GIMMEcpg: Global Imputation of Mean CPG Methylation

Niuzheng Chai[1], Ismail Moghul[1,2], Nikolas Pontikos[1,2], Alison Hardcastle[1,2], Javier Herrero[3], Stephan Beck[3†]

[1]UCL Institute of Ophthalmology, [2]Moorfields Eye Hospital, [3]UCL Cancer Institute

Background

DNA methylation is the addition of a methyl group to a cytosine nucleotide (**Figure 1**). Aberrant DNA methylation has been implicated in several human diseases, including cancer[1].

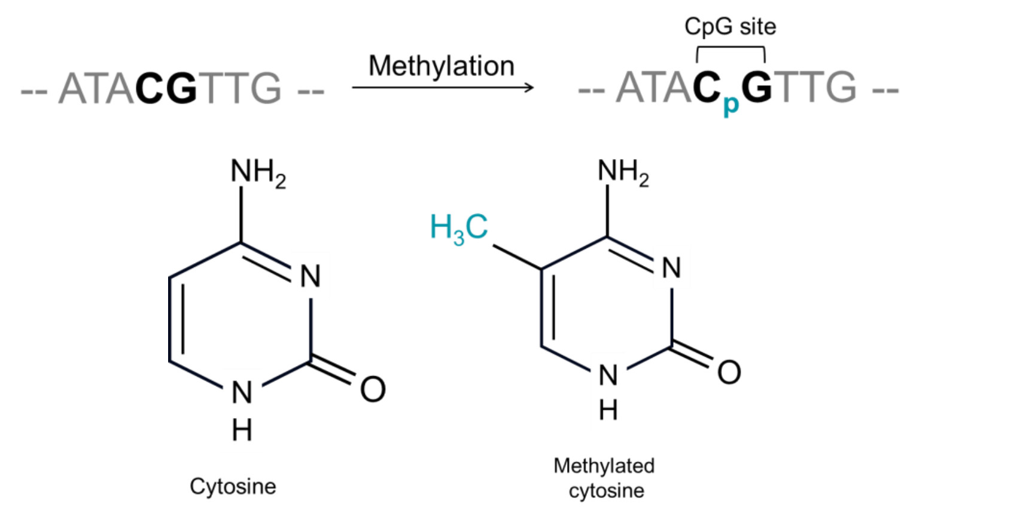


Figure 1. Schematic of DNA methylation

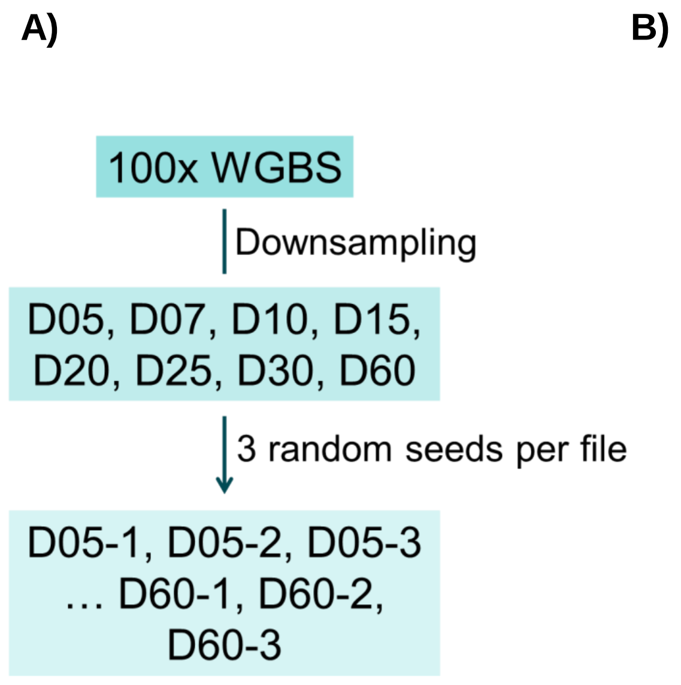
There is therefore great biomedical interest in studying DNA methylation. One way to do so is through Whole Genome Bisulfite Sequencing (WGBS). However, WGBS is currently very expensive. This makes it difficult to produce quality data at large scales. To address the issue of incomplete datasets, several tools have been developed to impute missing values[2-4]. We have developed GIMMEcpg, which is up to 2,675x faster and roughly as accurate as existing tools (**Δ**R: -0.05, **Δ**MAE: +0.009). GIMMEcpg is available as both R and Python packages.

Main Datasets Used

Datasets used for benchmarking were produced by the International Human Epigenome Consortium (IHEC).

- 2 WGBS files, each with a CpG coverage of ~100x (very high quality)
- Downsampled to simulate lower coverage data (**Figure 2**)

A)



100x WGBS

Downsampling

D05, D07, D10, D15, D20, D25, D30, D60

3 random seeds per file

D05-1, D05-2, D05-3 … D60-1, D60-2, D60-3

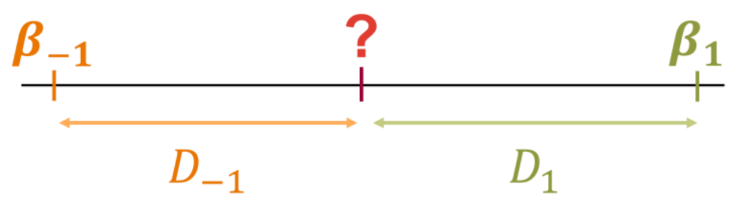
B)

Simulated Coverage (%)	ID	CpG Coverage	Number of CpG sites (M)
5	D05	6.3	13.5
7	D07	8.2	15.8
10	D10	11.2	17.8
15	D15	16.0	19.7
20	D20	20.8	20.7
25	D25	25.6	21.4
30	D30	30.3	21.9
60	D60	57.9	23.4

Figure 2. Downsampling of high-quality 100x WGBS data. A) Workflow to generate downsampled files from one 100x WGBS file. In total, this was done for two 100x WGBS files to produce 48 downsampled files (2x6x3). **B)** Summary of CpG coverage and counts at each downsampled level.

Methods

GIMMEcpg utilises neighbouring CpG methylation statuses and a simple distance-weighted formula to impute missing methylation values (**Figure 3**).



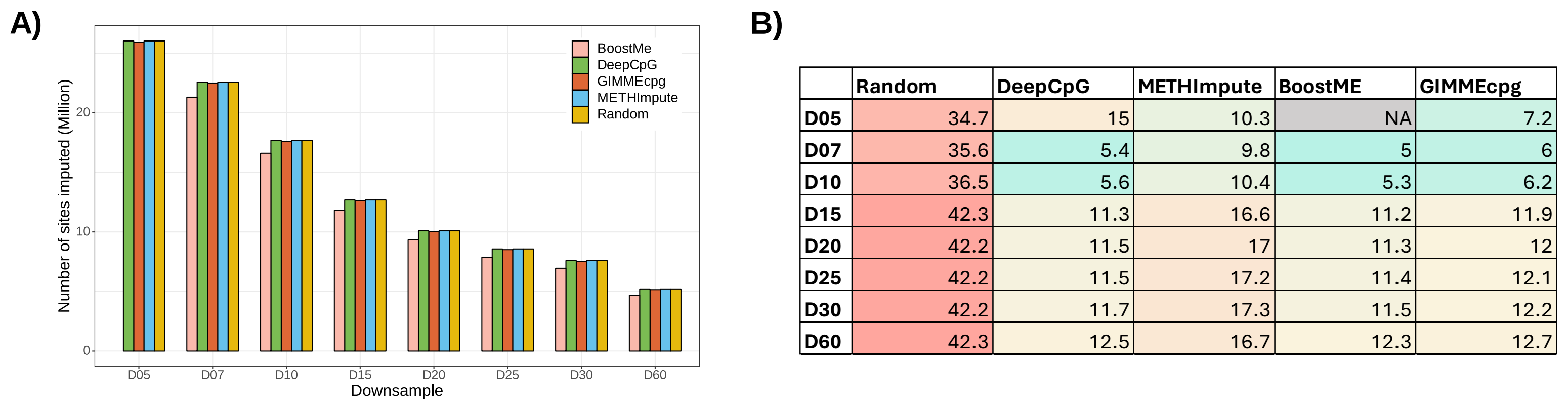
$$\textit{imputed methylation value} = \beta_{-1}\left(1-\frac{|D_{-1}|}{|D_{-1}|+|D_1|}\right)+\beta_1\left(1-\frac{|D_1|}{|D_{-1}|+|D_1|}\right)$$

$$=\frac{\beta_{-1}|D_1|+\beta_1|D_{-1}|}{|D_{-1}|+|D_1|}$$

Figure 3. Formula used to impute missing methylation values based on neighbouring CpG information (distance and methylation).

- CpG sites <1000 nucleotides of each other show similar methylation values[5]
- Use of simple calculations should greatly reduce time and memory usage compared to complex machine-learning based counterparts
- Polars' lazy API to process data in parallel and for automatic query optimisation[6]
- Optional 'accurate mode' makes use of H2OAutoML to train several models based on neighbouring CpG sites[7]
  - Best model based on mean residual deviance used for imputation

Results - Performance

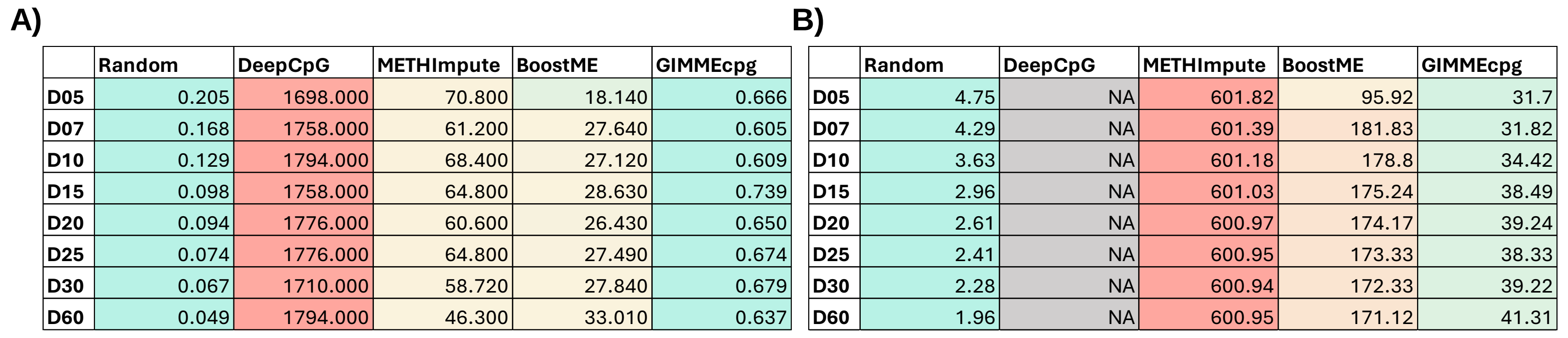


	Random	DeepCpG	METHImpute	BoostME	GIMMEcpg
D05	34.7	15	10.3	NA	7.2
D07	35.6	5.4	9.8	5	6
D10	36.5	5.6	10.4	5.3	6.2
D15	42.3	11.3	16.6	11.2	11.9
D20	42.2	11.5	17	11.3	12
D25	42.2	11.5	17.2	11.4	12.1
D30	42.2	11.7	17.3	11.5	12.2
D60	42.3	12.5	16.7	12.3	12.7

Figure 4. Performance of GIMMEcpg in comparison with other imputation methods, averaged out across 6 files per downsampled level. A) Number of CpG sites (in millions) imputed. As the simulated coverage increases, the number of missing values to impute decreases. **B)** Relative Mean Absolute Error (R-MAE) values of different imputation methods.

- GIMMEcpg imputed as many missing CpG sites as existing tools (**Figure 4A**)
- Accuracy of GIMMEcpg in its default mode was comparable to other tools and much better than random imputation (**Figure 4B**)

Results - Resource Usage



A)

	Random	DeepCpG	METHImpute	BoostME	GIMMEcpg
D05	0.205	1698.000	70.800	18.140	0.666
D07	0.168	1758.000	61.200	27.640	0.605
D10	0.129	1794.000	68.400	27.120	0.609
D15	0.098	1758.000	64.800	28.630	0.739
D20	0.094	1776.000	60.600	26.430	0.650
D25	0.074	1776.000	64.800	27.490	0.674
D30	0.067	1710.000	58.720	27.840	0.679
D60	0.049	1794.000	46.300	33.010	0.637

B)

	Random	DeepCpG	METHImpute	BoostME	GIMMEcpg
D05	4.75	NA	601.82	95.92	31.7
D07	4.29	NA	601.39	181.83	31.82
D10	3.63	NA	601.18	178.8	34.42
D15	2.96	NA	601.03	175.24	38.49
D20	2.61	NA	600.97	174.17	39.24
D25	2.41	NA	600.95	173.33	38.33
D30	2.28	NA	600.94	172.33	39.22
D60	1.96	NA	600.95	171.12	41.31

Figure 5. Time and RAM usage of GIMMEcpg in comparison with other imputation methods, averaged out across 6 files per downsampled level. A) Time taken (minutes) for different imputation methods to calculate missing values. **B)** Random access memory (RAM; GB) used by different imputation methods to perform required calculations. RAM benchmarking for DeepCpG has not been included due to DeepCpG requiring a different machine with GPUs.

- Compared to existing imputation tools, GIMMEcpg was a lot faster (**Figure 5A**)
- RAM usage of GIMMEcpg was also lower than other imputation methods (**Figure 5B)**
- The reduced run time allows GIMMEcpg to scale to large WBGS datasets
  - We tested this scalability on a subset of available IHEC datasets (497 files)
  - GIMMEcpg completed imputation for all 497 files (~376 billion data points) in under 10 hours

Discussion

- Compared to existing machine-learning based imputation tools, GIMMEcpg performed with similar accuracy but with a marked reduction in computation time
  - Suggests that machine learning is not always the superior choice over simpler methods
- Benchmarking highlighted the inability of BoostME to impute sparse data (**Figure 4B**), where imputation is arguably needed the most
  - Unlike BoostME, GIMMEcpg did not have the same issue
- Run times of GIMMEcpg is poised to reduce even further as Polars announced their partnership with NVIDIA, bringing GPU acceleration to future versions

References

1 Jin, Z. *et al. Genes & Diseases* 5, 1–8 (2018), **2** Angermueller, C. *et al. Genome Biol* 18, 67 (2017), **3** Taudt, A. *et al. BMC Genomics* 19, 444 (2018), **4** The McDonnell Genome Institute *et al. BMC Genomics* 19, 390 (2018), **5** Eckhardt, F. *et al. Nat Genet* 38, 1378–1385 (2006), **6** Vink, R. *et al.* (2024), **7** LeDell, E. *et al. 7th ICML Workshop on Automated Machine Learning (AutoML)* (2020

niuzheng.chai.21@ucl.ac.uk

https://github.com/ucl-medical-genomics