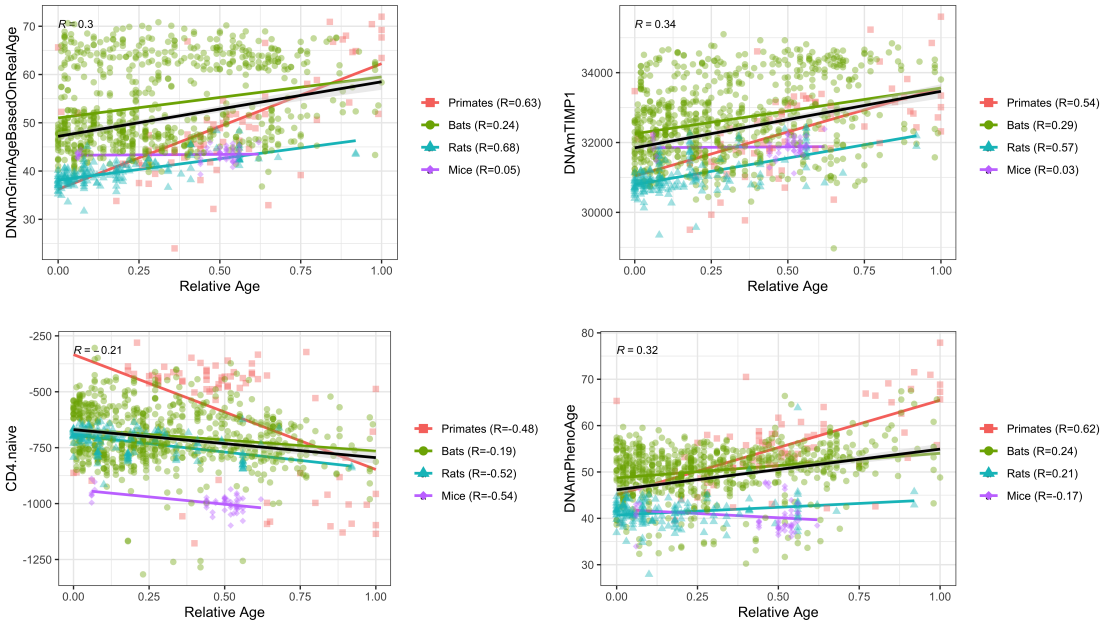


Figure 4.5: Scatterplots between Estimated Blood DNAm Biomarkers and Relative Age in Primates, Bats, Mice, and Rats. Overall correlation presented in black on figure, and within animal classification correlations presented in legend.

and the potential impacts of various interventions at the animal level. This alignment not only validates our algorithms' efficacy but also sets the stage for advancing our understanding of aging processes through animal models.

Table 4.8: Correlations between Predicted Blood DNAm Biomarkers and Relative Age in Primates, Rats, and Bats

	Primates (n=74)			Rats (n=153)			Bats (n=723)		
Biomarker	Pearson R	p-value	In expected direction?	Pearson R	p-value	In expected direction?	Pearson R	p-value	In expected direction?
DNAmAge	0.695	6.14E-12	TRUE	-0.275	5.73E-04	FALSE	0.155	2.89E-05	TRUE
DNAmGait_noAge	-0.675	4.15E-11	TRUE	0.393	5.02E-07	FALSE	0.151	4.40E-05	FALSE
DNAmGrimAgeBasedOnRealAge	0.628	2.11E-09	TRUE	0.684	1.83E-22	TRUE	0.242	4.31E-11	TRUE
DNAmPhenoAge	0.624	2.78E-09	TRUE	0.206	0.010	TRUE	0.242	4.00E-11	TRUE
DNAmGrimAgeBasedOnPredictedAge	0.577	7.19E-08	TRUE	0.087	0.285	TRUE	0.328	1.19E-19	TRUE
CD8.naive	-0.562	1.94E-07	FALSE	0.363	3.91E-06	TRUE	-0.331	5.91E-20	FALSE
DNAmCystatinC	0.556	2.66E-07	TRUE	-0.064	0.428	FALSE	-0.164	9.12E-06	FALSE
DNAmTIMP1	0.538	7.71E-07	TRUE	0.573	1.02E-14	TRUE	0.288	2.77E-15	TRUE
DNAmB2M	0.517	2.40E-06	TRUE	0.140	0.084	TRUE	-0.041	0.270	FALSE
DNAmPAI1	0.502	5.09E-06	TRUE	0.472	7.58E-10	TRUE	0.244	2.91E-11	TRUE
DNAmGrimAge2BasedOnPredictedAge	0.491	9.14E-06	TRUE	0.003	0.974	TRUE	0.258	1.73E-12	TRUE
CD4.naive	-0.479	1.59E-05	TRUE	-0.515	9.63E-12	TRUE	-0.191	2.46E-07	TRUE
DNAmPACKYRS	0.440	8.65E-05	FALSE	0.064	0.435	TRUE	0.319	1.42E-18	FALSE
DNAmGDF15	-0.411	2.78E-04	FALSE	0.319	5.76E-05	TRUE	-0.217	3.67E-09	FALSE
DNAmTL	-0.273	0.019	TRUE	0.182	0.024	FALSE	-0.125	7.93E-04	TRUE
DNAmFitAge	0.232	0.046	TRUE	0.338	1.96E-05	TRUE	0.330	8.10E-20	TRUE
DNAmGrip_noAge	-0.140	0.235	TRUE	0.151	0.062	FALSE	-0.215	5.68E-09	TRUE
DNAmADM	-0.130	0.271	FALSE	0.657	2.95E-20	TRUE	0.236	1.41E-10	TRUE
DNAmFEV1_noAge	0.129	0.275	FALSE	0.218	0.0069	FALSE	-0.270	1.65E-13	TRUE
DNAmVO2max	0.125	0.288	FALSE	-0.042	0.606	TRUE	-0.304	6.97E-17	TRUE
DNAmLeptin	0.082	0.487	TRUE	-0.044	0.590	TRUE	0.256	2.67E-12	FALSE
DNAmGrimAge2BasedOnRealAge	-0.014	0.906	FALSE	0.605	1.26E-16	TRUE	0.198	8.37E-08	TRUE
DNAmAgeHannum	0.003	0.982	TRUE	0.306	1.18E-04	TRUE	0.225	9.09E-10	TRUE

Overall and Within-Species Correlation

The correlation between predicted DNAm biomarkers and relative age is significantly related for 19 out of 23 of the biomarkers across all mammals (n=2069). Of these, 15 out of 18 biomarkers adhere to the anticipated direction of correlation (DNAmLeptin doesn't have a direction) as shown in Table 4.9. The magnitude of these correlations was modest compared to those seen in primates, with the strongest overall correlations being DNAmTIMP1 (R=0.357, p=4.6E-63), DNAmFitAge (R=0.34, p=3.7E-56), DNAmAgeHannum (R=0.32, p=1.8E-50), and DNAmVO2max (R=-0.27, p=9.1E-37) (Figure 4.6). CD8.naive and DNAmGDF15 biomarkers exhibited significant but inverse correlations, deviating from expected trends but aligning with results seen with the common laboratory animals.

The analysis of the average weighted within-species correlations revealed a tendency for biomarkers' correlations to maintain directionality observed with their overall counterparts. The biomarkers showing the most pronounced average correlations in the predicted direction were DNAmAge (R=0.45), DNAmVO2max (R=-0.31), DNAmAgeHannum (R=0.29), and DNAmFitAge (R=0.25) (Table 4.9). Twelve biomarkers have average weighted within-species correlation above 0.15, and all of those are in the expected directions. The three biomarkers with correlations in the unexpected directions overall (CD8.naive, DNAmGDF15, and DNAmPACKYRS) have low within-species average correlations ($|R| < 0.15$), further demonstrating their lack of applicability in mammalian skin samples.

When taking the average, unweighted biomarker correlations within species, many of the relationships closely parallel the weighted within-species correlations. These correlations, however, generally exhibited a slight increase in strength. For example, DNAmAge displayed a noticeable increase of 0.09 in its correlation coefficient (R=0.54), and DNAmVO2max showed a magnitude increase of 0.04 (R=-0.35) (Supp Table B.13). An exception was noted in the case of DNAmCystatinC, which shifted from a positive weighted correlation of 0.14 to a neutral correlation when unweighted.

In conclusion, the predictive accuracy of most DNAm biomarkers does not appear influenced by a few largely sampled species. Instead, our findings conclude both the weighted and unweighted within-species averages are similar and representative of their performance across different mammalian species. Biomarkers with correlation magnitude above 0.15 seem to generally perform well across and within species, with exceptions noted above.

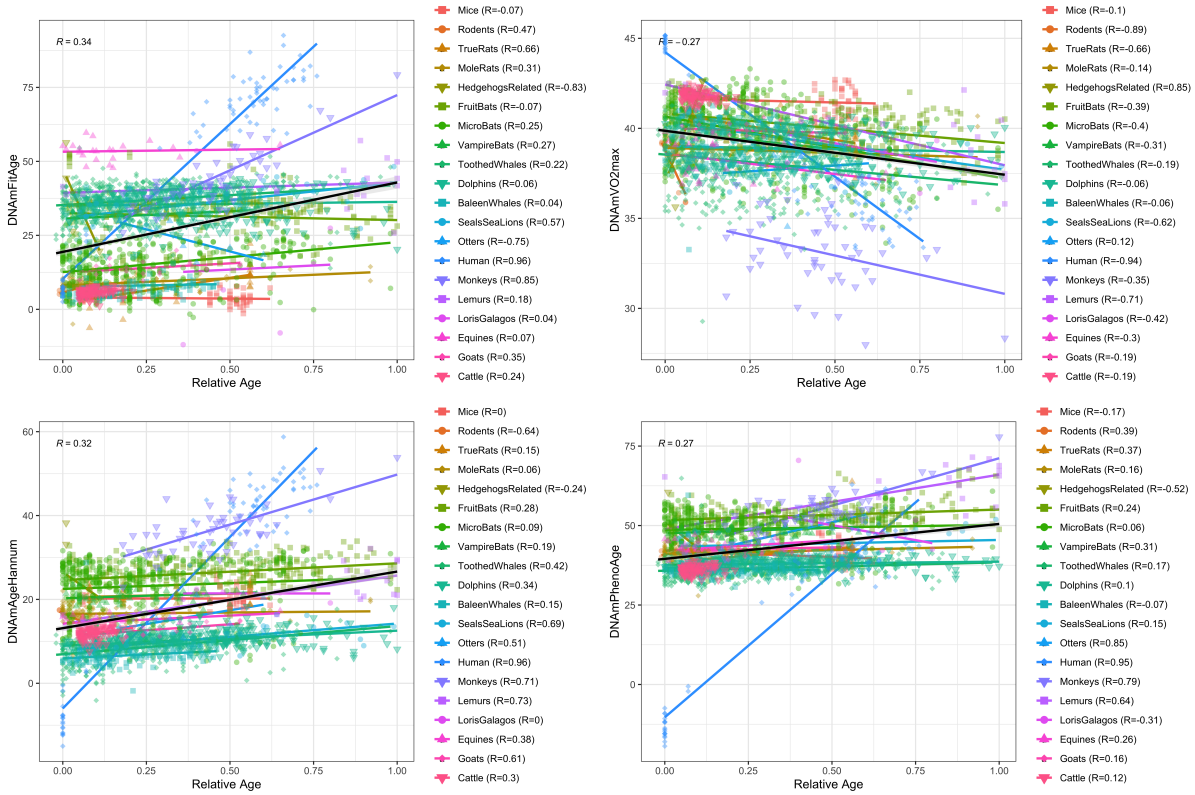


Figure 4.6: Scatterplots between Estimated Blood DNAm Biomarkers and Relative Age in All Mammals. Mammal species are grouped into smaller animal classifications for visual purposes. Overall correlation presented in black on figure, and within animal classification correlations presented in legend. Animal groupings can be found in Appendix A.7

Table 4.9: Overall Correlation and Average Weighted Within-Species Correlation

	All Mammals (n=2069)			Average Weighted Within-Species (n=2065, groups=58)		
Biomarker	Overall Pearson R	p-value	In expected direction?	Biomarker	Within-Species Pearson R	In expected direction?
DNAmTIMP1	0.357	4.6E-63	TRUE	DNAmAge	0.450	TRUE
DNAmFitAge	0.337	3.7E-56	TRUE	DNAmVO2max	-0.305	TRUE
DNAmAgeHannum	0.320	1.8E-50	TRUE	DNAmAgeHannum	0.285	TRUE
DNAmVO2max	-0.273	9.1E-37	TRUE	DNAmFitAge	0.250	TRUE
DNAmPhenoAge	0.272	1.8E-36	TRUE	CD4.naive	-0.241	TRUE
DNAmGrimAgeBasedOnRealAge	0.270	8.4E-36	TRUE	DNAmGrip_noAge	-0.231	TRUE
DNAmFEV1_noAge	-0.255	3.8E-32	TRUE	DNAmGrimAgeBasedOnRealAge	0.226	TRUE
DNAmGrimAge2BasedOnRealAge	0.237	9.7E-28	TRUE	DNAmGrimAgeBasedOnPredictedAge	0.200	TRUE
CD8.naive	-0.236	1.7E-27	FALSE	DNAmFEV1_noAge	-0.173	TRUE
DNAmADM	0.234	4.3E-27	TRUE	DNAmGait_noAge	-0.166	TRUE
DNAmGDF15	-0.209	8.4E-22	FALSE	DNAmPhenoAge	0.162	TRUE
DNAmGrimAge2BasedOnPredictedAge	0.205	4.4E-21	TRUE	DNAmTIMP1	0.152	TRUE
DNAmGait_noAge	-0.200	4.9E-20	TRUE	CD8.naive	-0.147	FALSE
DNAmGrip_noAge	-0.190	2.7E-18	TRUE	DNAmADM	0.141	TRUE
DNAmTL	-0.167	2.5E-14	TRUE	DNAmCystatinC	0.140	TRUE
DNAmLeptin	0.156	9.2E-13	-	DNAmGrimAge2BasedOnRealAge	0.087	TRUE
DNAmGrimAgeBasedOnPredictedAge	0.146	2.8E-11	TRUE	DNAmB2M	0.084	TRUE
DNAmAge	0.079	3.1E-04	TRUE	DNAmGDF15	-0.083	FALSE
DNAmPACKYRS	-0.057	0.009	FALSE	DNAmGrimAge2BasedOnPredictedAge	0.079	TRUE
DNAmPAI1	-0.038	0.083	FALSE	DNAmPACKYRS	-0.055	FALSE
DNAmB2M	-0.035	0.107	FALSE	DNAmLeptin	-0.018	-
CD4.naive	0.034	0.127	FALSE	DNAmPAI1	0.015	TRUE
DNAmCystatinC	0.014	0.539	TRUE	DNAmTL	-0.004	TRUE

Biomarker Stability and Consistency in Mammalian Species

To be classified as stable, the difference between the biomarkers' overall and average weighted within-species species correlation needed to be less than 0.1. Biomarkers with correlation difference under 0.05 were classified as very stable, whereas differences beyond 0.15 were labeled unstable.

Six biomarkers were very stable in their correlations, and an additional six biomarkers were stable. Specifically, DNAmVO2max, DNAmGaitspeed, DNAmAgeHannum, DNAmGripmax, and GrimAgeBasedOnRealAge were identified as highly stable biomarkers, showing strong correlations across and within species (All Mammals |R| = 0.19-0.32, WithinSpecies |R| = 0.17-0.30) (Supp Table B.13). In contrast, four biomarkers displayed less stability in their correlation patterns, including DNAmAge, CD4.naive, DNAmTIMP1, and DNAmTL. Notably, DNAmTIMP1 exhibited the strongest correlation across all mammalian species (R=0.36), yet this correlation diminished to an average of only 0.152 within individual species. Inversely, DNAmAge had a weak correlation across all mammalian species (R=0.079), but had the strongest within-species average correlation of 0.45. Deviations in biomarker correlations across all mammals and within species demonstrates biomarkers like DNAmAge can reliably be compared to other mammals within the same species, whereas biomarkers like DNAmTIMP1 may have more insight when comparing across species. Highly stable and stable biomarkers can reliably be used and compared across or within species, as the strength of relationship remains largely the same.

To understand biomarker consistency across species, we calculated the percent of species where the biomarkers' correlation was in the expected direction in species with a minimum of 10 samples, encompassing a total of 40 species. A large majority of biomarkers (17 out of 22) demonstrated correlations in the anticipated direction in over half of these species. Notably, 10 of these 22 biomarkers were consistent in at least 75% of the species examined (Supp Table B.13). The best alignment was observed with DNAmAge and DNAmVO2max, where an impressive 97.5% and 92.5% of species showed correlations in the expected direction, respectively. This was closely followed by DNAmGripmax and DNAmAgeHannum at 85% consistency and DNAmGrimAgeBasedOnRealAge and DNAmFitAge at 80% consistency rate. Despite the relatively small sample sizes within some species, 75% of the 40 species showed significant correlations (p-value ¡ 0.05) between DNAmAge and relative age in the predicted direction. DNAmVO2max and DNAmFitAge also demonstrated notable consistency and significance, with 52.5% and 45%

of species, respectively, showing significant correlations in the expected directions.

Intriguingly, biomarkers related to physical fitness emerged as some of the most consistent and predictive across the mammalian spectrum. This consistency underscores a potential universality of the biological benefits of fitness, suggesting that the underlying mechanisms of physical fitness and biological aging may be deeply conserved across mammalian species. As a result, these fitness-related biomarkers could provide a rich area for cross-species aging studies, particularly focusing on CpG sites conserved across mammals. This insight not only affirms the validity of these biomarkers in capturing essential aging processes but also highlights their potential as valuable tools for comparative biological research.

Predicted Biomarkers Demonstrate Rejuvenation in Mice

We applied the C algorithms to mice skin DNAm samples to predict their blood DNAm biomarkers, and twelve C predicted DNAm biomarkers have significant rejuvenation in the long treatment group ($p< 0.05$). All 12 are in the expected direction even after controlling for relative age. For example, many DNAm biomarkers are estimated to be younger in the long treatment mice group after adjusting for relative age: DNAmPhenoAge = -3.72 (p=0.022), DNAmGrimAge = -3.79 (p=0.023), DNAmAge = -1.88 (p=0.003), DNAmAgeHannum = -2.81 (p=0.008), and DNAmFitAge = -2.3 (p=0.03) (Table 4.10, Figure 4.7). Eight additional C predicted DNAm biomarkers have effects in the expected direction but are not significant at the $p< 0.05$ threshold. Only 2 biomarkers have insignificant associations in the unexpected direction (DNAmGDF15 and DNAmGaitspeed), validating 20 out of 22 of the C algorithms for capturing rejuvenation effects. Table 4.10 presents the regression coefficients and p-values compared to the control group. Figure 4.7 displays DNAm biomarker effect sizes for effect sizes under 10. The consistency of results across 91% (20/22) of the tested biomarkers emphasizes the C algorithms' potential as surrogate aging biomarkers in animal models, offering a groundbreaking tool for exploring aging and rejuvenation in mammalian models with surrogate human biomarkers.

4.4 Discussion

In this study, we explored the utility of Transfer Learning (TL) methodologies for predicting blood DNA methylation (DNAm) biomarkers from saliva DNAm, a critical advancement in the non-invasive assessment of epigenetic biomarkers. Our comprehensive approach not only sheds light on optimal TL application strategies but also provides researchers with tools to

Table 4.10: Partial Reprogramming Effects of Mice Compared to Controls using Skin DNAm as input to C Algorithms to Predict Blood DNAm Biomarkers

Biomarker	Long Trt Effect	Long Trt p-value	Short Trt Effect	Short Trt p-value	Relative Age Coef	Relative Age p-value
DNAmAge	-1.88	0.0029	-1.03	0.229	10.19	1.65E-06
DNAmPhenoAge	-3.72	0.022	-2.04	0.368	5.88	0.193
DNAmGDF15	3.26	0.922	-90.0	0.075	-24.7	0.798
DNAmPAI1	-435	0.016	3.82	0.988	91.6	0.853
DNAmGrimAgeBasedOnPredictedAge	-3.79	0.023	-2.19	0.348	5.71	0.216
DNAmADM	-4.95	0.274	-19.2	0.006	22.3	0.093
CD8.naive	19.0	0.046	17.6	0.197	-59.6	0.031
CD4.naive	26.3	0.083	24.1	0.271	-153	0.001
DNAmTL	0.07	0.0016	-0.03	0.256	-0.03	0.533
DNAmB2M	-30083	0.0062	-19397	0.201	24501	0.408
DNAmCystatinC	-7025	0.153	-2.86	1.00	-4425	0.751
DNAmLeptin	-1279	0.114	413	0.721	635	0.780
DNAmPACKYRS	-3.98	0.138	-1.04	0.788	3.44	0.651
DNAmTIMP1	-79.1	0.484	366	0.034	-734	0.031
DNAmAgeHannum	-2.81	0.0080	-0.76	0.602	6.08	0.041
DNAmGrimAge2BasedOnPredictedAge	-3.79	0.0084	-0.19	0.925	2.86	0.462
DNAmGrimAge2BasedOnRealAge	-2.87	0.013	-3.04	0.064	3.96	0.213
DNAmGrimAgeBasedOnRealAge	-2.01	0.0079	-0.44	0.673	2.61	0.207
DNAmGait_noAge	-0.01	0.262	-0.01	0.429	0.03	0.258
DNAmGrip_noAge	0.39	0.107	0.73	0.034	-2.95	1.28E-04
DNAmVO2max	0.01	0.921	0.04	0.821	-0.68	0.077
DNAmFEV1_noAge	0.04	0.128	0.02	0.475	-0.14	0.040
DNAmFitAge	-2.30	0.030	-2.68	0.067	6.40	0.030

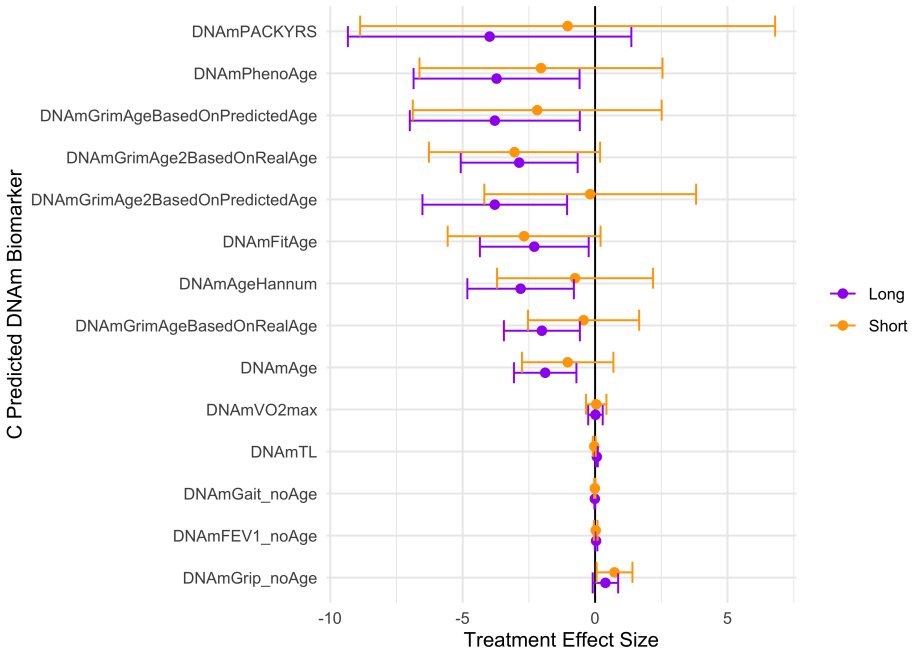


Figure 4.7: Effect Sizes and 95% Confidence Intervals of (n=3) and Long (n=7) Partial Reprogramming in Mice Compared to Controls. Predicted C Biomarkers are the estimated blood DNAm biomarker when using skin DNA methylation as input to the C algorithms. All effect sizes are adjusted for relative age, and DNAm fitness biomarkers are additionally adjusted for sex.

estimate 24 blood DNAm biomarkers from saliva, as well as functions to incorporate TL into their work. This research underscores the potential of TL to enhance the prediction accuracy and development of DNAm biomarkers.

Through extensive experimentation with different parameter settings and auxiliary data configurations, we identified optimal strategies for applying transfer learning in our context. This includes determining the best set of auxiliary data, the informative auxiliary set, and tuning the combination of model coefficients for improved DNAm biomarker prediction. Notably, our models demonstrated superior blood DNAm biomarker prediction accuracy from saliva compared to saliva surrogates and Lasso models, highlighting the effectiveness of TL in integrating diverse datasets and the utility of incorporating auxiliary data in high-dimensional settings.

A critical aspect of TL is the integration of auxiliary datasets, and our exploration into oracle and estimation-based methods for determining dataset informativeness addresses a broader challenge in TL: the need to leverage additional data without compromising or misdirecting the predictive focus of the model. Our comprehensive analysis, coupled with provided functions and recommendations, empowers researchers to employ our methods and final algorithms for cross-tissue DNAm biomarker prediction effectively. The Oracle 1df method, in particular, demonstrated good performance across both C+S and C methods. The divergence in optimal coefficient thresholding between the C+S and C methods suggests that the inclusion of saliva DNAm biomarkers requires a different approach to noise control compared to using CpGs alone.

The ability of TL predictions to reflect known biological trends validates the biological relevance of our algorithms. We demonstrate they reflect the longer telomeres in females and successfully differentiate between COPD, asthma, and healthy controls, with certain predicted biomarkers aligning with known disease characteristics. We show that predicted DNAm fitness biomarkers are responsive to an intervention study, demonstrating the promise of our algorithms for monitoring lifestyle changes and health interventions. This not only validates our algorithms but also illustrates their potential for novel discoveries, as evidenced by the identification of elevated sputum DNAmFitAge in COPD patients and increased saliva DNAmVO2max following an 8 week exercise intervention. These findings demonstrate recently developed fitness DNAm biomarkers can be measured in saliva, expanding the horizons of non-invasive health monitoring.

We provide 34 algorithms to estimate 24 blood DNAm biomarkers from saliva DNAm. We provide a comprehensive outline for preferred algorithm usage in various settings, and include 10 additional C algorithms with good predictive power for cases where saliva DNAm biomarkers

are unmeasurable - like due to array differences. By developing our algorithms on CpG loci measured across multiple arrays, we ensure broad compatibility and utility.

The strength of our study lies in its extensive evaluation of TL algorithms across various parameters, coupled with comparisons to conventional methods. This provides a robust framework for understanding the relative performance of different approaches and offers detailed guidelines for employing TL in practice. The versatility and broader applicability of our algorithms are demonstrated through their successful application to various validation datasets, including alternative tissues. The ability of the algorithms to perform well in other tissues, like lymph nodes, provides promise for their application and reliability in other tissues by capturing shared information across different DNA methylation profiles. However, the observed poor performance in adipose and muscle tissues highlights the necessity of verifying biomarker reliability when applied to non-saliva tissues, emphasizing the need for careful tissue selection in research and clinical applications.

Our analysis validates the use of skin DNAm as a viable input for our algorithms, setting the stage for their application in non-human species. For a majority of biomarkers (22 out of 23), a positive correlation was observed between the algorithm-predicted and actual blood DNAm biomarkers, with notable significance achieved beyond the stringent Bonferroni threshold (0.002) for 10 biomarkers and the conventional threshold (0.05) for 13 biomarkers. Remarkably, 19 biomarkers exhibited lower or comparable MSE using our C algorithms compared to direct biomarker calculation from skin DNAm, suggesting a preferable alternative for accurate DNAm biomarker estimation. These findings are bolstered by the strong correlations (95%) of predicted biomarkers with chronological age, where several key biomarkers demonstrated significant associations. We showcase the novel capability of the C algorithms to detect rejuvenation effects through partial reprogramming in mice. This further validates our C algorithms, as our predicted human DNAm biomarkers can capture epigenetic rejuvenation effects from partial reprogramming. Collectively, these results not only affirm the efficacy of our algorithms in accurately mirroring aging patterns akin to blood DNAm biomarkers but also establish skin DNAm as a reliable source for biomarker estimation with the C algorithms, thereby enhancing the scope of their utility in diverse biological research contexts.

Our approach, rooted in the familiar terrain of penalized linear regression, aims to offer epigenetic and aging researchers advancements without overwhelming complexity. While the simplicity of linear regression broadens accessibility and improves its potential methodological

integration, it does have limitations. For example, linear regression assumes a linear relationship between the saliva CpGs and the DNAm biomarker. Other methods, like Random forest and boosted tree models, may be better able to incorporate non-linear relationships, such as the presence of SNPs into models. Future research could explore TL frameworks incorporating non-linear or advanced regression models, potentially capturing complex biological relationships. Furthermore, our methods only employed 1 method for estimating auxiliary data informativeness, and future research could explore other methods not based in marginal associations.

These insights open new avenues for non-invasive biomarker development and facilitate cross-tissue studies. The methodologies we propose are particularly advantageous for creating biomarkers in tissues that are traditionally challenging to access or available only in limited quantities, such as the brain and other internal organs. Additionally, given expansive data derived from animal models, we envision our outlined TL methods as ways for future research to integrate animal data as auxiliary datasets. The capacity of our approaches to accurately estimate informative datasets is a crucial asset in this context. It acts as a safeguard, ensuring that non-informative or distantly related data does not compromise the model's integrity. This approach has the potential to significantly accelerate biomarker development, particularly for human tissues that are typically elusive to study. We strongly encourage researchers to adopt these TL strategies, not only as a means to expand the utility of extensive animal data but also to transform it into actionable insights and human biomarkers.

The study's results provide valuable insights into the optimal parameters and comparative performance of TL algorithms in predicting blood DNAm biomarkers from saliva DNAm. The findings demonstrate the potential of TL to enhance the prediction accuracy of DNAm biomarkers, offer guidelines for selecting the right TL approach, and showcase the algorithms' applicability to various biological and clinical scenarios. These contributions significantly advance the field of epigenetic research and open new avenues for non-invasive biomarker development and cross-tissue and/or cross-species studies.

5 Final Discussion

Our research endeavors have collectively advanced the understanding and use of DNA methylation data in the realm of physical fitness, data imputation accuracy, and transfer learning methodologies. This integrated discussion synthesizes insights from three distinct arms of the dissertation, highlighting their interconnected contributions to epigenetic and biostatistical research.

Novel DNAm Biomarkers for Physical Fitness

We have introduced groundbreaking DNAm biomarkers for key fitness parameters including maximum handgrip strength, gait speed, FEV1, and VO2max. These biomarkers, constituting DNAmFitAge, offer a novel epigenetic perspective on biological age by integrating physical fitness with DNAm-based mortality risk estimates. Our findings underscore the association between physical fitness and younger biological ages, evidenced through extensive validation across phenotypic outcomes in diverse aging datasets and observations in athletic populations. These biomarkers provide the epigenetic community tools to understand the molecular benefits of exercise captured through blood methylation.

Transformation Methods in DNAm Imputation

Our exploration into the efficacy of copula-based transformation to improve DNAm imputation addresses common complexities and limitations in traditional imputation for DNAm and other continuous outcomes. By addressing the inherent asymmetry and non-gaussian distribution in methylation values, our approach leads to a more accurate representation of methylation values. This methodological innovation, validated across various tissues and arrays, underscores the importance of considering the potential biases in standard imputation analyses. Our findings advocate for the integration of probe-specific information into the imputation process, enhancing the accuracy and reliability of epigenetic studies. Our results and transformation algorithms provide researchers methods to ensure their data meet common statistical assumptions of normality without losing interpretability or scale in their imputed values.

Transfer Learning for Cross-Tissue DNAm Biomarker Prediction

The third arm of our research illuminates the utility of transfer learning in predicting DNAm biomarkers from alternative tissues and in developing new DNAm biomarkers. By identifying optimal TL application strategies and providing tools for researchers to implement our TL methods, we provide workflows for other researchers to integrate information across diverse, yet related epigenetic studies. We demonstrate the power of TL in estimating blood DNAm biomarkers from saliva, with the development of 34 algorithms to estimate 24 blood DNAm biomarkers from saliva DNAm. Our analyses reveal our algorithms can accurately predict blood DNAm biomarkers using saliva and other tissue DNAm as input. Beyond this, we show they accurately reflect known aging relationships and accurately capture epigenetic rejuvenation effects, thereby validating their biological relevance and practical utility.

5.0.1 Cross-Domain Insights

Collectively, these studies underscore the complexity and potential of leveraging high-dimensional DNAm data across different contexts. Within each project, our research identifies and utilizes previously overlooked information embedded within the high-dimensional methylation landscape. By using blood methylation levels to make fitness biomarkers, enhancing methylation imputation through distributional transformations, and harnessing shared information across studies and tissues for biomarker prediction, our research showcases innovative ways to extract and leverage hidden insights from methylation data. Our discoveries underscore the richness of DNAm data, even when faced with the challenges posed by high dimensionality, missing values, small sample sizes, and seemingly distally related biological contexts.

Central to our methodology is the foundation on simple yet powerful statistical concepts, such as penalized regression models and the normal distribution, ensuring that the complexity of our approaches does not deter researchers. By adhering to these well-understood principles, we aim to democratize access to advanced epigenetic analysis, making our methods both accessible and actionable for a broad audience. We extend accessibility to include code, functions, and algorithms for all sections of my dissertation in GitHub repositories. Collectively, our research contributes to a deeper understanding of epigenetics and the suite of biostatistical tools we can use or adapt to gain insight to such data.

5.0.2 Possible Research Project Extensions

Future epigenetic research projects can integrate multiple parts of our research together to develop novel outcomes and algorithms. One such project can capitalize on our TL methodology (3rd arm) to improve our fitness biomarkers (1st arm). While we were able to capture information in blood tissue methylation to measure fitness biomarkers, tissues more directly related to fitness, like muscle, bone, or adipose, may be better tissues to build such tools in. However, only small datasets exist with methylation in such tissues, which naturally opens up the opportunity for our TL approach to improve fitness biomarker prediction. Additionally, other projects may be interested in predicting individual methylation levels in different tissues. These researchers could benefit from incorporating our second and third projects together. For example, instead of developing models to predict tissue k's CpG locus j's methylation beta value, x_{kj}, models could instead be built to predict the normal-transformed j's level, z_{kj}. Both target and auxiliary data sources would have their outcomes transformed into gaussian variables using the original methylation distribution. By doing this initial transformation step, the methylation error will now be normally distributed, allowing researchers to properly implement our penalized linear regression TL technique.

5.0.3 Future Research Advancements and Open Questions

Beyond developing biomarkers, there is also substantial room to use DNAm data as a way to test and improve new transfer learning methodologies. Researchers can vary how auxiliary datasets are calculated as informative, and potentially leverage the commonalities in the covariance structure of X. In our current research, we used marginal correlations between individual CpG levels and the outcome of interest to calculate informativeness. Research could instead classify auxiliary data informativeness using information remaining after removing similarities to the target source. Instead of the similarity to marginal associations between auxiliary and target data, the strength of signal left between covariates and outcome residuals (by applying the coefficients from the initial target data model) could capture additional informativeness. As such, this would rank auxiliary datasets as more informative if they had more unique, additional information that the target data did not already capture.

Future research should continue to explore the integration of auxiliary data from diverse sources, particularly focusing on the refinement of TL methodologies to accommodate the het-

erogeneity inherent in epigenetic data. Additionally, the exploration of non-linear or advanced regression models could capture complex biological relationships more effectively, broadening the scope of DNAm biomarker prediction and application.

5.0.4 Final Remarks

In conclusion, our integrated findings highlight the innovative application of DNAm biomarkers to assess physical fitness, the critical role of transformation methods in DNAm imputation, and the promising use of TL for cross-tissue biomarker prediction and development. This comprehensive approach not only advances the field of epigenetics but also sets the stage for future breakthroughs in non-invasive biomarker development and cross-tissue or cross-species studies, thereby contributing significantly to personalized medicine and the broader understanding of aging.

APPENDICES

Appendix A

A.1 Functional CpG Annotation

I provide biological insight to the 627 unique CpG loci used in constructing our DNAm fitness biomarkers by exploring genomic enrichment in the entire human genome and analyzing specific enrichment in chromatin states.

A.1.1 GREAT

I use the Genomic Regions Enrichment of Annotations Tool (GREAT) for analyzing broad genomic enrichment [83]. GREAT analyzes the genes within and nearby the genomic region covered by the CpGs. To avoid confounding the enrichment analysis by gene size, the GREAT algorithm performs a binomial test (over genomic regions) using a whole genome background. We performed the enrichment based on default settings (Proximal: 5.0 kb upstream, 1.0 kb downstream, plus Distal: up to 1,000 kb) using the hg19 assembly. We report nominal, Bonferroni, and FDR p-values for gene, biological, cellular, and molecular function in Table 5A for the top results.

The CpG loci were enriched in 5 gene sets, 11 cellular processes, and 7 molecular processes mostly related to inflammation at FDR Q-value < 0.05 (Table B.6). The top genes enriched include zinc ribbon domain containing 1 (ZNRD1; Bonferroni p=0.005) and histocompatibility antigen (HLA-G; p=0.02). Cellular processes relate to major histocompatibility complex (MHC) proteins (p=3.1E-7) and molecular processes relate to peptide antigen binding (p=0.032) and tapasin binding (p=0.047). Tapasin is a MHC class I antigen-processing molecule present in the lumen of the endoplasmic reticulum [84]. The relationship to inflammation-based genes and processes like HLA, MHC, and tapasin support hypotheses relating physical fitness and systemic inflammation [85]. In addition, previous research found inflammation response and endoplasmic reticulum stress were down-regulated in people following a 12-week endurance exercise regime compared to the non-exercising control group [86]. Both biological findings are intriguing and

may provide direction for studying modifiable methylation from fitness parameters.

A.1.2 Chromatin States

To annotate the CpGs used to construct the DNAm fitness biomarkers based on chromatin state, we assigned a state for the CpGs based on the detailed universal ChromHMM chromatin state annotation of the human genome in which chromatin structure and their associated characteristics are annotated [87]. This annotation generated 100 distinct states using 1,032 experiments into 16 major categories such as weak enhancers (EnhW) and flanking promoter states (PromF). We used one-sided hypergeometric tests to study both the enrichment (OR > 1) and depletion (OR < 1) patterns of CpGs across the chromatin states as detailed in [88]. Genomic CpG regions on the 450K array with chromatin state information were used as background (n=483,090). The genomic regions of DNAm fitness biomarker CpG sites with chromatin state information were used as foreground (n=626), which only excluded 1 CpG. This yielded one-sided hypergeometric p-values not confounded by the number of CpGs within a gene. We report the chromatin state, number of CpG loci enriched in each state, Odds Ratios, and hypergeometric p-values in Table B.6B, and complete results are presented in Supplemental Table 10. Because the underlying chromatin states follow a multinomial distribution, we do not adjust our p-values for multiple comparisons.

The chromatin states are significantly depleted in heavily acetylated promoters and transcription start sites (TSS) and enriched in regions with polycomb repressive complex 2 (PRC2) binding (Table B.6). The odds ratios (OR) are significantly less than one in the chromatin state PromF4 (heavily acetylated promoters, OR=0.45, hypergeometric p=6.5E-6) and TSS1 (acetylated TSS, OR=0.37, p=6.8E-6) (Table 5B). BivProm1 (OR=1.50, p=0.009), BivProm2 (OR=1.76, p=0.0006), and ReprPC1 (OR=1.87, p=0.007) regions are enriched in our DNAm fitness biomarkers and are known PRC2 binding sites [87]. BivProm1 and BivProm2 are weak bivalent promoters and ReprPC1 is a polycomb repressed region. Bivalent chromatin domains control expression of HOX and other developmental genes in all vertebrates. PRC2 is one of the main Polycomb repressive complexes (PRC) that act as negative epigenetic regulators of transcription; it helps to initiate gene silencing via H3K27 methylation [89]. These results coincide with the increasing observation that the process of development is connected to epigenetic aging and that PRC2 targets are enriched in the age-dependent methylome in human and mammals [6, 79].

A.2 DNAmFitAge Validation Datasets

The Budapest dataset was used as the training dataset for the DNAmVO2max biomarker. For the other biomarkers, this dataset was used for validation. The additional validation datasets involved six cohorts: the Lothian Birth Cohorts (1921 and 1936), Comprehensive Assessment of Long-term Effects of Reducing Intake of Energy (CALERIE), the Women's Health Initiative (WHI), Jackson Heart Study (JHS), and Invecchiare in Chianti, aging in the Chianti area (InChianti). In addition, the Polish Study is used to evaluate biomarkers across body builders and controls. Below we describe each study cohort/datasets in more detail.

Budapest

Budapest is a small, novel study (n=307) measuring physical fitness and DNA methylation in middle to older aged adults, some of whom are current or former athletes. A total of n=205 participants previously participated in the World Rowing Masters Regatta in Velence, Hungary. The study was approved by the National Public Health Center in accordance with the Helsinki Declaration and the regulations applicable in Hungary (25167-6/2019/EÜIG). This research study was undertaken by the Research Institute of Sport Science, Hungarian University of Sport Science, Budapest. Subjects completed a questionnaire regarding their health, educational status, and life-style- including exercise habits. Maximum hand gripping force was assessed using the CAMRY EH101 dynamometer. Relative maximal oxygen uptake (VO2max) was measured using the Chester step test on a treadmill. The strength of the legs (Jumpmax) was assessed by a person's maximal vertical jump, measured using a linear encoder.

Budapest DNAm methylation quantification

Epigenome wide DNA methylation was measured with the Infinium MethylationEPIC BeadChip (Illumina Inc., San Diego, CA) according to the manufacturer's protocol. DNA methylation was derived from whole blood samples using 500 ng of genomic DNA. Quality control of DNA methylation was performed using minfi, Meffil, and ewastools packages with R version 4.0.0. Samples which failed technical controls, including extension, hybridization and bisulfite conversion, according to the criteria set by Illumina, were excluded. Samples with a call rate $< 96\%$ or at least with 4% of undetected probes were also excluded. Probes with a detection p-value > 0.01 in at least 10% of the samples were set as undetected. Probes with a bead number < 3 in at least 10% of the samples were excluded. Methylation beta values were generated

using the Bioconductor minfi package in R with Noob normalization background correction.

Lothian Birth Cohorts

The Lothian Birth Cohorts (LBC) consists of two longitudinal studies evaluating cognition and brain aging of older adults who were born in either 1921 (LBC1921) or 1936 (LBC1936) and lived in Edinburgh or the surrounding Lothian regions of Scotland. LBC1921 was started in 1999 and LBC1936 began in 2004. LBC1936 was established to study cognitive aging in surviving members of the 1947 Scottish Mental Survey. Ethical approval was obtained from the Multi-Centre Ethics Committee for Scotland and Lothian Research Ethics Committee. National Records of Scotland provided regular updates on mortality data for the LBC participants via data linkage with the National Health Service Central Register.

LBC1921

Participants were born in 1921 and most completed a cognitive ability test around age of 11 years in the Scottish Mental Survey 1932 (SMS1932). The SMS1932 was administered nationwide to almost all 1921-born children who attended school in Scotland in June 1932. The cognitive test was the Moray House Test No. 12. The LBC1921 study attempted to follow up individuals who might have completed the SMS1932 and resided in the Lothian region (Edinburgh and its surrounding areas) of Scotland; 550 people (N=234, 43% men) were successfully traced and participated in the study from the age of 79 years. To date, there have been four additional follow-up waves at average ages of 83, 87, 90, and 92 years. The cohort has been studied during the later-life waves, including blood biomarkers, cognitive testing, and psycho-social, lifestyle, and health measures.

LBC1936

The methylation mortality survival analysis was investigated in LBC1936. All participants were born in 1936 and most had taken part in the Scottish Mental Survey 1947. These participants attended Scottish schools in June 1947. The cognitive test administered was the same Moray House Test No. 12. A total of 1,091 participants (n=548, 50% men) who were living in the Edinburgh and Lothian area of Scotland were re-contacted in later life. Data has since been collected in waves at five time points.

Whole blood DNA methylation was measured using the Illumina HumanMethylation 450BeadChips from 514 whole blood samples in LBC1921 and from 1,004 samples in LBC1936. Raw intensity data were background-corrected and methylation beta-values generated using the

R minfi package. Quality control analysis was performed to remove probes with a low (<95%) detection rate at $P < 0.01$. Manual inspection of the array control probe signals was used to identify and remove low quality samples (for example, samples with inadequate hybridization, bisulfite conversion, nucleotide extension, or staining signal).

CALERIE

Comprehensive Assessment of Long-term Effects of Reducing Intake of Energy (CALERIE) was a Phase 2 clinical trial started in 2007 studying young to middle-aged healthy adults [47]. CALERIE is the first clinical trial to focus on the effects of sustained CR in humans. It was completed in May 2013 as a two-year three-site randomized controlled trial in young and middle-aged non-obese healthy men and women (N = 220). Participants were randomized in a 2:1 fashion to 25% caloric restriction (CR) or ad libitum control group (diet is available at all times). All participants needed to have a baseline body mass index (BMI) of 22-27.9 kg/m^2 (lean to slightly overweight). Each participant has 1) behavioral counselor (Masters of doctoral in psychology) AND 2) registered dietician who follow with them for the whole 2 years. 25% reduction and caloric goals are calculated based on each person's initial food intake at baseline. They must meet with the dietician 2-3 times a week and record food intake. Two consecutive 14-day doubly labeled water studies are conducted with each participant at baseline with the average used to determine AL TEE (total energy expenditure); from this, the 25% CR prescription for that participant is derived. An average of 12% caloric reduction was achieved in the CR group throughout the study.

DNA methylation was measured from Illumina EPIC 850k Arrays (Illumina Inc., San Diego, CA) as per the manufacturer's protocol. DNA methylation was derived from whole blood samples. CALERIE methylation assays were run by the Molecular Genomics Shared Resource at Duke Molecular Physiology Institute, Duke University (USA). Quality control of sample handling included comparison of clinically reported sex versus sex of the same samples determined by analysis of methylation levels of CpG sites on the X chromosome. Methylation beta values were generated using the Bioconductor minfi package with Noob background correction. CALERIE data are available at https://calerie.duke.edu/samples-data-access-and-analysis.

Women's Health Initiative

The WHI is a national study that enrolled postmenopausal women aged 50-79 years into the clinical trials (CT) or observational study (OS) cohorts between 1993 and 1998 [90]. We included 4,079 WHI participants with available phenotype and DNA methylation array data: 2,107 women from "Broad Agency Award 23" (WHI BA23). WHI BA23 focuses on identifying miRNA and genomic biomarkers of coronary heart disease (CHD), integrating the biomarkers into diagnostic and prognostic predictors of CHD and other related phenotypes, and other objectives can be found in
https://www.whi.org/researchers/data/WHIStudies/StudySites/BA23/Pages/home.aspx. The total number of age-related conditions was based on Alzheimer's disease, amyotrophic lateral sclerosis, arthritis, cancer, cataract, CVD, glaucoma, emphysema, hypertension, and osteoporosis.

Bisulfite conversion using the Zymo EZ DNA Methylation Kit (Zymo Research, Orange, CA, USA) as well as subsequent hybridization of the HumanMethylation450k Bead Chip (Illumina, San Diego, CA), and scanning (iScan, Illumina) were performed according to the manufacturers protocols by applying standard settings. DNA methylation levels were determined by calculating the ratio of intensities between methylated (signal A) and un-methylated (signal B) sites.

Jackson Heart Study

The JHS is a large, population-based observational study evaluating the etiology of cardiovascular, renal, and respiratory diseases among African Americans residing in the three counties (Hinds, Madison, and Rankin) that make up the Jackson, Mississippi metropolitan area. The age at enrollment for the unrelated cohort was 35-84 years; the family cohort included related individuals >21 years old. Participants provided extensive medical and social history, had an array of physical and biochemical measurements and diagnostic procedures, and provided genomic DNA during a baseline examination (2000-2004) and two follow-up examinations (2005-2008 and 2009-2012). Annual follow-up interviews and cohort surveillance are ongoing. In our analysis, we used the visits at baseline from 1747 individuals as part of project JHS ancillary study ASN0104, available with both phenotype and DNA methylation array data. Total numbers of age-related conditions were based on hypertension, type 2 diabetes, kidney dysfunction based on ever dialysis, and CVD.

Peripheral blood samples were collected at the baseline. DNA was extracted using the Gentra Puregene blood kit (Gentra System, MN, Minnesota, USA). Methylation beta values were generated using the Bioconductor minfi package with Noob background correction.

Invecchiare in Chianti, aging in the Chianti area (InChianti)

The InChianti (Invecchiare in Chianti, aging in the Chianti area) cohort is a representative population-based study of older persons enrolling individuals aged 20 years and older from two areas in the Chianti region of Tuscany, Italy, http://inchiantistudy.net/wp/. One major goal of the study is to translate epidemiological research into geriatric clinical tools, ultimately advancing clinical applications in older persons. Of the cohort, 924 observations from 484 individuals with both phenotype information and DNA methylation data were including in our studies. The observations were collected from baseline in 1998 and the third follow-up visit in 2007. All participants provided written informed consent to participate in this study. The study complied with the Declaration of Helsinki. The Italian National Institute of Research and Care on Aging Institutional Review Board approved the study protocol. We computed the total number of age-related conditions based on cancer, hypertension, myocardial infarction, Parkinson's disease, stroke and type 2 diabetes.

Genomic DNA was extracted from buffy coat samples prior to bisulfite conversion. Blood DNA methylation was taken twice over the span of nine years in a total of 966 people. CpG methylation status of 485,577 CpG sites was determined using the Illumina Infinium HumanMethylation450 BeadChip (Illumina Inc., San Diego, CA) as per the manufacturer's protocol and as previously described [11]. Threshold call rate for inclusion of samples was 95%. Quality control of sample handling included comparison of clinically reported sex versus sex of the same samples determined by analysis of methylation levels of CpG sites on the X chromosome. Methylation beta values were generated using the Bioconductor minfi package with Noob background correction.

Polish Study

The Polish Study is a small, novel study (n=416) measuring blood DNA methylation and lifestyle behaviors in Polish body builders and similar aged healthy controls ranging from 17 to 56 years of age. It is part of a larger cohort representing the general population of Poland, for which blood samples, buccal swabs or semen samples were collected as part of the local

project EPIGENOME (DOB-BIO10 / 06/2019). Participants of the Polish Study recorded the total number of years they regularly trained, average number of intensity trainings per week, sports training they participate in, and dietary supplements or drugs they take. There were a total of 66 male body builders and 30 female body builders. Because of the small sample size in females, we restricted the analysis to males only, which decreases the sample size to 215 individuals total, 149 controls and 66 body builders. 88 males in the study reported dietary supplements or drugs, and a total of 147 unique substances were reported. The use of each analyzed supplement was coded based on presence of multiple phrases in the open question of the questionnaire about drug/supplements intake. Specifically, multivitamins include reported use of vitamins, multivitamins, and vitamins + minerals. Proteins included reported use of protein supplement, branched chain amino acids (bcaa), amino acids, and training supplements. Energy supplements included creatine, energy gels, and pre-workout. Magnesium included mg and magnesium. Vitamin D consisted of vitamins D and D3. Omega-3 consisted of Omega-3 and cod liver oil.

Epigenome wide DNA methylation was measured with the Infinium MethylationEPIC BeadChip using DNA from whole blood. The quality and quantity of DNA isolates were assessed using NanoDrop 8000 UV-Vis Spectrophotometer and Qubit 4 Fluorometer. Then, the DNA concentration was normalized to 50 ng/µl and subjected to microarray analysis. Quality control and preprocessing were done using minfi and ENmix packages with R version 4.2.1. Methylation beta values were generated using the Bioconductor minfi package with Noob normalization background correction.

A.3 Body Builder Supplement Use

I evaluated whether the improvement in DNAmFitAge and DNAmVO2max in male body builders can be explained by the dietary supplements taken using a linear regression model with DNAmFitAge or DNAmVO2max as the outcome with age as a covariate and indicator variables for taking the supplement and being a body builder. We adjust for age in the model because age was significantly related to taking certain supplements, therefore if age was not included, the differences observed in DNAmFitAge or DNAmVO2max may actually represent differences in chronological ages between supplement usage groups. Linear model results are presented in Supplemental Table B.2. To ensure adequate power, we evaluated supplements and drugs with at least 10 people reporting use across both body builders and controls. Only six sup-

plements met this threshold: multivitamins (n=19), protein (n=17), energy (n=17) (creatine, pre-workout, and energy gels), magnesium (n=16), vitamin D (n=14), and omega-3 (n=12). We also evaluated if these supplements were disproportionately taken by male body builders compared to male controls using Fisher's Exact test (Supplemental Table B.3). Finally, we considered using a linear model with an interaction term between being a body builder and taking a supplement; if the interaction term was significant, then we would deem the supplement can adequately explain the improvement in DNAmFitAge or DNAmVO2max. However, the small number of subjects taking supplements would not adequately power the interaction term which would likely prevent any supplement from being significant. Therefore, we chose to use main effects to determine supplement contribution as explained above.

Dietary supplement use cannot explain improvement in DNAmFitAge, but multivitamin dietary supplements are associated with improvement in DNAmVO2max after controlling for athlete status and age in males. Males from the Polish Study who take multivitamins have a 0.68 mL/kg/sec fitter DNAmVO2max on average after adjusting for athlete status and age (p=0.041, Supplemental Table B.2). Multivitamins, energy, vitamin D, and Omega-3 all are disproportionately taken by the male body builders (Supplemental Table B.3), however, supplement use is not sufficient to explain younger DNAmFitAge regardless of athlete status (Supplemental Table B.2). These insignificant results may point to other components of athleticism that contribute to younger estimated biological ages, such as increased physical activity and decreased body fat. We note that supplement and athlete coefficients for multivitamins, proteins, and Omega-3 are statistically insignificant, but their relationships are in the expected direction for DNAmFitAge and DNAmVO2max. Our research does not establish the causative relationship of body building or supplement use on biological aging, but it does establish there are observable epigenetic benefits associated with being a male body builder.

A.4 Imputation Tools

I detail some of the methods used in the different R-based imputation tools below. These include methyLImp, a procedure that develops a linear model using Singular Value Decomposition, impute.PCA, a procedure that forms principle components of the methylation sites, and imputeknn, a procedure that uses the most similar CpG's (via Euclidean distance) to impute the CpG values. These three methods span linear regression

A.4.1 impute.PCA

impute.PCA is a regularized iterative PCA algorithm. based on the Fixed effect model, the classical PCA model is a bilinear model where

$$X_{ik} = m_k + (\mathbf{FU}')_{ik} + \epsilon_{ik}, \epsilon \sim \mathcal{N}(0, \sigma^2).$$

In this model, individuals have different expectations and the randomness is only due to the error term. Iterative PCA is an expectation maximization algorithm for this model. The regularization comes from adding a shrinkage term to reduce overfitting. Specifically, the regularized model matrix is

$$\hat{X} = \hat{\mathbf{M}} + \hat{\mathbf{Z}}\hat{\mathbf{B}}' = \hat{\mathbf{M}} + \sum_{s=1}^{S} \frac{\hat{\lambda}_s - \hat{\sigma}^2}{\hat{\lambda}_s} \mathbf{F_s U_s}' = \hat{\mathbf{M}} + \sum_{s=1}^{S} \sqrt{\hat{\lambda}_s} \times \frac{\hat{\lambda}_s - \hat{\sigma}^2}{\hat{\lambda}_s} \mathbf{F_s U_s}'$$

The singular values ($\sqrt{\hat{\lambda}_s}$) are shrunk by a ratio of the signal over signal plus noise where $\hat{\sigma}^2 = \frac{1}{K-S} \sum_{s=S+1}^{K} \hat{\lambda}_s$, indicating the first S eigenvectors with corresponding λ eigenvalues.

A.4.2 impute.knn

The 'impute.knn' algorithm is a method specifically developed for the imputation of missing values in gene expression datasets, but it is also applicable to DNA methylation (DNAm) data. This algorithm employs the k-nearest neighbors approach to estimate missing values based on a set of similar, non-missing CpG loci. The core concept of 'impute.knn' revolves around the identification of k nearest neighbors for each CpG locus with missing data, utilizing the Euclidean distance as a metric for similarity.

Given a DNAm dataset with missing values, let X represent the matrix of methylation levels, where X_{ij} indicates the methylation level of the jth CpG locus in the ith sample. For a CpG locus i with a missing value, 'impute.knn' identifies k nearest neighbors based on the Euclidean distance calculated from the non-missing values. The Euclidean distance between two samples, i and i', for all CpG loci is defined as:

$$d(i, i') = \sqrt{\sum_{j=1}^{n} (X_{ij} - X_{i'j})^2}$$

where n is the total number of CpG loci. After identifying the k nearest neighbors, the missing value X_{ij} is imputed using a weighted average of these neighbors:

$$\hat{X}_{ij} = \frac{\sum_{l=1}^{k} w_{il} X_{lj}}{\sum_{l=1}^{k} w_{il}}$$

The weight w_{il} inversely correlates with the Euclidean distance $d(i, l)$, emphasizing closer neigh-

bors more significantly in the averaging process. Commonly, weights are calculated using:

$$w_{il} = \frac{1}{d(i,l)^2}$$

to ensure that nearer neighbors contribute more to the imputed value.

The choice of k, the number of nearest neighbors, is crucial for the performance of 'impute.knn'. In the context of DNAm data, $k = 50$ is recommended based on prevalent practices and the typical structure of methylation datasets. This parameter balances the trade-off between capturing sufficient local information and avoiding the influence of distant, less relevant CpG loci.

A.5 `CONCORDANT` Functions

A.5.1 `TransformDataset` Function

Transform your dataset to pseudo gaussian variables. This function calculates and applies a normal transformation on a per column basis, allowing for missing values. The function calculates the forward and backwards transformation functions using an empirically smooth CDF to convert continuous data to pseudo gaussian variables. This function requires 3 other helper functions supplied below named, `%!in%`, `extract_transformed_values`, and `qfun_extract`. Please ensure these helper functions are in your global environment prior to calling the function.

To use this function, supply 4 arguments: `dataset`: The dataset to transform select columns to gaussian variables `columns`: The specified columns to transform (as column names) `rownamevar`: The observation ID variable to ensure the data are aligned when missing values are present. iteration: Either `TRUE` or `FALSE` to print the current iteration to track progress.

The function returns a 2 item list: `transformed_df`: has the row IDs and columns transformed to gaussian variables. Note that columns transformed have `_normal_var` added to column names `qfunctions`: the inverse functions needed to backtransform the data

```
TransformDataset <- function(dataset, columns, rownamevar, iteration=TRUE){
  # Output list initialization
  col_output <- list()

  # Preparing the dataset for transformation
  row_name <- rownames(dataset)
  dataset_prep <- data.frame(dataset[, columns], ID = row_name)

  # Last column is the ID now
  last_val <- dim(dataset_prep)[2]
  # Define the last CpG column
  last_cpg <- last_val - 1

  # Loop through each column to apply transformations
  for(i in 1:last_cpg){
```

```
  col_name <- colnames(dataset_prep)[i]

  # Prepare the data: only non-missing values are processed.
  # The data frame now has ID in the first column and CpG in the second column
  dat_prep <- dataset_prep[!is.na(dataset_prep[, i]), c(last_val, i)]

  # Density estimation for non-missing CpG values,
  # using kernel density estimation for smoothing
  # 2nd column used because the second column is the current CpG #
  density_x <- density(dat_prep[, 2], adjust = 0.5, from=min(dat_prep[, 2]),
              to=max(dat_prep[, 2]), n = 1000)

  # Transform the frequency into density function
  dapproxfun <- splinefun(x = density_x$x, y = density_x$y)
  dfun <- function(x) dapproxfun(x)
  support <- range(dat_prep[, 2])

  # Integrate the density function to generate the empirical CDF
  pfun_integrate_dfun_1 <- function(v) integrate(dfun, support[1], v,
                                                 subdivisions=2000, rel.tol =1e-15,
                                                 stop.on.error = FALSE)$value
  pfun_integrate_dfun <- function(x) Vectorize(pfun_integrate_dfun_1)(x)

  # pfun is the empirical (smoothed) CDF
  pfun <- splinefun(x = dat_prep[, 2],
                y = pfun_integrate_dfun(dat_prep[, 2]))

  # Transform to uniform space using the empirical smoothed CDF
  uniform_var <- pfun(dat_prep[, 2])

  # If some values are over 1, rescale and make sure to not have exact 1's or 0's
  if(sum(uniform_var > 1) > 0){
    uniform_var <- uniform_var / max(uniform_var)
    uniform_var <- ifelse(uniform_var == 1, 0.9999999, uniform_var)
  }
  uniform_var <- ifelse(uniform_var == 0, 0.0000001, uniform_var)

  # Compute the inverse of the empirical CDF (qfun)
  qfun <- splinefun(x = uniform_var, y = dat_prep[, 2])

  # Transform to standard normal space
  norm_var <- qnorm(uniform_var)
  # Handling infinite values by setting them to +/- 5
  norm_var <- ifelse(is.infinite(norm_var), sign(norm_var) * 5, norm_var)

  # Create dataframe with normal transformed variable and ID
  df_sm <- data.frame(norm_var, ID = dat_prep[, 1])
  colnames(df_sm) <- c(paste0(col_name, "_normal_var"), "ID")

  # Merge the original scale variable and normal_var
  df_sm2 <- merge(dataset_prep[, c("ID", col_name)], df_sm, by = "ID", all.x=TRUE)

  # Rename ID variable to original name
  colnames(df_sm2) <- c(paste0({{rownamevar}}), colnames(df_sm2)[2:3])

  # Prepare output for the current column
  col_output1 <- list(qfun, df_sm2)
  new_names <- c(paste0(col_name, "_qfun"), paste0(col_name, "_normal_var"))
  names(col_output1) <- new_names

  # Add the output to the main output list
  col_output[[i]] <- col_output1

  if(iteration == TRUE){
    print(i) # Print current iteration number for tracking progress
  }
}

# convert output to list of qfunctions and transformed dataframe
qfunctions_output <- qfun_extract(col_output)

# extract the transformed values
```

```
  transformed_df <- extract_transformed_values(col_output)

  output_return <- list(transformed_df, qfunctions_output)
  names(output_return) <- c("transformed_df", "qfunctions")
  return(output_return)
}
```

A.5.2 `BackTransformDataset` Function

Back transform your normal transformed data to the original data scale. This function applies the inverse CDF (q-function) of each column to back-transform. It only applies the backtransformation to columns with supplied q-functions.

To use this function, supply 2 arguments: `dataset`: The dataset on the gaussian scale to transform select columns back to variables on the original data scale

`qfunctions`: A list of the inverse functions to apply column-wise

The function returns a dataframe with original variable names and variables on the original datascale.

```
BackTransformDataset <- function(dataset, qfunctions){

  # Re-order the columns of the imputed data to match the order of qfunctions
  newqfuncname <- gsub("*_qfun", "_normal_var", names(qfunctions))
  col_neworder <- match(newqfuncname, colnames(dataset))
  impute_df_matched <- dataset[, col_neworder]

  # Identify the remaining columns that were imputed but don't need to be transformed
  # These will be combined with the back-transformed columns later
  rest_imputed <- colnames(dataset)[colnames(dataset) %!in% newqfuncname]
  impute_df_rest <- dataset[, rest_imputed]


  # Start back transformation process
  col_num <- length(qfunctions)
  row_num <- nrow(impute_df_matched)

  # Apply the standard normal CDF (pnorm) to the imputed data
  # This transforms it to a uniform distribution
  missnormal_rep_unif <- pnorm(impute_df_matched)

  # Pre-allocate a dataframe for the back-transformed data
  backtransform_missnormal_rep <- data.frame(matrix(NA, ncol = col_num,
                                          nrow = row_num))

  # For each column, apply the corresponding inverse CDF (q-function)
  for(i in 1:col_num){
    backtransform_missnormal_rep[, i] <- qfunctions[[i]](missnormal_rep_unif[, i])
  }

  # Restore the original column names
  new_colnames <- gsub("*_normal_var", "\\1", colnames(missnormal_rep_unif))
  colnames(backtransform_missnormal_rep) <- new_colnames

  # Combine the back-transformed columns with the remaining imputed columns
  impute_df_merged <- cbind(backtransform_missnormal_rep, impute_df_rest)

  return(impute_df_merged)
}
```

A.5.3 `TestNormalityofMissingCols` Function

This function finds columns with at least 1 missing value and evaluates whether it is normally distributed or not using Shapiro Wilks test at your specified p-value threshold.

To use this function, supply 2 arguments:

`dataset`: dataset with variables in columns and observations in rows

`p_value_threshold`: either 'any', 0.05, 0.01 or 0.001. When set to "any", the function will return the names of all columns with missing values, regardless of the p-value from the Shapiro-Wilks test.

The function returns a dataframe two columns: `Columns` contains the names of the columns with p-values below the specified threshold, and

`P_values` contains the corresponding p-values from the Shapiro-Wilk tests.

```
TestNormalityofMissingCols <- function(dataset, p_value_threshold = 0.001) {

  if(!(p_value_threshold %in% c(0.05, 0.01, 0.001, "any"))) {
    stop("p_value_threshold must be either 'any', 0.05, 0.01 or 0.001")
  }

  # Identify columns with missing values
  cols_with_na <- colnames(dataset)[colSums(is.na(dataset)) > 0]

  # List to store columns with p-values below the specified threshold
  cols_below_threshold <- vector()
  p_values_below_threshold <- vector()

  # Loop through each column with missing values
  for(col_name in cols_with_na) {

    # Exclude missing values
    col_values <- dataset[[col_name]][!is.na(dataset[[col_name]])]

    # Only calculate Shapiro-Wilk test for column with at least 3 unique values
    if(length(unique(col_values)) >= 3) {
      # Calculate the Shapiro-Wilk test
      shapiro_test <- shapiro.test(col_values)

      # If p-value is below the specified threshold, add column name
      # and p-value to lists
      if((shapiro_test$p.value < p_value_threshold) |
                                    (p_value_threshold == "any")) {
        cols_below_threshold <- c(cols_below_threshold, col_name)
        p_values_below_threshold <- c(p_values_below_threshold,
                                      shapiro_test$p.value)
      }
    }
  }

  output <- data.frame(Columns = cols_below_threshold,
                       P_values = p_values_below_threshold)
  # names(output) <- c("Columns", "P_values")
  return(output)
}
```

A.5.4 Helper Functions

```
# Not in from dplyr
`%!in%` = Negate(`%in%`)
```

```
extract_transformed_values <- function(input_list){

  transformed_df <- data.frame(ID = input_list[[1]][[2]][, 1])
  colnames(transformed_df) <- colnames(input_list[[1]][[2]])[1]

  for (i in seq_len(length(input_list))) {
    transformed_var <- input_list[[i]][[2]][, 3]
    colname <- colnames(input_list[[i]][[2]])[3]
    transformed_df[colname] <- transformed_var
  }

  return(transformed_df)
}

qfun_extract <- function(list_vals){

  qfunc_names <- vector()
  return_list <- list()
  res_1_list <- list()

  for(j in 1:length(list_vals)){
    # res_1 this has q function and transformed data for jth cpg
    res_1 <- list_vals[[j]]

    # this is just q function, which is first list element
    res_1_list[[j]] <- res_1[[1]]
    # get name of the qfunction
    qfunc_names[j] <- names(list_vals[[j]][1])

    # jth item is inverse function
    return_list[[j]] <- res_1_list[[j]]
  }
  # rename the list
  names(return_list) <- qfunc_names
  return(return_list)
}
```

A.6 `methyLImp` Imputation Results

SVD floating point operations per second (Flops) are $4n^2p + 8np^2 + 9p^3$. In the `methyLImp` algorithm, SVD is performed for every unique missing pattern. In our case, every CpG column and person had a unique missing pattern, so a new SVD would be performed for every column. `methyLImp` uses people as columns, so in the case of the FHS dataset, transposing the data would result in $n = 455,200$ and $p = 2,544$, which is $2.1x10^15$ flops for every single column needing imputation (2,544). Assume the imputation would be performed on a Mac M1 chip with 8 cores running at max 3.2 GHz (cycles per second) with 1.3 TeraFLOPS at double precision. That corresponds to a maximum theoretical throughput = (Number of cores) x (Clock speed) x (Operations per cycle) = 33.28 TeraFLOPS DP. Therefore, for every single column, we can estimate the most optimistic time needed in minutes for computation as $((flops/10^{12})/33.28)/60) = 1.07$ minutes. This means just over a minute would be needed in the optimal case for just the SVD process itself, which is 2,722 minutes for 1 dataset or about 1.9 days. However, this calculation assumes ideal conditions and maximum efficiency, which are usually achievable. Factors such

as power management, system load, thermal conditions, and software optimizations can impact the actual performance.

When we attempt to use `methyLImp` on the FHS dataset, we cannot run this within the maximum time or memory available on the cluster, and the program aborts after specifying 13 cores, 1 week run time, and 100 GB of memory. We attempt to circumvent these issues by subsetting the data and performing imputation on smaller dataframes, which successfully runs, but slowly. It averages to 25 minutes user time needed for every CpG with missing values. We restrict our analysis to 300 missing CpGs within each of the missing rates (1, 5, and 10%) for a total of 900 CpGs to be imputed in 1 replication. This resulted in $900 * 25/60/24 = 15.6$ days of run time. If this process was continued for all missing CpG values (13,656 of them), the program would require 238 days = 33.8 weeks for 1 replication. Because we only did one replication and 900 loci, results cannot be summarized based on probe properties or at the individual CpG locus level. Overall, the Untransformed approach has better imputation accuracy than the Missing Normal approach in the FHS dataset (Supplemental Figure **??**). As seen with Missing Normal with `imputePCA` in FHS, this method performs quite poorly. The median correlation is just above 0 in this transformed method, indicating almost no accuracy. This can be expected, however, because our transformation method performs transformations on the CpG loci whereas `methyLImp` performs imputation using observations. As such, our transformation would not guarantee the imputed values are being imputed in the transformed gaussian space.

A.7 Animal Groupings

Here we provide the exact labels used to classify animals into subgroups. We chose to keep the labels as R code to ensure other researchers could reproduce results when accessing the mammalian data in the Mammalian Consortium Repository.

```
primates <- c("Aye-aye", "Bamboo lemur", "Black lemur", "Blue-eyed black lemur",
         "Brown lemur", "Chimpanzee", "Collared brown lemur", "Crowned lemur",
         "Diademed sifaka", "Fat-tailed dwarf lemur", "Francois leaf monkey",
         "Golden-crowned sifaka", "Gorilla", "Gray mouse lemur", "Greater galago",
         "Marmoset", "Orangutan", "Potto(P.potto)", "Potto(P.coquereli)",
         "Rhesus macaque", "Ring-tailed lemur", "Sanford's brown lemur",
         "Slow loris", "Vervet", "White-fronted marmoset", "White-headed lemur",
         "Mongoose lemur", "Northern giant mouse lemur", "Red lemur",
         "Red ruffed lemur(V.rubra)", "Red ruffed lemur(V.variegata)",
         "Red-bellied lemur", "Olive baboon", "Orangutan",
         "Slender loris", "Pygmy slow loris", "South African galago")

mice <- c("African pygmy mouse", "Deer mouse",
         "Four-striped grass mouse", "Mouse")
```

```
rats <- c("African mole rat", "Black rat", "Blind mole rat", "Brown rat",
          "Bush tail rat", "Cape mole rat", "Cape-dune mole rat",
          "Damaraland mole rat", "Muskrat", "Namaqua dune mole-rat",
          "Namaqua rock rat", "Naked mole rat", "Pouched rat", "Rat")

bats <- c("Big brown bat", "Common vampire bat", "Egyptian fruit bat",
          "Fish-eating bat", "Greater horseshoe bat", "Greater mouse-eared bat",
          "Greater sac-winged bat", "Greater spear-nosed bat",
          "Grey-headed flying fox", "Halcyon horseshoe bat", "Indian fruit bat",
          "Jamaican fruit bat", "Large flying fox", "Lesser long-nosed bat",
          "Lesser short-nosed fruit bat", "Little brown bat",
          "Little golden-mantled flying fox", "Mexican free-tailed bat",
          "Noack's roundleaf bat", "Noctule", "Pallid bat", "Pale spear-nosed bat",
          "Pallas's mastiff bat", "Proboscis bat", "Seba's short-tailed bat",
          "Straw-colored fruit bat", "Variable flying fox", "Rodriguez flying fox")

fruit bats <- c("Egyptian fruit bat", "Grey-headed flying fox", "Indian fruit bat",
                "Jamaican fruit bat", "Large flying fox", "Lesser long-nosed bat",
                "Lesser short-nosed fruit bat", "Little golden-mantled flying fox",
                "Straw-colored fruit bat", "Variable flying fox",
                "Rodriguez flying fox")

microbats <- c("Big brown bat", "Fish-eating bat", "Greater horseshoe bat",
               "Greater mouse-eared bat", "Greater sac-winged bat",
               "Greater spear-nosed bat", "Halcyon horseshoe bat",
               "Little brown bat", "Mexican free-tailed bat",
               "Noack's roundleaf bat", "Noctule", "Pallid bat",
               "Pale spear-nosed bat", "Pallas's mastiff bat",
               "Proboscis bat", "Seba's short-tailed bat")

vampire bats <- c("Common vampire bat")

lemurs <- c("Aye-aye", "Bamboo lemur", "Black lemur", "Blue-eyed black lemur",
            "Brown lemur", "Collared brown lemur", "Crowned lemur",
            "Diademed sifaka", "Fat-tailed dwarf lemur", "Golden-crowned sifaka",
            "Gray mouse lemur", "Ring-tailed lemur", "Sanford's brown lemur",
            "White-headed lemur", "Mongoose lemur", "Northern giant mouse lemur",
            "Red lemur", "Red ruffed lemur(V.rubra)",
            "Red ruffed lemur(V.variegata)", "Red-bellied lemur")

monkeys <- c("Francois leaf monkey", "Rhesus macaque", "Vervet", "Olive baboon")

marmosets tamarins <- c("Marmoset", "White-fronted marmoset")

lorises galagos <- c("Slow loris", "Greater galago", "Slender loris",
                     "Pygmy slow loris", "South African galago", "Potto(P.potto)",
                     "Potto(P.coquereli)")
```

A.8 `EpigenTL` Functions

All of these functions can be found on Github with examples and more details on their use and function calls: https://github.com/kristenmcgreevy/EpigenTL. These functions are supplements to Section 4.3.2.

A.8.1 `Epigen.TL.Lasso` Function

`Epigen.TL.Lasso` performs Transfer Learning Lasso allowing either the Auxiliary Dataset Information to be known (Oracle or Oracle 1df) or estimated in the process (Estimate A0).

As necessary input, X_matrix is the matrix of covariates (nxp) concatenated between the Target and all Auxiliary Datasets, in that order. n should therefore be n_0 + n_1 + ... +

n_k. X_matrix needs to be a matrix and not a dataframe to allow for matrix multiplication to be carried out in the function. Y_vector is the outcome vector to develop the TL Lasso model to estimate. It should align with your X_matrix with Target and Auxiliary outcomes concatenated and be a nx1 vector. N_vector is a vector of number of observations in the Target and each Auxiliary datasets in the order they are concatenated. AuxInformation is either "Estimate A0", "Oracle", or "Oracle 1df" to signify the informative auxiliary datasets need estimated (EstA0) or they should be treated as known. If known, they can either be treated as equally informative and combined into 1 dataset (Oracle), or they can be treated individually (Oracle 1df). RhatCount is either "n0/3" or a value between 1, ..., p to specify how many marginal correlations to consider when calculating information. This should be set to NULL if AuxInformation is "Oracle" or "Oracle 1df". LambdaType is either "Constant" or "CV" to indicate if the lambda calculated from the target dataset should be used and adjusted based on the aux dataset size ("Constant") or whether the optimal lambda should be calculated via cross validation ("CV") for each set of auxiliary information. Default is "CV". seedstart is the seed to set in the calculation for reproducibility. If not specified, it is set to 123.

This function will return a data.frame with TL coefficients in the columns. The first column, "Variable" labels the intercept and columns from your X matrix that coefficient values correspond to. Columns 2:7 or 2:4 have the final coefficients with slight variations in their calculation. "min" and "1se" correspond to the lambda that either minimizes CV error or is at most 1se above it, respectively. "allcoef", "halfcoef", and "lambcoef" correspond to the coefficient thresholding used before combining coefficients across the auxiliary and target dataset. If "Constant" LambdaType was specified, only the minimum lambda from the target data is used and so "1se" coefficients are not presented.

```
Epigen.TL.Lasso <- function(X_matrix, Y_vector, N_vector,
                            AuxInformation, RhatCount = NULL, LambdaType = "CV",
                            seedstart = 123) {

  if(AuxInformation %!in% c("Estimate A0", "Oracle", "Oracle 1df")){
    stop("You must specify a method to capture auxiliary data informativeness.
         Options include 'Estimate A0', 'Oracle', or 'Oracle 1df'")
  }

  if(AuxInformation == "Estimate A0" & is.null(RhatCount)){
    stop("You must provide the number of Rhats to use when calculating
         auxiliary data informativeness.
         Options include any integer up to the column size of X_matrix OR 'n0/3'")
  }

  if(AuxInformation == "Estimate A0" & !is.null(RhatCount)){
    if(RhatCount > dim(X_matrix)[2]){
      stop("Rhat must be smaller than the number of columns in X.
         Please specify any integer up to the column size of X_matrix OR 'n0/3'")
    }
```

```
  if(RhatCount == 0){
    stop("Rhat must be a positive number from 1, ..., column size of X_matrix.")
  }
}

if(LambdaType %!in% c("CV", "Constant")){
  stop("LambdaType must be either 'CV' or 'Constant' to specify which
       lambda parameter is used for the auxiliary data.")
}

if(is.null(X_matrix)){
  stop("Please supply the X matrix")
}
if(is.null(Y_vector)){
  stop("Please supply the outcome")
}
if(is.null(N_vector)){
  stop("Please supply the N vector specifying the number of observations
       in target and auxiliary datasets")
}

if(sum(N_vector) != dim(X_matrix)[1]){
  stop("N_vector and X_matrix do not have the same number of observations")
}

if(sum(N_vector) != length(Y_vector)){
  stop("N_vector and Y do not have the same number of observations")
}

if(!is.null(RhatCount)){
  if(RhatCount == 'n0/3'){
      # set before calling the other functions.
      RhatCount <- round(N_vector[1] / 3)
  }
}


  # TransLasso for Estimating informative set
  if(AuxInformation == "Estimate A0"){
    # for reproducibility
    set.seed(seedstart)

    Translasso_output <- TransLasso.EstA0(X = X_matrix, y = Y_vector,
                          n.vec = N_vector, AuxInformation = AuxInformation,
                          LambdaType = LambdaType, RhatCount = RhatCount)
  }
  # TransLasso for Oracle or Oracle 1df
  if(AuxInformation != "Estimate A0"){
    # for reproducibility
    set.seed(seedstart)

    Translasso_output <- TransLasso.Oracles(X = X_matrix, y = Y_vector,
                          n.vec = N_vector, AuxInformation = AuxInformation,
                          LambdaType = LambdaType)
  }

  # calculate intercepts
  intercept_beta_min <- mean(Y_vector - X_matrix %*%
                               Translasso_output$beta.hat_min,
                             na.rm=TRUE)
  intercept_beta_1se <- mean(Y_vector - X_matrix %*%
                               Translasso_output$beta.hat_1se,
                             na.rm=TRUE)

  intercept_beta_min_lambda <- mean(Y_vector -
                       X_matrix %*% Translasso_output$beta.hat_min_lambda,
                                    na.rm=TRUE)
  intercept_beta_1se_lambda <- mean(Y_vector -
                       X_matrix %*% Translasso_output$beta.hat_1se_lambda,
                                    na.rm=TRUE)
```

```
    intercept_beta_min_halflambda <- mean(Y_vector -
                        X_matrix %*% Translasso_output$beta.hat_min_halflambda,
                                    na.rm=TRUE)
    intercept_beta_1se_halflambda <- mean(Y_vector -
                        X_matrix %*% Translasso_output$beta.hat_1se_halflambda,
                                    na.rm=TRUE)

    # keep all coefficients.
    TL_beta_coef <- data.frame(Variable = c("Intercept", colnames(X_matrix)),
                        beta_min_allcoef = c(intercept_beta_min,
                        Translasso_output$beta.hat_min),
                        beta_1se_allcoef = c(intercept_beta_1se,
                        Translasso_output$beta.hat_1se),
                        beta_min_halfcoef = c(intercept_beta_min_halflambda,
                        Translasso_output$beta.hat_min_halflambda),
                        beta_1se_halfcoef = c(intercept_beta_1se_halflambda,
                        Translasso_output$beta.hat_1se_halflambda),
                        beta_min_lambcoef = c(intercept_beta_min_lambda,
                        Translasso_output$beta.hat_min_lambda),
                        beta_1se_lambcoef = c(intercept_beta_1se_lambda,
                        Translasso_output$beta.hat_1se_lambda))

    # if Constant, only the min lambda is used.
    if(LambdaType == "Constant"){
          TL_beta_coef <- data.frame(Variable = c("Intercept", colnames(X_matrix)),
                              beta_min_allcoef = c(intercept_beta_min,
                                 Translasso_output$beta.hat_min),
                              beta_min_halfcoef = c(intercept_beta_min_halflambda,
                                 Translasso_output$beta.hat_min_halflambda),
                              beta_min_lambcoef = c(intercept_beta_min_lambda,
                                 Translasso_output$beta.hat_min_lambda))
    }

  # return Final TL coefficients as a dataframe
  return(TL_beta_coef)

}
```

A.8.2 `Saliva.2.Blood.DNAmBiomarkers` Function

`Saliva.2.Blood.DNAmBiomarkers` functions calculates the blood DNAm Biomarkers from methylation values using either the C or C+S method algorithms.

As input, X is the matrix or dataframe of saliva DNA methylation beta values with samples in rows and methylation sites in columns. To see the list of CpG loci needed for computation, please call colnames to CS_Algorithms_GitHub or C_Algorithms_GitHub (which are loaded in the global environment). method should be either "C+S" or "C" to indicate which set of algorithms you are interested in calculating. If "C+S", you must also provide SalivaDNAmBiom. SalivaDNAmBiom should be the matrix or dataframe of Saliva DNAm biomarkers (ie DNAm biomarkers directly calculated with saliva methylation values) if using the C+S algorithms. Otherwise, this should be NULL. Default is NULL.

This function outputs a dataframe with predicted DNAm Biomarkers in each column with rows in the same order as the originally supplied X.

```
Saliva.2.Blood.DNAmBiomarkers <- function(X, method, SalivaDNAmBiom = NULL) {
```

```
# make sure ppl provide the saliva DNAm biomarkers if its C+S
if (method == "C+S" && is.null(SalivaDNAmBiom)) {
  stop("For method C+S, saliva DNAm Biomarker matrix must be provided.")
}


if (method == "C") {
  TLcoeffMatrix <- C_Algorithms_GitHub
  Varstart <- 2
} else if (method == "C+S") {
  TLcoeffMatrix <- CS_Algorithms_GitHub
  Varstart <- 3
} else {
  stop("Invalid method specified. Please specify either 'C' or 'C+S'")
}


cpgs_needed <- TLcoeffMatrix$Variable[Varstart:length(TLcoeffMatrix$Variable)]
if(sum(cpgs_needed %!in% colnames(X)) > 0){
  cpgs_missing <<- cpgs_needed[cpgs_needed %!in% colnames(X)]
  stop("Not all CpG columns are present in X for this prediction.
       Please see cpgs_missing for a list of missing but necessary CpGs.")
}


X_keep <- X[, cpgs_needed]
if (any(is.na(X_keep))) {
  stop("Missing values are not allowed.
         Please impute missing values in the X matrix.")
}

### Now can start the computations ###

new_biomarker_list <- colnames(TLcoeffMatrix)[-1]

n_biom <- dim(TLcoeffMatrix)[2]
n_var <- dim(TLcoeffMatrix)[1]

# make temp dataset with intercept, saliva, and columns in correct order
if (method == "C"){
  newdata_temp <- data.frame(intercept = 1,
                             X[, TLcoeffMatrix$Variable[Varstart:n_var]])
}else{
  newdata_temp <- data.frame(intercept = 1, SalivaDNAmBiom = NA,
                             X[, TLcoeffMatrix$Variable[Varstart:n_var]])
}

# Turn Coefficient matrix into a real matrix for multiplying
TLcoeffMatrix2 <- matrix(unlist(TLcoeffMatrix[, 2:n_biom]),
                         ncol = (n_biom-1), nrow = n_var)

# initialize predictions dataframe
pred_df <- data.frame(rep(NA, dim(X)[1]))
k <- 1

for (i in new_biomarker_list) {

  if(method == "C+S"){
    # set Saliva value to the biomarker of interest
    newdata_temp$SalivaDNAmBiom <- SalivaDNAmBiom[, i]
  }

  # get the column of the TL coef matrix the biomarker is in
  TL_coef_Col <- (which(colnames(TLcoeffMatrix) == i) - 1)

  # turn into matrix for multiplication
  newdata_temp2 <- matrix(unlist(newdata_temp), ncol = dim(newdata_temp)[2],
                          nrow = dim(newdata_temp)[1])

  # data has ppl in rows, variables in columns,
  # TLcoef matrix has coefficients for each biomarker in a column
  cur_pred <- newdata_temp2 %*% TLcoeffMatrix2[, TL_coef_Col]
```

```
    # set prediction to the dataframe
    pred_df[, k] <- c(cur_pred)
    colnames(pred_df)[k] <- i

    # make k go up
    k <- k + 1

    # remove values so we don't repeat by accident
    rm(cur_pred); rm(newdata_temp2)

    if(method == "C+S"){
      # set Saliva value to NA so we don't repeat by accident
      newdata_temp$SalivaDNAmBiom <- NA
    }

  } # end of for loop in biomarker list

  # return the predicted DNAm values.
  if(method == "C+S"){
    colnames(pred_df) <- paste0(colnames(pred_df), "_CS_Pred")
  } else{
    colnames(pred_df) <- paste0(colnames(pred_df), "_C_Pred")
  }

  return(pred_df)
}
```

A.8.3 TL_Lasso Function

TL_Lasso is the actual Transfer Learning Lasso loop process called inside both TL.Lasso.EstA0 and TL.Lasso.Oracles. As input, X is the matrix of covariates, y is the outcome vector, A0 is the informative Aux Set. It is either NULL or estimated. n.vec is a vector of number of observations in the target and aux datasets in that order, lam.const is whether we are calculating the optimal lambda via CV in each informative aux set or the constant value if using the methods outlined in TransLasso paper.

It outputs the estimated beta coefficients with and without thresholding (when performed in auxiliary data) and the estimated lambda values.

```
TL_Lasso <- function(X, y, A0, n.vec, lam.const=NULL, lam.const_1se = NULL, ...){

  p <- ncol(X)
  size.A0 <- length(A0) # set to NULL so its 0

  if(size.A0 > 0){ # only for Aux data, otherwise SKIP to below

    ind.kA <- ind.set(n.vec, c(1, A0+1))
    ind.1 <- 1:n.vec[1] # vector of all values to build initial model.

    y.A <- y[ind.kA]

    # if null, CV done for each Informative Set and both min and 1se lambda kept.
    if(is.null(lam.const)){
    # this gets run on first run because we have it set to NULL
      # does its own grid search for lambda
      cv.init<-cv.glmnet(X[ind.kA,], y.A, nfolds=8)
      # now, it will just take whatever the best value was and calculate the constant.
      lam.const <- cv.init$lambda.min/sqrt(2*log(p)/length(ind.kA))
      lam.const_1se <- cv.init$lambda.1se/sqrt(2*log(p)/length(ind.kA))
    }
```

```
  if(!is.null(lam.const) & is.null(lam.const_1se)){
    lam.const_1se <- lam.const
  }

  # w.kA = coefficients from Xk predicts Yk
  w.kA_min <- as.numeric(glmnet(X[ind.kA,], y.A,
                  lambda=lam.const*sqrt(2*log(p)/length(ind.kA)))$beta)
  w.kA_1se <- as.numeric(glmnet(X[ind.kA,], y.A,
                  lambda=lam.const_1se*sqrt(2*log(p)/length(ind.kA)))$beta)

  # w.k coefficient thresholding
  w.kA_min_halflambda <- w.kA_min*(abs(w.kA_min) >=
                           0.5*lam.const*sqrt(2*log(p)/length(ind.kA)))
  w.kA_min_lambda <- w.kA_min*(abs(w.kA_min) >=
                            lam.const*sqrt(2*log(p)/length(ind.kA)))

  w.kA_1se_halflambda <- w.kA_1se*(abs(w.kA_1se) >=
                           0.5*lam.const_1se*sqrt(2*log(p)/length(ind.kA)))
  w.kA_1se_lambda <- w.kA_1se*(abs(w.kA_1se) >=
                           lam.const_1se*sqrt(2*log(p)/length(ind.kA)))

  # build model in target where outcome is what is left after
  # taking the yhat from kth model coefficients.
  # delta.kA = coefficients from X0 predicts (Y0 - Y0hat from w.kA)
  delta.kA_min <- as.numeric(glmnet(x=X[ind.1,],y=y[ind.1]-X[ind.1,]%*%w.kA_min,
                       lambda=lam.const*sqrt(2*log(p)/length(ind.1)))$beta)
  delta.kA_1se <- as.numeric(glmnet(x=X[ind.1,],y=y[ind.1]-X[ind.1,]%*%w.kA_1se,
                      lambda=lam.const_1se*sqrt(2*log(p)/length(ind.1)))$beta)

  # delta.k coefficient thresholding
  delta.kA_min_halflambda <- delta.kA_min*(abs(delta.kA_min) >=
                               0.5*lam.const*sqrt(2*log(p)/length(ind.1)))
  delta.kA_min_lambda <- delta.kA_min*(abs(delta.kA_min) >=
                               lam.const*sqrt(2*log(p)/length(ind.1)))

  delta.kA_1se_halflambda <- delta.kA_1se*(abs(delta.kA_1se) >=
                               0.5*lam.const_1se*sqrt(2*log(p)/length(ind.1)))
  delta.kA_1se_lambda <- delta.kA_1se*(abs(delta.kA_1se) >=
                               lam.const_1se*sqrt(2*log(p)/length(ind.1)))

  # final beta coefficients (the kth lasso coefficients are the weights
  # from kth lasso model + kth lasso on target)

  # no thresholding
  beta.kA_min <- w.kA_min + delta.kA_min
  beta.kA_1se <- w.kA_1se + delta.kA_1se

  # half lambda thresholding
  beta.kA_min_halflambda <- w.kA_min_halflambda + delta.kA_min_halflambda
  beta.kA_min_lambda <- w.kA_min_lambda + delta.kA_min_lambda

  # lambda thresholding
  beta.kA_1se_halflambda <- w.kA_1se_halflambda + delta.kA_1se_halflambda
  beta.kA_1se_lambda <- w.kA_1se_lambda + delta.kA_1se_lambda

  lam.const=NULL # reset lambda because we don't want it to loop through as !null
  # first auxillary dataset because we supply the lambda constant.

  # output all coefficients
  # recall if constant lambda was specified, min and 1se will be identical.
  list(beta.kA_min = as.numeric(beta.kA_min), w.kA_min=w.kA_min,
     beta.kA_1se = as.numeric(beta.kA_1se), w.kA_1se=w.kA_1se,
     beta.kA_min_halflambda = as.numeric(beta.kA_min_halflambda),
          w.kA_min_halflambda=w.kA_min_halflambda,
     beta.kA_1se_halflambda = as.numeric(beta.kA_1se_halflambda),
          w.kA_1se_halflambda=w.kA_1se_halflambda,
     beta.kA_min_lambda = as.numeric(beta.kA_min_lambda),
          w.kA_min_lambda=w.kA_min_lambda,
     beta.kA_1se_lambda = as.numeric(beta.kA_1se_lambda),
          w.kA_1se_lambda=w.kA_1se_lambda,
     lam.const=lam.const, lam.const_1se=lam.const_1se)
}else{ # end of if(size.A0 > 0)
```

```
    # BEGINNING CODE FOR INITIAL / TARGET DATA
    cv.init <- cv.glmnet(X[1:n.vec[1],], y[1:n.vec[1]], nfolds=8)

    # When constant lambda selected, min lambda is used.
    lam.const <- cv.init$lambda.min/sqrt(2*log(p)/n.vec[1])
    lam.const_1se <- cv.init$lambda.1se/sqrt(2*log(p)/n.vec[1])

    # extract coefficients (excluding the intercept)
    beta.kA_min <- predict(cv.init, s='lambda.min', type='coefficients')[-1]
    beta.kA_1se <- predict(cv.init, s='lambda.1se', type='coefficients')[-1]
    w.kA_min <- w.kA_1se <- NA

    list(beta.kA_min = as.numeric(beta.kA_min), w.kA_min=w.kA_min,
       beta.kA_1se = as.numeric(beta.kA_1se), w.kA_1se=w.kA_1se,
       lam.const=lam.const, lam.const_1se = lam.const_1se)
  }

}
```

A.8.4 `TransLasso.Oracles` Function

`TransLasso.Oracles` performs Oracle Transfer Learning Lasso meaning the set of auxiliary datasets is specified to either run all at once (Oracle) or 1 dataset at a time (Oracle 1df). As input, X is the matrix of covariates, y is the outcome vector, n.vec is a vector of number of observations in the target and aux datasets in that order, AuxInformation is either "Oracle" or "Oracle 1df", LambdaType is either "Constant" or "CV" to indicate if the lambda calculated from the target dataset should be used and adjusted based on the aux dataset size ("Constant") or whether the optimal lambda should be calculated via cross validation ("CV") for each set of auxiliary information.

It outputs the aggregated coefficients at various parameterizations and the weights used when aggregating.

```
TransLasso.Oracles <- function(X, y, n.vec, AuxInformation = "Oracle 1df",
                               LambdaType = "CV", ...) {

  M = length(n.vec)-1
  p <- ncol(X)

  # row indices of where these observations are for target
  ind.1 <- ind.set(n.vec, 1)

  Tset <- list()

  if(AuxInformation == "Oracle"){
      # make Tset actually just all the aux datasets (for oracle all at once)
      # aux datasets start at 2.
      Tset[[1]] <- c(1:M)
  }
  # take 1 aux dataset at a time, noting that we index by dataset before.
  if(AuxInformation == "Oracle 1df"){
    for(kk in 1:M){ #use Rhat as the selection rule
     Tset[[kk]] <- kk
    } # the sets of aux datasets to take for ranking of datasets.
  }

  k0 = length(Tset)
```

```
Tset <- unique(Tset)

beta.T_min <- beta.T_min_lambda <- beta.T_min_halflambda <- list()
beta.T_1se <- beta.T_1se_lambda <- beta.T_1se_halflambda <- list()

# Lasso on Target Data only
init.re <- TL_Lasso(X=X, y=y, A0=NULL, n.vec=n.vec)

beta.T_min[[1]] <- init.re$beta.kA_min
beta.T_1se[[1]] <- init.re$beta.kA_1se

# if constant lambda specified, it is here.
c1_lambda_const <- init.re$lam.const
c1_lambda_const_1se <- init.re$lam.const_1se

og_lasso_coef_min <- init.re$beta.kA_min
og_lasso_coef_1se <- init.re$beta.kA_1se


beta.T_min_lambda <- beta.T_min_halflambda <- beta.T_min
beta.T_1se_lambda <- beta.T_1se_halflambda <- beta.T_1se

# go through TL Lasso for each informative set
for(kk in 1:length(Tset)){
  T.k <- Tset[[kk]]

  # changed function call to lam.const = NULL to do Aux CV
  if(LambdaType == "CV"){
    re.k <- TL_Lasso(X=X, y=y, A0=T.k, n.vec=n.vec, lam.const = NULL)
  }else{ # constant lambda specification
    re.k <- TL_Lasso(X=X, y=y, A0=T.k, n.vec=n.vec, lam.const = c1_lambda_const,
                     lam.const_1se = c1_lambda_const_1se)
  }

  # extract coefficients for each informative auxiliary set
  beta.T_min[[kk+1]] <- re.k$beta.kA_min
  beta.pool.T_min[[kk+1]] <- re.k$w.kA_min

  beta.T_1se[[kk+1]] <- re.k$beta.kA_1se
  beta.pool.T_1se[[kk+1]] <- re.k$w.kA_1se

  beta.T_min_halflambda[[kk+1]] <- re.k$beta.kA_min_halflambda
  beta.pool.T_min_halflambda[[kk+1]] <- re.k$w.kA_min_halflambda

  beta.T_1se_lambda[[kk+1]] <- re.k$beta.kA_1se_lambda
  beta.pool.T_1se_lambda[[kk+1]] <- re.k$w.kA_1se_lambda
}

beta.T_min <- beta.T_min[!duplicated((beta.T_min))]
beta.T_min <- as.matrix(as.data.frame(beta.T_min))

beta.T_1se <- beta.T_1se[!duplicated((beta.T_1se))]
beta.T_1se <- as.matrix(as.data.frame(beta.T_1se))

beta.T_min_halflambda <- beta.T_min_halflambda[
                          !duplicated((beta.T_min_halflambda))]
beta.T_min_halflambda <- as.matrix(as.data.frame(beta.T_min_halflambda))
beta.T_min_lambda <- beta.T_min_lambda[!duplicated((beta.T_min_lambda))]
beta.T_min_lambda <- as.matrix(as.data.frame(beta.T_min_lambda))

beta.T_1se_halflambda <- beta.T_1se_halflambda[
                               !duplicated((beta.T_1se_halflambda))]
beta.T_1se_halflambda <- as.matrix(as.data.frame(beta.T_1se_halflambda))
beta.T_1se_lambda <- beta.T_1se_lambda[!duplicated((beta.T_1se_lambda))]
beta.T_1se_lambda <- as.matrix(as.data.frame(beta.T_1se_lambda))

## aggregate coefficients using squared error.
## aggregate w.kA and delta.kA
agg.re1_min <- coef.aggr(B= beta.T_min, X = X, y = y, N_vector = n.vec)
agg.re1_1se <- coef.aggr(B= beta.T_1se, X = X, y = y, N_vector = n.vec)

agg.re1_min_halflambda <- coef.aggr(B= beta.T_min_halflambda, X = X, y = y,
```

```
                                        N_vector = n.vec)
  agg.re1_1se_halflambda <- coef.aggr(B= beta.T_1se_halflambda, X = X, y = y,
                                        N_vector = n.vec)

  agg.re1_min_lambda <- coef.aggr(B= beta.T_min_lambda, X = X,
                            y = y, N_vector = n.vec)
  agg.re1_1se_lambda <- coef.aggr(B= beta.T_1se_lambda, X = X,
                            y = y, N_vector = n.vec)


  return(list(beta.hat_min = agg.re1_min$beta, theta.hat_min = agg.re1_min$theta,
              beta.hat_1se = agg.re1_1se$beta, theta.hat_1se = agg.re1_1se$theta,
              beta.hat_min_halflambda = agg.re1_min_halflambda$beta,
              theta.hat_min_halflambda = agg.re1_min_halflambda$theta,
              beta.hat_1se_halflambda = agg.re1_1se_halflambda$beta,
              theta.hat_1se_halflambda = agg.re1_1se_halflambda$theta,
              beta.hat_min_lambda = agg.re1_min_lambda$beta,
              theta.hat_min_lambda = agg.re1_min_lambda$theta,
              beta.hat_1se_lambda = agg.re1_1se_lambda$beta,
              theta.hat_1se_lambda = agg.re1_1se_lambda$theta,
              c1_lambda_const = c1_lambda_const))
}
```

A.8.5 `TransLasso.EstA0` Function

`TransLasso.EstA0` performs Transfer Learning Lasso while estimating the Informative Auxiliary Data. X is the matrix of covariates (nxp), y is the outcome vector (nx1), n.vec is a vector of number of observations in the target and aux datasets in that order, RhatCount is either "n0/3" or a value between 1, ..., p to specify how many marginal correlations to consider when calculating information. LambdaType is either "Constant" or "CV" to indicate if the lambda calculated from the target dataset should be used and adjusted based on the aux dataset size ("Constant") or whether the optimal lambda should be calculated via cross validation ("CV") for each set of auxiliary information.

It outputs the aggregated coefficients at various parameterizations and the weights used when aggregating.

```
TransLasso.EstA0 <- function(X, y, n.vec, RhatCount, LambdaType, ...){

  # count of aux datasets
  M = length(n.vec)-1

  Rhat <- rep(0, M+1)
  p <- ncol(X)

  # make row indices of where target observations are
  ind.1 <- ind.set(n.vec, 1)

  # calculate informativeness for each aux study
  for(k in 2: (M+1)){
    ind.k <- ind.set(n.vec, k) # row indices for kth aux sample.

    # calculate difference in marginal correlations between k aux and target data.
    Xty.k <- t(X[ind.k, ])%*%y[ind.k] / n.vec[k] - t(X[ind.1,])%*%y[ind.1]/ n.vec[1]

    # Rhat adjusts how many marginal correlations are looked at
    # take the top largest correlation differences based on RhatCount
```

```
  margin.T <- sort(abs(Xty.k), decreasing=T)[1:RhatCount]

  # estimated sparse index for kth aux sample.
  Rhat[k] <- sum(margin.T^2)
}

Tset <- list()
k0 = 0
# get ordering of smallest to largest Rhat for aux samples.
kk.list <- unique(rank(Rhat[-1]))

for(kk in 1:length(kk.list)){#use Rhat as the selection rule
  Tset[[k0+kk]] <- which(rank(Rhat[-1]) <= kk.list[kk])
} # the sets of aux datasets to take for each ranking of datasets to include.

k0 = length(Tset)
Tset <- unique(Tset)

beta.T_min <- beta.T_min_lambda <- beta.T_min_halflambda <- list()
beta.T_1se <- beta.T_1se_lambda <- beta.T_1se_halflambda <- list()

# Lasso on Target Data only
init.re <- TL_Lasso(X=X, y=y, A0=NULL, n.vec=n.vec)

beta.T_min[[1]] <- init.re$beta.kA_min
beta.T_1se[[1]] <- init.re$beta.kA_1se

# if constant lambda specified, it is here.
c1_lambda_const <- init.re$lam.const
c1_lambda_const_1se <- init.re$lam.const_1se

og_lasso_coef_min <- init.re$beta.kA_min
og_lasso_coef_1se <- init.re$beta.kA_1se

beta.T_min_lambda <- beta.T_min_halflambda <- beta.T_min
beta.T_1se_lambda <- beta.T_1se_halflambda <- beta.T_1se

# go through TL Lasso for each informative set
for(kk in 1:length(Tset)){
  T.k <- Tset[[kk]]

  # which lambda type changes which section of function call it goes into
  if(LambdaType == "CV"){
    re.k <- TL_Lasso(X=X, y=y, A0=T.k, n.vec=n.vec, lam.const = NULL)
  }
  if(LambdaType == "Constant"){
    re.k <- TL_Lasso(X=X, y=y, A0=T.k, n.vec=n.vec,
                         lam.const = c1_lambda_const,
                         lam.const_1se = c1_lambda_const_1se)
  }

  # extract coefficients for each informative auxiliary set
  beta.T_min[[kk+1]] <- re.k$beta.kA_min
  beta.T_1se[[kk+1]] <- re.k$beta.kA_1se
  beta.T_min_halflambda[[kk+1]] <- re.k$beta.kA_min_halflambda
  beta.T_min_lambda[[kk+1]] <- re.k$beta.kA_min_lambda
  beta.T_1se_lambda[[kk+1]] <- re.k$beta.kA_1se_lambda
  beta.T_1se_halflambda[[kk+1]] <- re.k$beta.kA_1se_halflambda
}

beta.T_min <- beta.T_min[!duplicated((beta.T_min))]
beta.T_min <- as.matrix(as.data.frame(beta.T_min))

beta.T_1se <- beta.T_1se[!duplicated((beta.T_1se))]
beta.T_1se <- as.matrix(as.data.frame(beta.T_1se))

beta.T_min_halflambda <- beta.T_min_halflambda[
                          !duplicated((beta.T_min_halflambda))]
beta.T_min_halflambda <- as.matrix(as.data.frame(beta.T_min_halflambda))
beta.T_min_lambda <- beta.T_min_lambda[!duplicated((beta.T_min_lambda))]
beta.T_min_lambda <- as.matrix(as.data.frame(beta.T_min_lambda))
```

```
  beta.T_1se_halflambda <- beta.T_1se_halflambda[
                        !duplicated((beta.T_1se_halflambda))]
  beta.T_1se_halflambda <- as.matrix(as.data.frame(beta.T_1se_halflambda))
  beta.T_1se_lambda <- beta.T_1se_lambda[!duplicated((beta.T_1se_lambda))]
  beta.T_1se_lambda <- as.matrix(as.data.frame(beta.T_1se_lambda))

  ## aggregate coefficients using squared error.
  # No coef thresholding
  agg.re1_min <- coef.aggr(B= beta.T_min, X = X, y = y, N_vector = n.vec)
  agg.re1_1se <- coef.aggr(B= beta.T_1se, X = X, y = y, N_vector = n.vec)

  # half lambda coef thresholding
  agg.re1_min_halflambda <- coef.aggr(B= beta.T_min_halflambda, X = X, y = y,
                                      N_vector = n.vec)
  agg.re1_1se_halflambda <- coef.aggr(B= beta.T_1se_halflambda, X = X, y = y,
                                      N_vector = n.vec)

  # lambda coef thresholding
  agg.re1_min_lambda <- coef.aggr(B= beta.T_min_lambda, X = X,
                    y = y, N_vector = n.vec)
  agg.re1_1se_lambda <- coef.aggr(B= beta.T_1se_lambda, X = X,
                    y = y, N_vector = n.vec)


  # theta are the returned weights of each dataset.
  # betas are the final, aggregated coefficients for potential covariates
  return(list(beta.hat_min = agg.re1_min$beta, theta.hat_min = agg.re1_min$theta,
              beta.hat_1se = agg.re1_1se$beta, theta.hat_1se = agg.re1_1se$theta,
              beta.hat_min_halflambda = agg.re1_min_halflambda$beta,
              theta.hat_min_halflambda = agg.re1_min_halflambda$theta,
              beta.hat_1se_halflambda = agg.re1_1se_halflambda$beta,
              theta.hat_1se_halflambda = agg.re1_1se_halflambda$theta,
              beta.hat_min_lambda = agg.re1_min_lambda$beta,
              theta.hat_min_lambda = agg.re1_min_lambda$theta,
              beta.hat_1se_lambda = agg.re1_1se_lambda$beta,
              theta.hat_1se_lambda = agg.re1_1se_lambda$theta,
              c1_lambda_const = c1_lambda_const,
              og_lasso_coef_min = og_lasso_coef_min,
              og_lasso_coef_1se = og_lasso_coef_1se))
}
```

A.8.6 Helper Functions

`coef.aggr` function aggregates the coefficients from target and auxiliary data using the error in the target data. This function is called in the background; it is not being called directly by users.

As input, this function requires B, the coefficient vector, X, the matrix of covariates, y, the outcome vector, and N_vector, a vector of number of observations in the target and aux datasets in that order. It outputs a two item list, with the first item, theta, being the weights used to aggregate the coefficients, and beta, the final, aggregated coefficients.

```
coef.aggr <- function(B, X, y, N_vector){

  # if all coefficients are zero, just return zero
  if(sum(B == 0) == ncol(B)*nrow(B)){
    return(rep(0,nrow(B)))
  }
  p <- nrow(B)
  K <- ncol(B)
  colnames(B) <- NULL

    # Take the difference in target y and predicted y
```

```
    # from each aux data coefficients
    y0hatk <- -log(colSums(y[1:N_vector[1]] - X[1:N_vector[1], ] %*% B)^2)
    theta.hat <- exp(y0hatk)
    theta.hat = theta.hat / sum(theta.hat)
    # weights by fraction of total squared error

    # multiply betak by weights for each kth aux
    beta <- as.numeric(B%*%theta.hat)

  list(theta = theta.hat, beta = beta)

}
```

`ind.set` tells the TL functions where the first and last observation is for each dataset (target, aux 1, ..., aux k) based on the sample sizes in n.vec. n.vec is the vector of number of observations in the target and aux datasets in that order. k is index of the dataset we are interested in extracting values from. It returns the indices in the X matrix and y vector to extract.

```
ind.set <- function(n.vec, k.vec){
  ind.re <- NULL
  for(k in k.vec){
    if(k==1){
      ind.re<-c(ind.re,1: n.vec[1])
    }else{
      ind.re<- c(ind.re, (sum(n.vec[1:(k-1)])+1): sum(n.vec[1:k]))
    }
  }
  ind.re
}
```

Appendix B

B.1 Supplemental Table and Figures

B.1.1 DNAm Fitness Biomarkers and DNAmFitAge

Supplemental Figure B.1 accompanies Section 2.3.2 by providing additional results relating the DNAm fitness biomarkers to aging phenotypes in validation datasets.

Supplemental Table B.1 accompanies Section 2.3 by providing additional performance metrics beyond correlation and stratified by sex.

Supplemental Table B.2 and B.3 accompanies Section 2.3.3 and Section A.3 to evaluate whether the effects observed in younger FitAges in body builders can be explained by their supplement usage.

Supplemental Table B.4 accompanies Section 2.3 by providing hazard ratios and coefficient values for phenotypic outcomes as well as meta-analysis results across the validation datasets.

Table B.1: DNAmFitAge Performance in Validation Datasets

		Females	Males	Male Model in Females	Female Model in Males
Training Data	Median Absolute Deviation	2.7	3.0	11.9	13.5
	Mean Deviation	0.0	0.0	-12.2	13.1
	R	0.923	0.925	0.925	0.922
LBC1921	Median Absolute Deviation	3.7	4.8	11.0	14.5
	Mean Deviation	0.8	1.1	-11.1	13.8
	R	0.409	0.386	0.404	0.391
LBC1936	Median Absolute Deviation	3.2	3.4	11.6	13.3
	Mean Deviation	0.0	0.2	-11.9	12.9
	R	0.635	0.635	0.647	0.624
CALERIE	Median Absolute Deviation	4.9	2.3	17.1	11.0
	Mean Deviation	-5.0	-2.0	-17.1	11.0
	R	0.926	0.915	0.928	0.912
InChianti	Median Absolute Deviation	3.9	3.9	16.0	9.6
	Mean Deviation	-3.8	-4.3	-16.1	9.1
	R	0.969	0.964	0.969	0.963
JHS	Median Absolute Deviation	2.9	3.4	13.6	9.2
	Mean Deviation	-1.6	-2.8	-13.9	8.6
	R	0.937	0.917	0.940	0.914
WHI	Median Absolute Deviation	3.8		16.8	
	Mean Deviation	-3.4		-16.8	
	R	0.808		0.812	

Table B.2: Linear models evaluating supplement usage to DNAmFitAge and DNAmVO2max after adjusting for age

Supplement in Model		Outcome: DNAmFitAge		Outcome: DNAmVO2max	
		Supplement	BodyBuilder	Supplement	BodyBuilder
Multivitamins	coefficient	-0.32	-0.62	0.68	0.07
	p-value	0.690	0.208	0.041	0.746
Proteins	coefficient	-0.05	-0.65	0.45	0.10
	p-value	0.961	0.184	0.241	0.607
Energy	coefficient	0.16	-0.66	0.24	0.13
	p-value	0.852	0.175	0.518	0.513
Magnesium	coefficient	-1.03	-0.60	-0.12	0.15
	p-value	0.213	0.219	0.727	0.472
Vitamin D	coefficient	-0.56	-0.62	-0.32	0.16
	p-value	0.570	0.207	0.439	0.431
Omega-3	coefficient	-1.23	-0.46	0.33	0.08
	p-value	0.157	0.366	0.355	0.687

Table B.3: Dietary Supplement Use by Male Athlete Status

		Control	Body Builder	Fisher's Exact p-value
Multivitamins	No	141	55	0.016
	Yes	8	11	
Proteins	No	140	58	0.169
	Yes	9	8	
Energy	No	145	53	6.81E-05
	Yes	4	13	
Magnesium	No	140	59	0.265
	Yes	9	7	
Vitamin D	No	143	58	0.036
	Yes	6	8	
Omega-3	No	144	59	0.050
	Yes	5	7	

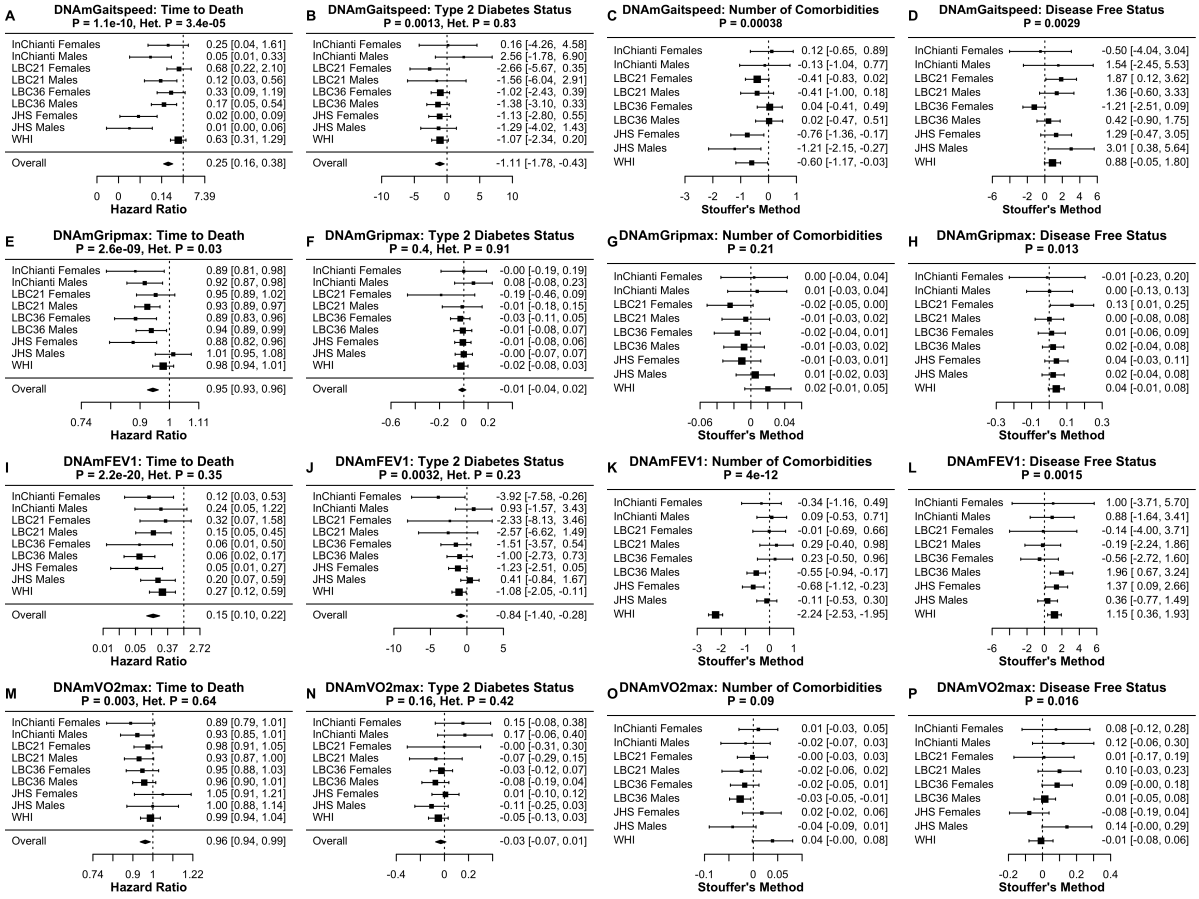


Figure B.1: Meta-analysis forest plots for DNAm fitness parameter biomarkers with age-related conditions. Each row reports a meta analysis forest plot for combining hazard ratios or regression coefficients across dataset cohorts for one DNAm biomarker estimate. (A-D) DNAmGaitSpeed without age, (E-H) DNAmGripmax without age, (I-L) DNAmFEV1, and (M-P) DNAmVO2max. Time-to-death, type 2 diabetes, comorbidity count, and disease-free status are presented. Meta-analysis p-values are displayed in the header of each panel, and test of heterogeneity Cochran Q test p-value (Het. P) are displayed for fixed effect models. Fixed effects models were used for time-to-death and type 2 diabetes whereas Stouffer's method was used for comorbidity count and disease-free status. All DNAm fitness biomarkers are predictive of mortality, and DNAmGaitspeed and DNAmFEV1 are predictive of number of comorbidities.

Supplemental Table B.5 accompanies Section 2.3.2 by providing additional results relating the DNAm fitness biomarkers to aging phenotypes in validation datasets.

Table B.4: FitAgeAcceleration Association to Phenotypic Outcomes

		Time to Death*	Time to Coronary Heart Disease*	Type 2 Diabetes	Number of Comorbidities	Disease Free Status	Total Cholesterol	Age at Menopause	Any Cancer	Hypertension
Meta Analysis p-values		p=7.2E-51	p=2.6E-8	p=2.7E-9	p=9.0E-9	p=1.1E-7	p=0.00048	p=6.6E-9	p=0.157	p=8.7E-5
LBC1921 Females	coeff	1.03		0.042	0.007	0.009	-0.039			
	p-value	0.013		0.230	0.171	0.922	0.0017			
LBC1921 Males	coeff	1.06		0.033	0.007	-0.027	-0.027			
	p-value	1.62E-06		0.392	0.273	0.161	0.029			
LBC1936 Females	coeff	1.08		0.046	0.008	-0.006	-0.010	-0.081		
	p-value	3.76E-08		0.0015	0.153	0.672	0.112	0.043		
LBC1936 Males	coeff	1.09		0.048	0.017	-0.038	-0.011			
	p-value	9.24E-12		0.0032	0.0031	0.0048	0.045			
InChianti Females	coeff	1.06		0.018	-0.002	-0.006	-0.046	-2.96	-0.059	0.0037
	p-value	0.011		0.635	0.842	0.868	0.223	0.00034	0.176	0.881
InChianti Males	coeff	1.07		-0.070	2.11E-05	-0.045	-0.004		0.045	0.026
	p-value	1.01E-06		0.135	0.998	0.215	0.905		0.292	0.219
WHI	coeff	1.05	1.04	0.050	0.021	-0.031	-0.008	-0.060	0.025	0.024
	p-value	8.06E-09	1.20E-05	0.00052	0.0011	0.0041	0.570	3.82E-05	0.063	0.014
JHS Females	coeff	1.15	1.13	0.054	0.029	-0.062	-0.116	0.747**		0.057
	p-value	1.96E-15	0.00025	0.0016	4.12E-07	0.00024	0.696	0.152		0.00071
JHS Males	coeff	1.06	1.06	0.012	0.012	-0.034	-1.148			0.021
	p-value	9.25E-07	0.041	0.495	0.066	0.082	0.0006			0.232

* Hazard Ratios

** Not age at menopause; menopause status

Supplemental Table B.6 accompanies Section A.1.1. We analyze the biological importance of the CpG loci chosen for the DNAm fitness biomarker models using GREAT.

Table B.5: Comparing DNAmFitAge Importance with other DNAm Biomarkers for Time-to-Death and Number of Comorbidities after controlling for age and sex

		LBC1921		LBC1936		InChianti		WHI		JHS	
Time-to-Death Model Comparison		LRT	LRT p-value	LRT	LRT p-value	LRT	LRT p-value	LRT	LRT p-value	LRT	LRT p-value
DNAmGrimAge + DNAmFitAge	Females	0.5	0.479	2.9	0.091	7.2	0.007	1.1	0.286	4.6	0.032
to DNAmGrimAge	Males	3.7	0.054	2.6	0.110	7.7	0.005			0.2	0.628
DNAmPhenoAge + DNAmFitAge	Females	9.1	0.003	36.0	1.98E-09	1.2	0.269	17.0	3.70E-05	30.4	3.53E-08
to DNAmPhenoAge	Males	11.3	7.64E-04	91.0	<1.0E-16	26.4	2.76E-07			4.3	0.039
DNAmPAI1 + DNAmFitAge	Females	9.8	0.002	51.1	8.67E-13	7.3	0.007	17.0	3.76E-05	38.5	5.36E-10
to DNAmPAI1	Males	30.4	3.49E-08	83.6	<1.0E-16	22.9	1.67E-06			7.1	0.008
DNAmGDF15 + DNAmFitAge	Females	5.2	0.023	44.8	2.16E-11	6.1	0.014	23.1	1.54E-06	46.6	8.88E-12
to DNAmGDF15	Males	25.0	5.78E-07	70.7	<1.0E-16	14.1	1.73E-04			4.4	3.66E-02
DNAmAgeHannum + DNAmFitAge	Females	13.2	2.79E-04	60.8	6.11E-15	2.0	0.157	31.5	1.95E-08	41.6	1.11E-10
to DNAmAgeHannum	Males	15.0	1.09E-04	104.0	<1.0E-16	22.0	2.70E-06			7.0	0.008
DNAmAgeSkinBloodClock + DNAmFitAge	Females	16.4	5.05E-05	92.3	<1.0E-16	3.6	0.058	34.7	3.79E-09	57.2	3.89E-14
to DNAmAgeSkinBloodClock	Males	22.4	2.17E-06	133.8	<1.0E-16	21.8	3.01E-06			10.9	9.46E-04
Number of Comorbidities Model Comparison											
DNAmGrimAge + DNAmFitAge	Females	2.1	0.148	1.2	0.269	0.4	0.513	3.4	0.065	0.0	0.910
to DNAmGrimAge	Males	2.5	0.117	3.1	0.080	0.05	0.828			1.2	0.267
DNAmPhenoAge + DNAmFitAge	Females	0.05	0.828	2.6	0.110	3.4	0.067	2.9	0.091	8.6	0.003
to DNAmPhenoAge	Males	5.2	0.023	38.7	4.98E-10	0.01	0.927			0.7	0.412
DNAmPAI1 + DNAmFitAge	Females	0.7	0.401	1.3	0.255	0.9	0.344	1.4	0.230	2.7	0.101
to DNAmPAI1	Males	1.4	0.233	26.5	2.70E-07	0.1	0.800			0.1	0.817
DNAmGDF15 + DNAmFitAge	Females	0.5	0.476	5.3	0.021	0.01	0.944	6.7	0.010	22.1	2.61E-06
to DNAmGDF15	Males	2.6	0.105	31.6	1.86E-08	0.6	0.453			4.2	0.041
DNAmAgeHannum + DNAmFitAge	Females	0.03	0.871	2.4	0.123	0.7	0.411	6.5	0.011	13.1	2.88E-04
to DNAmAgeHannum	Males	2.6	0.108	39.2	3.78E-10	0.2	0.624			1.1	0.305
DNAmAgeSkinBloodClock + DNAmFitAge	Females	0.3	0.596	5.6	0.018	0.2	0.682	8.3	0.004	16.5	4.83E-05
to DNAmAgeSkinBloodClock	Males	1.3	0.256	55.0	1.21E-13	0.2	0.676			2.9	0.089

Table B.6: CpG Annotation and Chromatin State Results

A. Top GREAT CpG Annotation Results

Genes	Observed Regions	Fold Enrichment	Binomial p-value	Bonferroni p-value
ZNRD1	4	77.9	2.75E-07	0.0051
HLA-G	4	55.0	1.09E-06	0.020
Cellular				
MHC protein complex	9	25.1	1.86E-10	3.11E-07
integral component of endoplasmic reticulum membrane	21	3.7	4.49E-07	0.00075
intrinsic component of endoplasmic reticulum membrane	21	3.6	6.39E-07	0.0011
MHC class II protein complex	5	26.9	1.56E-06	0.0026
integral component of lumenal side of endoplasmic reticulum membrane	7	12.7	1.81E-06	0.0030
Molecular				
peptide antigen binding	6	13.3	7.71E-06	0.032
tapasin binding	2	421.0	1.12E-05	0.047

B. Chromatin State Enrichment

State	Description	CpG loci	Odds Ratio	Hypergeometric p-value
PromF4	promoter; heavily acetylated - flanking tss downstream bias	25	0.45	6.5E-06
TSS1	TSS more acetylated and active	15	0.37	6.8E-06
BivProm2	weak bivalent promoter- stronger on H3K27me3	43	1.76	0.00057
TxEx3	exon; H3K36me3 strong	4	0.30	0.0030
DNase1	DNase I only	13	2.41	0.0041
ReprPC1	polycomb repressed; H3K27me3 strong and H3K4me1 weak	21	1.87	0.0065
BivProm1	weak bivalent promoter - more balanced H3K4me3/ H3K27me3	43	1.50	0.0092

B.1.2 Copula Transforms and Imputation

Supplemental Table B.7 accompanies Section 3.3.3 to evaluate the impact of SNP proximity in imputation accuracy.

Table B.7: Imputation Performance by Presence of SNP

Dataset	Imputation Tool	Dataset Form	No SNP		SNPs	
			Median RMSE	Median Correlation	Median RMSE	Median Correlation
FHS	impute.knn	Untransformed	0.019	0.64	0.02	0.628
		Missing Normal	0.018	0.684	0.019	0.678
		Normal 05	0.0185	0.698	0.0195	0.688
	imputePCA	Untransformed	0.014	0.788	0.015	0.775
		Missing Normal	0.016	0.742	0.017	0.725
		Normal 001	0.014	0.784	0.015	0.773
CALERIE	impute.knn	Untransformed	0.015	0.371	0.017	0.312
		Missing Normal	0.014	0.549	0.016	0.49
		Normal 001	0.014	0.517	0.016	0.456
	imputePCA	Untransformed	0.012	0.634	0.014	0.569
		Missing Normal	0.012	0.633	0.014	0.574
		Normal 001	0.012	0.633	0.014	0.578

Supplemental Table B.8 accompanies Section 3.3.4 to evaluate the computational demand of each imputation method in different datasets.

Table B.8: Imputation Tool Computational Demand

Dataset	Imputation Tool	Time (hours)	Memory (GB)	Time per data point (microseconds)	Memory per datapoint kilobytes)
FHS	impute.knn	1.64	64.3	2.83	0.185
	imputePCA	28.7	99.6	49.6	0.287
CALERIE	impute.knn	0.048	16.1	2.34	1.50
	imputePCA	1.31	15.0	63.1	1.35

Supplemental Figure B.2 accompanies Section 3.3.3 to evaluate the impact of SNP proximity in imputation accuracy.

Supplemental Figure B.3 accompanies Section A.6 by providing additional results evaluating methyLImp in a small subset of the FHS dataset.

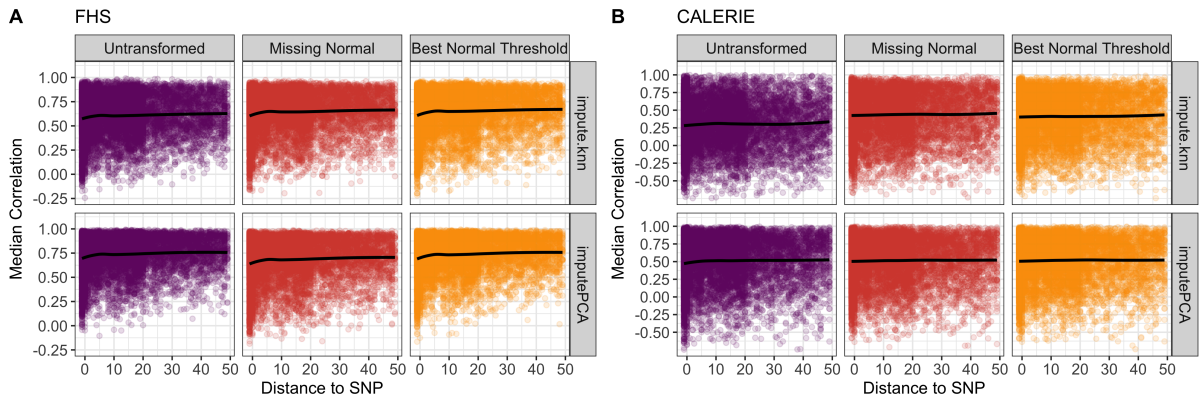


Figure B.2: Scatterplot of median imputation correlation to distance to SNP. Regardless of transformation or not, median correlation stays relatively constant except near low distances (<5 bp) where accuracy decreases.

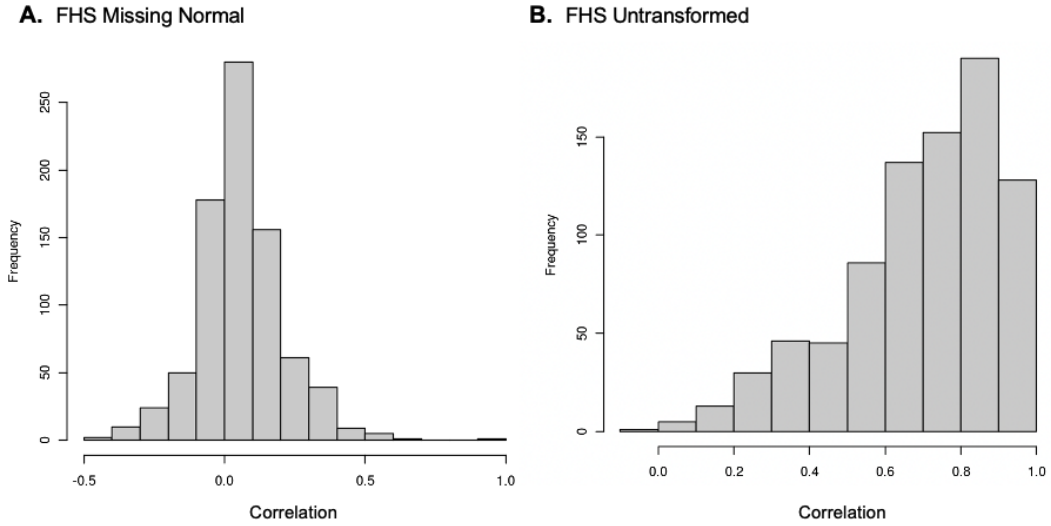


Figure B.3: Distribution of imputation correlation accuracy in 828 CpG probes using methyLImp. (A) features the Missing Normal transformation method and (B) features the Untransformed method.

B.1.3 Cross Tissue DNAm Biomarker Prediction

Supplemental Table B.9 accompanies Section 4.3.3 to compare all methods to predict DNAm biomarkers across tissues.

Table B.9: Complete Comparison Among Transfer Learning, Lasso, and Saliva Surrogate Methods to Estimate Blood DNAm Biomarkers in C+S Methods

	Transfer Learning C+S Method		Transfer Learning C Method		Lasso with Saliva DNAm Biomarkers		Saliva DNAm Biomarkers		C+S to Saliva		C+S to Lasso	
Biomarker	MSE	Corr	MSE	Corr	MSE	Corr	MSE	Corr	Corr Diff	MSE Diff	Corr Diff	MSE Diff
CD4.naive	5.4E+04	0.285	4.5E+04	0.54	1.1E+06	0.597	4.7E+04	0.590	-0.305	-7413.94	-0.312	1.1E+06
CD8.naive	9.3E+03	0.104	1.0E+04	0.274	5.9E+05	-0.091	8.3E+03	0.480	-0.376	-962.02	0.195	5.8E+05
CD8pCD28nCD45RAn	51.23	-0.165	55.9	-0.0831	525.5	0.118	52.7	0.335	-0.500	1.50	-0.283	474.2
DNAmADM	1198.0	0.641	2380.0	0.665	28482.6	-0.368	2.1E+03	0.525	0.116	909.65	1.009	27284.7
DNAmAge	75.31	0.873	75.4	0.937	1265.2	0.476	164.9	0.713	0.160	89.64	0.398	1189.9
DNAmAgeHannum	107.5	0.845	112.0	0.832	6391.1	-0.128	227.9	0.703	0.141	120.38	0.972	6283.5
DNAmB2M	1.0E+11	0.627	1.1E+11	0.781	2.3E+11	0.732	5.0E+10	0.650	-0.024	-5.2E+10	-0.106	1.3E+11
DNAmCystatinC	9.3E+09	0.415	5.6E+09	0.674	4.1E+10	0.638	8.7E+09	0.513	-0.099	-6.3E+08	-0.223	3.2E+10
DNAmFEV1_noAge	0.47	0.588	0.65	0.589	14.50	-0.261	0.6	0.630	-0.042	0.10	0.849	14.0
DNAmFitAge	131.3	0.875	134.0	0.896	1894.0	0.399	211.5	0.721	0.154	80.13	0.476	1762.7
DNAmGait_noAge	0.05	0.803	0.0485	0.83	4.98	0.258	0.1	0.535	0.268	0.03	0.545	4.9
DNAmGDF15	6.2E+04	0.661	1.1E+05	0.437	1.0E+06	0.024	6.8E+04	0.694	-0.033	6477.09	0.638	9.6E+05
DNAmGrimAge2Based OnPredictedAge	151.9	0.775	207	0.689	2175.0	-0.302	206.1	0.754	0.021	54.22	1.077	2023.1
DNAmGrimAge2Based OnRealAge	124.6	0.741	150	0.792	1292.1	0.235	196.1	0.694	0.048	71.58	0.506	1167.5
DNAmGrimAgeBased OnPredictedAge	168.8	0.749	109	0.881	876.2	0.157	141.0	0.791	-0.043	-27.89	0.592	707.3
DNAmGrimAgeBased OnRealAge	151.3	0.701	94.4	0.872	684.2	0.671	126.9	0.741	-0.039	-24.41	0.030	532.9
DNAmGrip_noAge	27.93	0.885	21.6	0.897	800.3	0.302	92.4	0.521	0.364	64.44	0.583	772.3
DNAmLeptin	3.7E+07	0.660	9.0E+07	0.425	2.3E+09	-0.221	1.2E+08	0.302	0.359	8.5E+07	0.881	2.2E+09
DNAmPACKYRS	136.9	0.835	183	0.517	1049.7	0.774	125.3	0.690	0.145	-11.68	0.061	912.7
DNAmPAI1	2.5E+07	0.351	2.7E+07	0.503	1.1E+08	-0.014	1.1E+07	0.450	-0.098	-1.4E+07	0.365	8.6E+07
DNAmPhenoAge	188.5	0.768	228	0.758	5612.1	-0.192	306.6	0.718	0.050	118.09	0.959	5423.7
DNAmTIMP1	4.6E+06	0.731	3.9E+06	0.905	7.0E+07	0.804	5.0E+06	0.724	0.006	4.4E+05	-0.073	6.5E+07
DNAmTL	0.30	0.515	0.257	0.622	57.57	-0.494	0.5	0.541	-0.025	0.21	1.009	57.3
DNAmVO2max	7.36	0.625	7.04	0.736	2337.0	0.451	8.1	0.618	0.007	0.72	0.174	2329.6
Gran	0.13	-0.109	0.137	-0.232	1.42	-0.390	0.2	-0.208	0.099	0.05	0.281	1.3
PlasmaBlast	0.20	0.246	0.236	0.0297	3.12	0.017	0.4	0.010	0.236	0.19	0.229	2.9

Table B.10: Relationship of DNAmTL to Sex in GSE119078

Biomarker	Mean Males (n=25)	Mean Females (n=34)	Females - Males	t-test p-value	Kruskal Wallis p-value
Saliva DNAmTL	6.54	6.88	0.33	1.4E-04	4.2E-04
C+S DNAmTL Blood Prediction	6.86	7.16	0.29	8.3E-06	2.1E-05
C DNAmTL Blood Prediction	7.20	7.50	0.30	5.3E-04	3.5E-04

Supplemental Table B.10 accompanies Section 4.3.4 to validate the DNAmTL algorithms in relating to females having longer telomere lengths.

Supplemental Figure B.4 accompanies Section 4.3.3 by providing additional performance metrics comparing different methods for cross tissue prediction of DNAm biomarkers.

Supplemental Figure B.5 accompanies Section 4.3.1 by demonstrating variability in LODO performance based on different parameterizations within each biomarker.

Supplemental Table B.11 accompanies Section 4.3.5 to test the algorithms in supplying tissues other than saliva for predicting blood DNAm biomarkers.

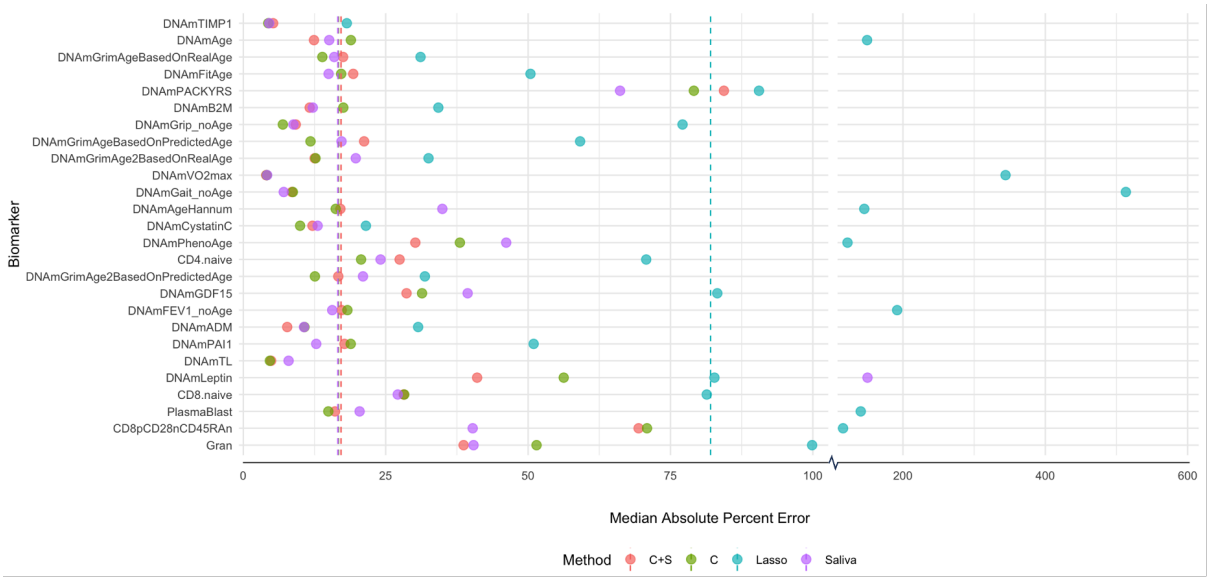


Figure B.4: Median Absolute Percent Error (MeAPE) between True and Estimated Blood DNAm Biomarkers by top Performing TL Methods, Lasso, and Saliva Surrogates. Median MeAPE presented as dotted line with a change in X axis scaling. LODO MeAPE presented for C+S, C, and Lasso methods.

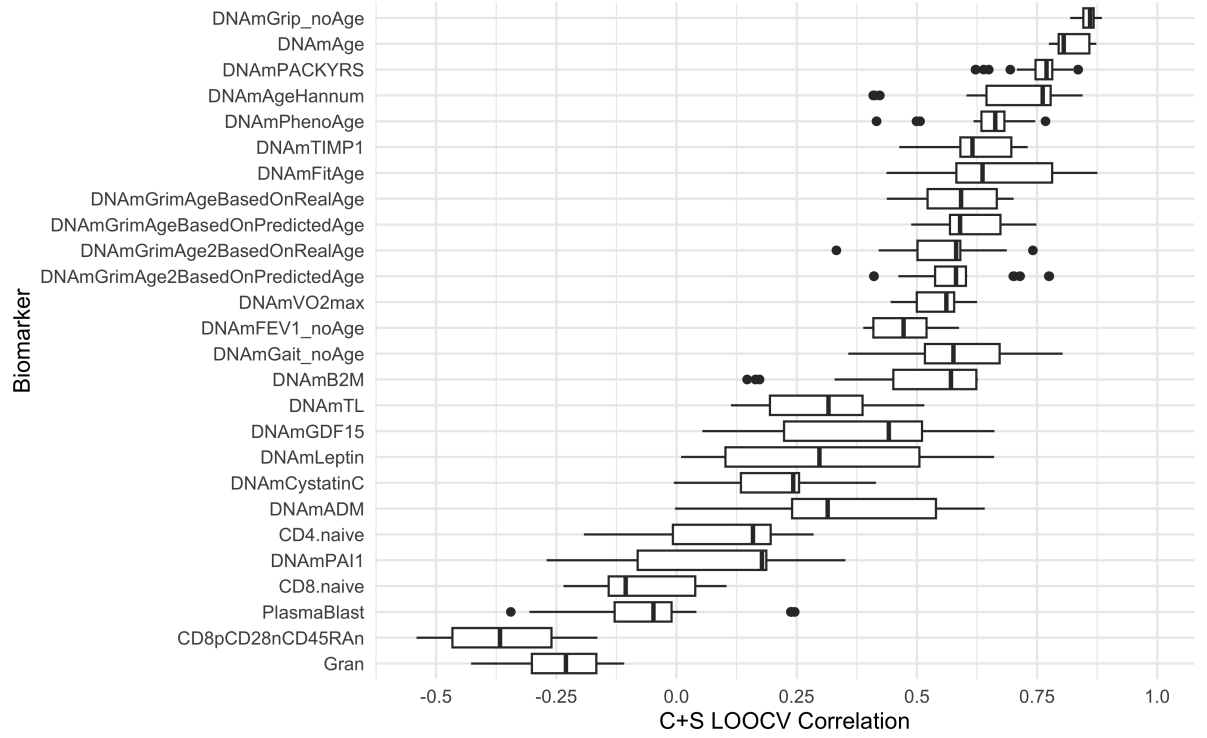


Figure B.5: Boxplots of C+S Method LODO Correlation by DNAm Biomarker to demonstrate variability in C+S method parameterizations.

Table B.11: Evaluation of Lymph, Adipose, and Muscle Tissue as TL Algorithm Input After Controlling for Age in HorvathHIV dataset

Biomarker	Lymph Node (n=28)		Adipose (n=58)		Muscle (n=57)	
	HIV + Effect	p-value	HIV + Effect	p-value	HIV + Effect	p-value
CD4.naive C Prediction	-39.6	0.113	-21.5	0.213	2.06	0.801
CD8.naive C Prediction	-23.8	0.177	0.39	0.957	-9.26	0.164
DNAmADM C Prediction	10.01	0.095	-7.36	0.095	-10.6	0.014
DNAmAge C Prediction	6.11	0.043	-2.72	0.091	-3.13	0.050
DNAmAgeHannum C Prediction	5.59	0.080	-1.64	0.285	-1.69	0.201
DNAmB2M C Prediction	65721	0.064	-32378	0.131	-34684	0.114
DNAmCystatinC C Prediction	46021	0.018	-21372	0.021	-9953	0.141
DNAmFEV1_noAge C Prediction	-0.24	0.168	0.36	0.009	0.26	0.038
DNAmFitAge C Prediction	2.45	0.295	-4.66	0.014	-1.69	0.278
DNAmGait_noAge C Prediction	0.04	0.243	0.04	0.107	0.02	0.409
DNAmGDF15 C Prediction	265	0.039	-28.48	0.453	-68.5	0.177
DNAmGrimAge2BasedOnPredictedAge C Prediction	3.52	0.127	-3.00	0.047	-0.79	0.543
DNAmGrimAge2BasedOnRealAge C Prediction	3.92	0.071	-2.63	0.042	-0.45	0.702
DNAmGrimAgeBasedOnPredictedAge C Prediction	4.24	0.067	-3.90	0.030	-1.10	0.440
DNAmGrimAgeBasedOnRealAge C Prediction	3.66	0.120	-3.15	0.019	-0.94	0.442
DNAmGrip_noAge C Prediction	-3.63	0.197	4.22	0.045	4.03	0.064
DNAmLeptin C Prediction	894	0.696	-1318	0.189	414.1	0.775
DNAmPACKYRS C Prediction	6.59	0.298	-0.54	0.750	0.36	0.847
DNAmPAI1 C Prediction	519	0.182	5.07	0.983	-51.4	0.822
DNAmPhenoAge C Prediction	9.45	0.012	-2.25	0.199	-1.56	0.355
DNAmTIMP1 C Prediction	931	0.060	-561.5	0.080	-298	0.191
DNAmTL C Prediction	-0.12	0.458	0.01	0.870	-0.02	0.633
DNAmVO2max C Prediction	-0.94	0.098	0.97	0.009	0.65	0.049

Supplemental Table B.12 accompanies Section 4.3.5 to test the algorithms in supplying tissues other than saliva for predicting blood DNAm biomarkers, specific for motivating their application in mammalian skin samples.

Supplemental Table B.13 accompanies Section 4.3.6 to demonstrate minimal difference in calculating correlation performance in mammalian species and consistency of performance in species.

Table B.12: Correlation Between Predicted DNAm Biomarker from Skin to Age at Skin Biopsy in TwinsUK Sample (n=95)

Biomarker	Pearson R	p-value	In expected direction?
DNAmAge	0.776	2.4E-20	Yes
DNAmFitAge	0.751	1.9E-18	Yes
DNAmGrimAgeBasedOnRealAge	0.723	1.2E-16	Yes
DNAmGait_noAge	-0.684	2.3E-14	Yes
DNAmTIMP1	0.667	1.5E-13	Yes
DNAmAgeHannum	0.638	3.5E-12	Yes
DNAmB2M	0.617	2.7E-11	Yes
DNAmGrimAgeBasedOnPredictedAge	0.575	1.1E-09	Yes
DNAmPhenoAge	0.554	5.6E-09	Yes
DNAmGrimAge2BasedOnRealAge	0.500	2.5E-07	Yes
DNAmADM	0.414	3.0E-05	Yes
DNAmGrimAge2BasedOnPredictedAge	0.364	2.9E-04	Yes
DNAmCystatinC	0.345	6.1E-04	Yes
DNAmVO2max	-0.237	0.021	Yes
DNAmTL	-0.202	0.049	Yes
DNAmGDF15	0.198	0.054	Yes
DNAmFEV1_noAge	-0.191	0.063	Yes
CD8.naive	-0.186	0.071	Yes
DNAmGrip_noAge	-0.172	0.096	Yes
DNAmLeptin	-0.154	0.137	-
DNAmPAI1	0.143	0.166	Yes
CD4.naive	-0.029	0.779	Yes
DNAmPACKYRS	-0.027	0.794	No

Table B.13: Average Unweighted Correlation and Biomarker Consistency in Species with at least 10 samples (species = 40, n=1959)

Biomarker	Average Correlation	In expected direction?	Species with Corr in Exp Direction (%)	Species with Sig Corr in Exp Direction (%)
DNAmAge	0.539	Yes	97.5%	75%
DNAmVO2max	-0.346	Yes	92.5%	52.5%
DNAmAgeHannum	0.292	Yes	85%	40%
DNAmGrip_noAge	-0.276	Yes	85%	37.5%
DNAmFitAge	0.267	Yes	80%	45%
CD4.naive	-0.233	Yes	77.5%	32.5%
DNAmGrimAgeBasedOnRealAge	0.233	Yes	80%	30%
DNAmFEV1_noAge	-0.210	Yes	77.5%	30%
DNAmGrimAgeBasedOnPredictedAge	0.204	Yes	75%	40%
DNAmPhenoAge	0.176	Yes	65%	20%
DNAmADM	0.168	Yes	77.5%	30%
CD8.naive	-0.157	No	20%	5%
DNAmGait_noAge	-0.150	Yes	70%	37.5%
DNAmTIMP1	0.124	Yes	65%	20%
DNAmGDF15	-0.118	No	35%	12.5%
DNAmGrimAge2BasedOnRealAge	0.102	Yes	60%	17.5%
DNAmGrimAge2BasedOnPredictedAge	0.101	Yes	62.5%	17.5%
DNAmB2M	0.070	Yes	62.5%	7.5%
DNAmPACKYRS	-0.042	No	47.5%	7.5%
DNAmTL	0.020	No	45%	10%
DNAmCystatinC	-0.008	No	45%	20%
DNAmLeptin	0.006		-	-
DNAmPAI1	-0.003	No	52.5%	10%

REFERENCES

[1] Benjamini Y and Hochberg Y, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society*, vol. Series B, no. 57, pp. 289–300, 1995.

[2] Bonferroni CE, "Il calcolo delle assicurazioni su gruppi di teste," *Studi in onore del professore salvatore ortu carboni*, pp. 13–60, 1935.

[3] Uffelmann E, et al, "Genome-wide association studies," *Nature Reviews Methods Primers*, vol. 1, no. 1, pp. 1–21, 2021.

[4] Visscher PM, et al, "10 years of gwas discovery: biology, function, and translation," *The American Journal of Human Genetics*, vol. 101, no. 1, pp. 5–22, 2017.

[5] Zou H and Hastie T, "Regularization and variable selection via the elastic net," *Journal of the royal statistical society: series B (statistical methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[6] Horvath S, "Dna methylation age of human tissues and cell types," *Genome Biol*, vol. 14, no. 10, 2013.

[7] Lu AT, et al, "Dna methylation grimage strongly predicts lifespan and healthspan," *Aging (Albany NY)*, vol. 11, no. 2, 2019.

[8] Moore LD, Le T, and Fan G, "Dna methylation and its basic function," *Neuropsychopharmacology*, vol. 38, no. 1, pp. 23–38, 2013.

[9] Du P, et al, "Comparison of beta-value and m-value methods for quantifying methylation levels by microarray analysis," *BMC bioinformatics*, vol. 11, no. 1, pp. 1–9, 2010.

[10] Akalin A, Kormaksson M, Li S, Garrett-Bakelman FE, Figueroa ME, Melnick A, and Mason CE, "methylkit: a comprehensive r package for the analysis of genome-wide dna methylation profiles," *Genome Biology*, vol. 13, no. 10, 2012.

[11] Dolzhenko E and Smith AD, "Using beta-binomial regression for high-precision differential methylation analysis in multifactor whole-genome bisulfite sequencing experiments," *BMC Bioinformatics*, vol. 15, no. 215, June 2014.

[12] Levine ME, Lu AT, Quach A, et al, "An epigenetic biomarker of aging for lifespan and healthspan," *Aging (Albany NY)*, vol. 10, no. 4, pp. 573–591, 2018.

[13] Marioni RE, Shah S, McRae AF, et al, "Dna methylation age of blood predicts all-cause mortality in later life," *Genome Biol*, vol. 16, no. 25, 2015.

[14] Chen BH, Marioni RE, et al, "Dna methylation-based measures of biological age: meta-analysis predicting time to death," *Aging (Albany NY)*, vol. 8, no. 9, pp. 1844–1865, 2016.

[15] Bocklandt S, et al, "Epigenetic predictor of age," *PloS one*, vol. 6, no. 6, 2011.

[16] Houseman EA, Accomando WP, Koestler DC, et al, "Dna methylation arrays as surrogate measures of cell mixture distribution," *BMC Bioinformatics*, vol. 13, no. 86, 2012.

[17] Belsky DW, et al, "Dunedinpace, a dna methylation biomarker of the pace of aging," *Elife*, vol. 11, no. e73420, Jan 2022.

[18] Jackson AS, Sui X, Hébert JR, Church TS, and Blair SN, "Role of lifestyle and aging on the longitudinal change in cardiorespiratory fitness," *Arch Intern Med*, vol. 169, no. 19, pp. 1781–1787, 2009.

[19] Radak Z, Torma F, et al, "Exercise effects on physiological function during aging," *Free Radic Biol Med*, vol. 132, pp. 33–41, Feb 2019.

[20] McGreevy KM, Radak Z, et al, "Dnamfitage: biological age indicator incorporating physical fitness," *Aging (Albany NY)*, vol. 15, no. 10, Feb 2023.

[21] Harridge SD and Lazarus NR, "Physical activity, aging, and physiological function," *Physiology*, vol. 32, no. 2, pp. 152–161, 2017.

[22] Agusti A and Faner R, "Lung function trajectories in health and disease," *The Lancet Respiratory Medicine*, vol. 7, no. 4, pp. 358–364, Apr 2019.

[23] Frederiksen H, et al, "Age trajectories of grip strength: Cross-sectional and longitudinal data among 8,342 danes aged 46 to 102," *Annals of Epidemiology*, vol. 16, no. 7, pp. 554–562, July 2006.

[24] Rapp D, Scharhag J, Wagenpfeil S, et al, "Reference values for peak oxygen uptake: cross-sectional analysis of cycle ergometry-based cardiopulmonary exercise tests of 10090 adult german volunteers from the prevention first registry," *BMJ Open*, vol. 8, no. e018697, 2018.

[25] van Oostrom SH, Engelfriet PM, Verschuren WMM, Schipper M, Wouters IM, Boezen M, et al, "Aging-related trajectories of lung function in the general population—the doetinchem cohort study," *PLoS ONE*, vol. 13, no. 5, p. page, 2018.

[26] Ching SM, Chia YC, Lentjes MAH, Luben R, Wareham N, and Khaw KT, "Fev1 and total cardiovascular mortality and morbidity over an 18 years follow-up population-based prospective epic-norfolk study," *BMC Public Health*, vol. 19, no. 1, May 2018.

[27] Rezwan FI, Imboden M, et al, "Association of adult lung function with accelerated biological aging," *Aging (Albany NY)*, vol. 12, no. 1, pp. 518–542, Jan 2020.

[28] Lazarus NR, et al, "The relationships and interactions between age, exercise and physiological function," *The Journal of Physiology*, vol. 597, no. 5, pp. 1299–1309, Dec 2018.

[29] Horvath S, et al, "Obesity accelerates epigenetic aging of human liver," *Proc Natl Acad Sci USA*, vol. 111, no. 43, Oct 2014.

[30] Soerensen M, et al, "Epigenome-wide exploratory study of monozygotic twins suggests differentially methylated regions to associate with hand grip strength," *Biogerontology*, vol. 20, no. 5, pp. 627–647, Oct 2019.

[31] Ferrari L, et al, "Effects of physical exercise on endothelial function and dna methylation," *Int. J. Environ. Res. Public Health*, vol. 16, no. 2530, 2019.

[32] Spólnicka M, Pośpiech E, et al, "Modified aging of elite athletes revealed by analysis of epigenetic age markers," *Aging (Albany NY)*, vol. 10, no. 2, pp. 241–252, Feb 2018.

[33] Hughes DC, Ellefsen S, and Baar K, "Adaptations to endurance and strength training," *Cold Spring Harb Perspect Med*, vol. 8, no. 6, Jun 2018.

[34] Studenski S, Perera S, Patel K, et al, "Gait speed and survival in older adults," *JAMA*, vol. 305, no. 1, pp. 50–58, 2011.

[35] Dawber TR, Meadors GF, and Moore FE, "Epidemiological approaches to heart disease: the framingham study," *Jr Am J Public Health Nations Health*, vol. 41, no. 3, pp. 279–281, Mar 1951.

[36] Ferrucci L, "The baltimore longitudinal study of aging (blsa): a 50-year-long journey and plans for the future," *J Gerontol A Biol Sci Med Sci*, vol. 63, no. 12, Dec 2008.

[37] Key TJ, Appleby PN, Allen NE, and Reeves GK, "Pooling biomarker data from different studies of disease risk, with a focus on endogenous hormones," *Cancer Epidemiol Biomarkers Prev*, vol. 19, no. 4, Apr 2010.

[38] Rankovic G, Mutavdzic V, Toskic D, et al, "Aerobic capacity as an indicator in different kinds of sports," *Bosn J Basic Med Sci*, vol. 10, no. 1, pp. 44–48, 2010.

[39] Tibshirani R, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[40] Hoerl AE and Kennard RW, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[41] Zou H and Hastie T, "Regularization and variable selection via the elastic net," *Journal of the royal statistical society: series B (statistical methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[42] Klemera P and Doubal S, "A new approach to the concept and computation of biological age," *Mech Ageing Dev*, vol. 127, no. 3, Mar 2006.

[43] Hannum G, et al, "Genome-wide methylation profiles reveal quantitative views of human aging rates," *Mol Cell*, vol. 49, no. 2, pp. 359–367, Jan 2013.

[44] Dhingra R, Kwee L, et al, "Evaluating dna methylation age on the illumina methylationepic bead chip," *PLoS ONE*, vol. 14, May 2019.

[45] Sklar A, "Fonctions de répartition à n dimensions et leurs marges," *Publ. Inst. Statist Univ. Paris*, vol. 8, pp. 229–231, 1959.

[46] Low RKY, Faff R, and Aas K, "Enhancing mean–variance portfolio selection by modeling distributional asymmetries," *Journal of Economics and Business*, vol. 85, pp. 49–72, May 2016.

[47] Rickman AD, Williamson DA, Martin CK, Gilhooly CH, Stein RI, Bales CW, Roberts S, Das SK, "The calerie study: design and methods of an innovative 25

[48] Bareja, A, Lee DE, et al, "Liver-derived plasminogen mediates muscle stem cell expansion during caloric restriction through the plasminogen receptor, plg-rkt (forthcoming)," 2023.

[49] Di Lena P, Sala C, Prodi A, and Nardini C, "Missing value estimation methods for dna methylation data," *Bioinformatics*, vol. 35, no. 19, pp. 3786–3793, Oct 2019.

[50] Husson F, "A package for handling missing values in multivariate data analysis," *Journal of Statistical Software*, vol. 70, no. 1, pp. 1–31, 2016.

[51] Troyanskaya O, et al, "Missing value estimation methods for dna microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, Jun 2001.

[52] Silverman BW, *Density Estimation*. London: Chapman and Hall / CRC, 1986.

[53] P. Hoff, "Extending the Rank Likelihood for Semiparametric Copula Estimation," *The Annals of Applied Statistics*, vol. 1, no. 1, pp. 265–283, 2007.

[54] Epanechnikov VA, "Non-parametric estimation of a multivariate probability density," *Theory of Probability Its Applications*, vol. 14, no. 1, pp. 153–158, 1969.

[55] Forsythe GE, Malcolm MA, and Moler CB, "Computer methods for mathematical computations," *Wiley*, 1977.

[56] Kolmogorov A, "Sulla determinazione empirica di una legge di distributione," *Giornale dell' Istituto Italiano degli Attuari*, vol. 4, pp. 83–91, 1933.

[57] Bakulski KM et al, "Epigenetic research in neuropsychiatric disorders: the 'tissue issue'," *Curr Behav Neurosci Rep*, vol. 3, no. 3, Sep 2016.

[58] Horvath S, Lin DTS, et al, "Hiv, pathology and epigenetic age acceleration in different human tissues," *Geroscience*, vol. 44, no. 3, June 2022.

[59] Ma B, Wilker EH, et al, "Predicting dna methylation level across human tissues," *Nucleic Acids Res*, vol. 42, no. 6, Apr 2014.

[60] Emily Maciejewski, Steve Horvath, Jason Ernst, "Cross-species and tissue imputation of species-level dna methylation samples across mammalian species," *bioRxiv*.

[61] Huang YT et al, "Epigenome-wide profiling of dna methylation in paired samples of adipose tissue and blood," *Epigenetics*, vol. 3, no. 11, Mar 2016.

[62] Braun PR, Han S, Hing B, Nagahama Y, et al, "Genome-wide dna methylation comparison between live human brain and peripheral tissues within individuals," *Transl Psychiatry*, vol. 9, no. 1, Jan 2019.

[63] Langie S, Moisse M, et al, "Salivary dna methylation profiling: aspects to consider for biomarker identification," *Basic Clin Pharmacol Toxicol*, vol. 121, 2017.

[64] Zhang Q, Vallerga CL, et al, "Improved precision of epigenetic clock estimates across tissues and its implication for biological ageing," *Genome Med*, vol. 11, no. 54, 2019.

[65] Hosna A, Merry E, Gyalmo J, et al, "Transfer learning: a friendly introduction," *J Big Data*, vol. 9, no. 102, Oct 2022.

[66] Dodlapati S, Jiang Z, and Sun J, "Completing single-cell dna methylome profiles via transfer learning together with kl-divergence," *Front Genet*, vol. 13, Jul 2022.

[67] Li S, Cai TT, and Li H, "Transfer learning for high-dimensional linear regression: Prediction, estimation and minimax optimality," *J of the Royal Statistical Society Series B: Statistical Methodology*, vol. 84, no. 1, Feb 2022.

[68] Weiss K, Khoshgoftaar TM, Wang DA, "A survey of transfer learning," *J Big Data*, vol. 3, no. 9, 2016.

[69] Pan SJ and Yang Q, "A survey of transfer learning," *IEEE Trans Knowl Data Eng*, vol. 22, no. 10, 2010.

[70] Torrey L and Shavlik J, *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques.* IGI Global, 2010.

[71] Fang ZX, Lu J, Liu F, Zhang GQ, "Semi-supervised heterogeneous domain adaptation: theory and algorithms," *IEEE Trans Pattern Anal Mach Intell*, vol. 45, 2022.

[72] Yao Y, Li X, Zhang Y, Ye Y, "Multisource heterogeneous domain adaptation with conditional weighting adversarial network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 4, Apr 2023.

[73] Cheng YH, Wang XS, Cao G, "Multi-source tri-training transfer learning," *IEICE Trans Inf Syst*, vol. 97, 2014.

[74] Song B, Pan J, Qu Q, et al, "Multi-source transfer learning based on the power set framework," *Int J Comput Intell Syst*, vol. 16, no. 103, 2023.

[75] Tian Y, Feng Y, "Transfer learning under high-dimensional generalized linear models," *J American Statistical Association*, vol. 118, no. 544, Jun 2022.

[76] Horvath S, Oshima J, et al, "Epigenetic clock for skin and blood cells applied to hutchinson gilford progeria syndrome and ex vivo studies," *Aging (Albany NY)*, vol. 10, no. 7, Jul 2018.

[77] Hillary RF and Marioni RE, "Methyldetectr: a software for methylation-based health profiling," *Wellcome Open Res*, vol. 13, no. 5, Apr 2021.

[78] Yang Z, Wong A, Kuh D, et al, "Correlation of an epigenetic mitotic clock with cancer risk," *Genome Biol*, vol. 17, no. 205, Oct 2016.

[79] Lu AT, Fei Z, Haghani A, et al, "Universal dna methylation age across mammalian tissues," *Nat Aging*, vol. 3, no. 9, Sep 2023.

[80] van der Laan MJ, Polley EC, and Hubbard AE, "Super learner," *Stat Appl Genet Mol Biol*, vol. 6, no. 25, 2007.

[81] Wilkinson GS, Adams DM, Haghani A, et al, "Na methylation predicts age and provides insight into exceptional longevity of bats," *Nat Commun*, vol. 12, no. 1615, 2021.

[82] Browder KC, Reddy P, Yamamoto M, et al, "In vivo partial reprogramming alters age-associated molecular changes during physiological aging in mice," *Nat Aging*, vol. 2, no. 3, Mar 2022.

[83] McLean CY, Bristor D, et al, "Great improves functional interpretation of cis-regulatory regions," *Nat Biotechnol*, vol. 28, 2010.

[84] Vigneron N, Peaper DR, Leonhardt RM, Cresswell P, "Functional significance of tapasin membrane association and disulfide linkage to erp57 in mhc class i presentation," *Eur J Immunol*, vol. 39, 2009.

[85] Silverman MN and Deuster PA, "Biological mechanisms underlying the role of physical fitness in health and resilience," *Interface Focus*, vol. 4, 2014.

[86] Munters LA, Loell I, et al, "Endurance exercise improves molecular pathways of aerobic metabolism in patients with myositis," *Arthritis Rheumatol*, vol. 68, 2016.

[87] Vu H, Ernst J, "Universal annotation of the human genome through integration of over a thousand epigenomic datasets," *Genome Biol*, vol. 23, no. 9, 2022.

[88] Lu AT, Hannon E, et al, "Genetic architecture of epigenetic and neuronal ageing rates in human brain regions," *Nat Commun*, vol. 8, 2017.

[89] Margueron R and Reinberg D, "The polycomb complex prc2 and its mark in life," *Nature*, vol. 469, 2011.

[90] Manson JE, Chlebowski RT, et al, "Menopausal hormone therapy and health outcomes during the intervention and extended poststopping phases of the women's health initiative randomized trials," *JAMA*, vol. 310, 2013.