Bioinformatics Advances, 2026, **6**(1), vbag146
https://doi.org/10.1093/bioadv/vbag146
Published: 26 May 2026
Application Note | Software

OXFORD

Pylluminator: fast and scalable analysis of DNA methylation data in Python

Elio Fanchon[1], Benjamin Loire [1,2], Jean-Philippe Trani[1], Frédérique Magdinier[1], Anaïs Baudot[1,3,*],

[1]Aix-Marseille Univ, INSERM, MMG, Marseille, France
[2]Neurology Therapeutic Area, R&D Servier Paris-Saclay Institut, Gif-sur-Yvette, France
[3]CNRS, Marseille, France
*Corresponding author. Anaïs Baudot, Aix-Marseille Univ, INSERM, MMG, Marseille, France; CNRS, Marseille, France. E-mail: anais.baudot@univ-amu.fr
Associate Editor: Mariike Kuijjer

Abstract

Motivation: **Illumina Infinium BeadChip technology for DNA methylation analysis continues to expand, with the latest EPICv2 arrays targeting about a million loci. As data volumes from this technology continue to grow, there is increasing demand for more scalable data processing solutions. Meanwhile, Python has gained significant interest in bioinformatics for its efficiency, versatility, and widespread use in data science and machine learning. Yet, no comprehensive Python toolkit exists for Illumina methylation array analysis.**

Results: **We present Pylluminator, a Python implementation of essential analysis methods including pre-processing tools, quality control, differential methylation analysis, and visualizations. Based on the established R packages SeSAMe and ChAMP, Pylluminator provides a scalable, user-friendly toolkit for DNA methylation analysis.**

Availability and implementation: **Pylluminator is an open-source package under MIT license available at https://github.com/eliopato/pylluminator. It was developed using Python 3.12 and can be installed with pip and uv. The documentation with thorough installation instructions and examples can be found at https://pylluminator.readthedocs.io**

1 Introduction

Epigenetics, a rapidly expanding field in biology, refers to the study of heritable changes able to modify the genome activity without alteration of the DNA sequence itself. Epigenetics involves chemical modifications of the DNA molecule, modifications of histone proteins around which DNA is wrapped to form chromatin, and the regulatory action of non-coding RNAs (Li 2021).

DNA methylation, one of the key epigenetic chemical modifications of DNA, is a powerful genome regulation mechanism. It plays a causative role in the development of various pathologies, including cancer (Sharma *et al.* 2010), autoimmune disorders, neurological disorders or rare genetic diseases such as ICF (Immunodeficiency, centromeric instability and facial anomalies), or Tatton-Brown-Rahman syndromes among others (Moosavi and Ardekani 2016). DNA methylation occurs when a methyl group is added to the DNA molecule on an adenine or cytosine base (Ratel *et al.* 2006). In mammals, DNA methylation mainly occurs on the cytosines of CpG dinucleotides and the majority of CpG dinucleotides (60%–90%) are methylated. CpG-rich regions called 'CpG islands', which are frequently found in gene promoter regions, show variable methylation patterns (Bird 1986). Methylation of CpG islands usually contribute to gene silencing. Methylation of repetitive DNA sequences also contribute to the maintenance of genome integrity.

Illumina Infinium Methylation BeadChips offer high-resolution tools to quantify genome-wide DNA methylation (Bibikova *et al.* 2006). Various bioinformatics packages have been developed over the years to process and analyze Illumina Infinium BeadChips data: ChAMP (Tian *et al.* 2017), SeSAMe (Triche *et al.* 2013) or Minfi (Aryee *et al.* 2014) in R, or Methylsuite [methylprep], CpGTools (Wei *et al.* 2021) or Mepylome (Brugger *et al.* 2026) in Python. Unfortunately, the R packages are being challenged by the increasing number of probes and samples in experiments. For instance, the latest Illumina Infinium BeadChips version for the human genome, EPICv2, targets over 930.000 loci, of which 99.5% are CpG sites. The Python packages are either not maintained (Methylsuite) or lacking pre-processing and analysis features (CpGTools, Mepylome).

With the exponential growth of interest in data science over the past decade, Python became one of the most used languages to handle large-scale datasets efficiently. An increasing number of bioinformatics tools are being developed or translated to Python, but there is no up-to-date tool available to fully process, analyze, and visualize Illumina methylation array data.

Received: 19 March 2026. **Revised:** 11 May 2026. **Accepted:** 18 May 2026
© The Author(s) 2026. Published by Oxford University Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose to fill the gap with Pylluminator, which implements the most common pre-processing methods, analysis algorithms, and data visualizations of DNA methylation (Fig. 1). Pylluminator is based on the reference R packages SeSAMe and ChAMP and provides a user-friendly, scalable, and reproducible toolkit for methylation data analysis, bridging the gap between bioinformatics and data science. Where a function also exists in SeSAMe, it is designed and tested to match its output and replicate its options.

2 Input data

Pylluminator handles all human (27k, 450k, MSA, EPIC, EPIC+, EPICv2), mammalian (Mammal40) and mouse (MM285) array versions, and supports different genome versions (hg38 and hg19 for human arrays, mm39 and mm10 for mouse arrays).

The annotations corresponding to each array, which we have created from SeSAMe and Illumina annotation files, are automatically downloaded from https://github.com/eliopato/pyllumination-data when needed—but custom annotations can also be provided by the users. The annotation files include the manifest file that maps each probe's Illumina ID to its metadata (e.g., color channel, location on the chromosome, probe type).

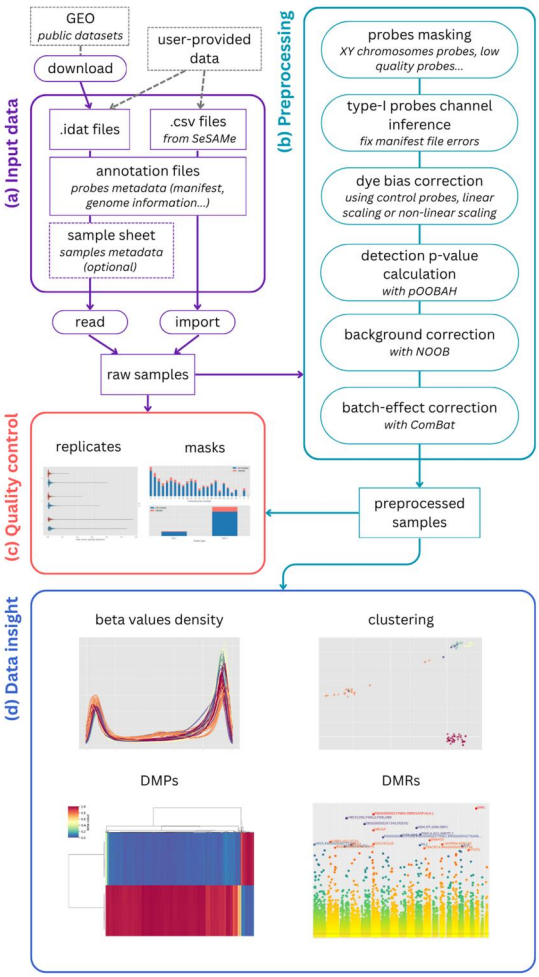

Figure 1 Flowchart of the main features of Pylluminator.

The annotation files also include genome information files for data enrichment and visualizations (e.g., transcripts description, chromosome regions information).

As described in Fig. 1(a), the input methylation data can be raw.*idat* files, as well as an existing SigDf object exported from SeSAMe as a.*csv* file. The raw.*idat* files can come from user-provided experimental data, or can be automatically downloaded from Gene Expression Omnibus (GEO) database by providing the samples GSM ids.

3 Pre-processing

As recommended by SeSAMe, the pre-processing pipeline starts by masking unwanted probes (Fig. 1(b)). Unwanted probes can be control probes, SNP probes, all non-CG probes, non-unique probes, probes that are known to have design issues (as defined by Zhou *et al.* [2017]), XY chromosome probes, as well as user-defined masks. The masking framework in Pylluminator makes it easy to exclude probes from the study, and masks are reversible at any time in the pipeline.

The second pre-processing step aims to infer the channel of type I probes. Type I and type II probes differ in design: type I probes use separate probes for methylated and unmethylated states, each measured in a distinct color channel, whereas type II probes use a single probe to measure both states in the same channel. As detailed in the 'Input Data' section, probes metadata, including color channel for type I probes, are provided by the manifest files. Running the Type-I channel inference method before any further processing allows to correct potential errors in the annotation of type I probes.

Then, dye bias, a well-known issue in multichannel microarray experiments that is due to a difference in fluorescence intensity between the green (Cy3) and red (Cy5) channels, can be corrected with linear or nonlinear scaling.

Next, detection *P*value are commonly used to assess the likelihood that a given probe signal corresponds to background fluorescence. We chose to integrate pOOBAH (*P*value with Out-Of-Band Array Hybridization) (Zhou *et al.* 2018), a method that calculates detection *P*value based on the empirical cumulative distribution function of overall Type-I out-of-band signal. Probes with detection *P*value higher than a certain threshold are then masked (by default, 0.05). Zhou *et al.* (2018) also show that masking these probes can highly reduce technical variations, such as batch effect (Leek *et al.* 2010).

For overall background signal correction, Pylluminator implements NOOB, the background subtraction algorithm based on Normal-exponential deconvolution using Out-Of-Band probes, as defined by Ritchie *et al.* (2007) and Triche *et al.* (2013).

If a batch effect remains, Pylluminator also integrates the most widely used algorithm for correcting technical biases in microarray data, ComBat (Johnson *et al.* 2007). ComBat is based on an empirical Bayes framework, and handles low sample sizes and more than two batches at the same time.

4 Samples quality control

Pylluminator includes comprehensive quality control (QC) tools to identify potential issues in methylation data. Users can

evaluate key metrics such as probe detection rates by probe type (CG, CH, SNP), average signal intensity by probe design type (I or II) and channel (red or green), type I probes color channel balance, and beta values distribution by probe type. These tools can be used to assess the quality of raw input data and the effectiveness of the pre-processing pipeline.

To further diagnose technical variability, the replicate analysis plot helps detect batch effects by comparing technical replicates. Additionally, visualizing the distribution of masked probes per chromosome and by probe design type provides valuable insights into experiment quality (Fig. 1 (c)).

5 Data insights

Beta-values are calculated for each probe using SeSAMe's formula, defined as:

$$\beta_i = \frac{\max(y_{i,methy}, 1)}{\max(y_{i,methy} + y_{i,unmethy}, 2)} \quad (1)$$

where $y_{i,methy}$ and $y_{i,unmethy}$ are the intensities measured by the i^{th} methylated and unmethylated probes, respectively.

The beta values density plot function offers options to plot each sample separately, or to group samples based on metadata (e.g. controls vs patients, or per batch to control for batch effect). Beta values of probes linked to a specific gene can be visualized in a heatmap with hierarchical clustering of samples, alongside the genomic positions of the probes.

Pylluminator provides an interface to the scikit-learn (Pedregosa *et al.* 2011) dimension reduction library, offering 14 different methods for unsupervised clustering of the samples, including PCA, MDS and FA. In addition to computing embeddings, it also generates visualizations to facilitate the interpretation of sample clustering patterns (Fig. 1(d)). The dendrogram visualization also gives information on the similarity between samples.

To detect Differentially Methylated Probes (DMPs), we fit an Ordinary Least Squares (OLS) linear regression model to each probe's methylation values. By default, the model uses the M-values (logit-transformed beta values) as the dependent variable, as they are more statistically appropriate than beta values for linear regression due to their homoscedasticity and unbounded range. However, we provided the option to use beta-values, for consistency with SeSAMe. Any metadata extracted from the sample sheet (e.g., age, sex, phenotype) can be used as predictor in the formula to describe the model. The resulting *P*value and effect size are then used to identify the significant DMPs, after correcting for multiple testing using the Benjamini-Hochberg procedure (Benjamini and Hochberg 1995). If the experimental design includes technical replicates, repeated measures, or any other source of random variability, a Linear Mixed Model (LMM) is used instead of OLS. In the LMM, predictors of direct interest (e.g., age, sex, phenotype) are modeled as fixed effects, while sources of random variability (e.g., patient ID to account for repeated measures, batch ID) are modeled as random effects. This accounts for within-group correlations and improves the accuracy of inference. The identified DMPs can be visualized in a heatmap, with flexible options to select probes based on effect size or *P*value, to display sample metadata, and to incorporate hierarchical clustering for both samples and probes (Fig. 1(d)).

The method implemented to compute the Differentially Methylated Regions (DMRs) from DMPs is the same as in SeSAMe. Neighboring probes with similar methylation patterns for a given predictor contrast (e.g., treated vs. control) are grouped. Similarity is calculated based on the Euclidean distance between probes' beta values. The distance cutoff used to separate the regions can be explicitly specified, or automatically set to the value at a specified quantile of all pairwise distances (by default, 0.5, i.e., the median). The significance of each DMR is then computed by combining the *P*value of its constituent DMPs with Stouffer's *Z*-score method (Stouffer *et al.* 1949). A Manhattan plot annotated with gene information can be generated from the DMRs (Fig. 1(d)).

The Copy Number Variation (CNV) calculation method implemented in Pylluminator relies on copy-number-normal samples to normalize probe signal intensity. By default, Pylluminator provides precomputed normal reference data for EPIC/hg38 and EPICv2/hg38 array versions, similar to SeSAMe's implementation. For other array versions, users must provide their own reference data. Regardless of the array version, it is strongly recommended to use normal samples that closely match the

Table 1 Comparison of average execution time ± standard deviation (in seconds, unless otherwise stated) of the main functions.

Function	SeSAMe v1.24.0	ChAMP v2.36.0	Pylluminator v1.0
Read samples	51.42±3.66	40.40±1.39[a]	16.79±1.10
Beta values calculation	5.96±1.68		0.07±0.00
Non-linear dye bias correction	21.62±1.05	–	5.78±0.24
pOOBAH	12.14±0.73	–	2.04±0.07
NOOB	19.92±3.43	–	2.51±0.11
ComBat	–	18.67±0.56	5.37±0.16
DMPs	1130.52±38.62	113.76±2.35	127.43±1.55
DMRs	21.53±0.48	61.52±1.75	25.00±0.31
CNV on all samples[b]	–	49*min*	13.99±0.45
CNV + CNS on one sample[c]	9.29±0.41	–	6.35±0.40

[a] The champ.load() function includes the beta values calculation.
[b] We use here ChAMP's default setting which is Copy Number Variation (CNV) calculation on all samples, without Copy Number Segmentation (CNS).
[c] We use here SeSAMe's default setting which is CNV calculation on one sample, followed by Copy Number Segmentation (CNS).

biological and technical characteristics of the target samples (e.g., same tissue type, platform, and processing batch) to ensure accurate normalization.

For Copy Number Segmentation (CNS), Pylluminator employs the Circular Binary Segmentation (CBS) algorithm to identify genomic regions with homogeneous CNV. This approach efficiently detects breakpoints and segments the genome into intervals of consistent copy number variation. The resulting segments and CNV values can be visualized in a Manhattan plot.

Pathway analysis methods provide valuable insights into the biological processes affected by differentially expressed genes (DEGs). To facilitate this, pylluminator offers a streamlined interface to the GSEApy package, enabling both Over-Representation Analysis (ORA) and Gene Set Enrichment Analysis (GSEA).

6 Results

We compared the major data processing and analysis functions of Pylluminator with their SeSAMe or ChAMP equivalents on a dataset of 24 EPIC samples from Laberthonnière *et al.* (2023). Computation time is evaluated by running the functions 10 times, on a laptop with 32 Gb RAM and an Intel Core i7 processor with 14 cores. Batch correction is applied using Sentrix ID as batch identifier, and samples' phenotype as covariate. We compute the DMPs using the samples phenotype predictor (Control/FSHD1/FSDH2/BAMS). Results are presented in Table 1.

Acknowledgments

Pylluminator exists thanks to the growing Python bioinformatics community. We would like to thank the developers of the packages used in Pylluminator: methylprep for the.idat files parser, inMoose (Behdenna *et al.* 2023) for the implementation of the ComBat algorithm, and pyRanges (Stovner and Sætrom 2020) for handling GenomicRanges. We also thank the developers of the R packages used as reference, namely SeSAMe (Triche *et al.* 2013) and ChAMP (Tian *et al.* 2017). Last but not least, we thank David Hirst and Youssef Boueimen for testing Pylluminator thoroughly and giving valuable feedbacks.

Conflicts of interest

BL is employed by Servier and declares no competing interests in relationship with this manuscript. AB is Associate Editor of Bioinformatics Advances but was not involved in the editorial process of this manuscript. All remaining authors declare no competing interests.

Funding

This work was supported by the European Rare Diseases Research Alliance (ERDERA, European Union's Horizon Europe research and innovation programme under grant agreement [N°101156595]); the French National Research Agency through the CAMUDI project [ANR-21-CE45-0001]; and the French government under the France 2030 investment plan, as part of the Initiative d'Excellence d'Aix-Marseille Université - A*MIDEX [AMX-19-IET-007] through the Marseille Maladies Rares Institute (MarMaRa).

Data availability

The data used in this article for the speed tests are available in the NCBI Gene Expression Omnibus (https://www.ncbi.nlm.nih.gov/geo/) with the access number GSE175527. Pylluminator's code and data are hosted on GitHub and are publicly available at https://github.com/eliopato/pylluminator and https://github.com/eliopato/pylluminator-data.

Code reliability

The code follows PEP8 standards and comes with extensive documentation, making it easy for external contributors to extend Pylluminator's functionalities.

To ensure consistency and reliability, 93% of the code is covered by unit tests that systematically check for regressions or unwanted changes, and compare the results of Pylluminator with the outputs of reference tools such as SeSAMe (Triche *et al.* 2013) and ChAMP (Tian *et al.* 2017).

As Pylluminator is implemented as a fully independent and standalone tool, updates to external packages will not affect its functionality. New functions can be readily integrated thanks to this modularity and the existing testing infrastructure.

References

Aryee M, Jaffe A, Corrada-Bravo H *et al.* Minfi: a flexible and comprehensive bioconductor package for the analysis of infinium DNA methylation microarrays. *Bioinformatics* 2014;**30**:1363–9.

Behdenna A, Colange M, Haziza J *et al.* Pycombat, a python tool for batch effects correction in high-throughput molecular data using empirical bayes methods. *BMC Bioinformatics* 2023;**24**:459.

Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approch to multiple testing. *J R Stat Soc. Ser B (Methodol)* 1995;**57**:289–300.

Bibikova M, Lin Z, Zhou L *et al.* High-throughput DNA methylation profiling using universal bead arrays. *Genome Res* 2006;**16**:383–93.

Bird A. CpG-rich islands and the function of DNA methylation. *Nature* 1986;**321**:209–13.

Brugger J, Hultschig C, Baumhoer D, *et al.* Mepylome: a point-of-care tumor diagnostic toolkit for tumor DNA methylation and copy number analysis. *Adv Intel Syst* 2026;**8**:e202500778. https://gseapy.readthedocs.io/en/latest/

Johnson W, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics* 2007;**8**:118–27.

Laberthonnière C, Delourme M, Chevalier R *et al.* In skeletal muscle and neural crest cells, smchd1 regulates biological pathways relevant for bosma syndrome and facioscapulohumeral dystrophy phenotype. *Nucleic Acids Res* 2023;**51**:7269–87.

Leek JT, Scharpf RB, Bravo HC *et al.* Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat Rev Genet* 2010;**11**:733–9.

Li Y. Modern epigenetics methods in biological research. *Methods* 2021;**187**:104–13. methylprep.https://github.com/FoxoTech/methylprep.

Moosavi A, Ardekani A. Role of epigenetics in biology and human diseases. *Iran Biomed J* 2016;**20**:246–58.

Pedregosa F, Varoquaux G, Gramfort A *et al.* Scikit-learn: machine learning in python. *J Mach Learn Res* 2011;**12**:2825–30.

Ratel D, Ravanat J, Berger F *et al.* N6-methyladenine: the other methylated base of DNA. *Bioessays* 2006;**28**:309–15.

Ritchie M, Silver J, Oshlack A *et al.* A comparison of background correction methods for two-colour microarrays. *Bioinformatics* 2007;**23**:2700–7.

Sharma S, Kelly T, Jones P. Epigenetics in cancer. *Carcinogenesis* 2010;**31**:27–36.

Stouffer S, Suchman E, DeVinney L *et al. Adjustment during Army Life*. Princeton, NJ: Princeton University Press, 1949.

Stovner E, Sætrom P. Pyranges: efficient comparison of genomic intervals in Python. *Bioinformatics* 2020;**36**:918–9.

Tian Y, Morris T, Webster A *et al.* Champ: updated methylation analysis pipeline for illumina beadchips. *Bioinformatics* 2017;**33**:3982–4.

Triche T, Weisenberger D, Van Den Berg D *et al.* Low-level processing of illumina infinium DNA methylation beadarrays. *Nucleic Acids Res* 2013;**41**:e90.

Wei T, Nie J, Larson N *et al.* Cpgtools: a Python package for DNA methylation analysis. *Bioinformatics* 2021; **12**:1598–9.

Zhou W, Laird P, Shen H. Comprehensive characterization, annotation and innovative use of infinium DNA methylation beadchip probes. *Nucleic Acids Res* 2017;**45**:e22.

Zhou W, Triche T, Laird P *et al.* Sesame: reducing artifactual detection of DNA methylation by infinium beadchips in genomic deletions. *Nucleic Acids Res* 2018;**46**:e123.