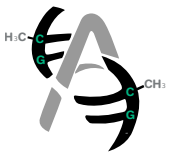
AutoMethyc



Documentation

AutoMethyc is a practical integrative analysis of methylation data from massive parallel bisulfite sequencing optimized for performance in massive data analysis.

1 Installation

1.1 docker

We created a docker container with all the necessary dependencies to run the program in order to provide a portable and self-sufficient container. To install it, you need to have docker installed and then download the docker image.

```
docker pull ambrizbiotech/automethyc
```

Listing 1: Download docker container

Then clone the repository and move to $PATH the script: "automethyc_docker" for greater simplicity when running the docker container, being able to use absolute and relative paths.

```
git clone https://github.com/FerAmbriz/AutoMethyc.git && cd AutoMethyc/scr
sudo mv automethyc_docker /usr/bin/
```

Listing 2: Moving docker container automount script AutoMethyc

1.2 Local installation

Local installation requires installing all dependencies in $PATH

Dependencies

- Bowtie2 v2.4.5
- Samtools v1.15.1-12
- Bismark v0.23.0
- python v3.10.6
  - pandas v1.5.2
  - numpy v1.23.1
  - plotly v5.10.0
  - plotly-express v0.4.1
  - scikit-learn v1.1.2
  - tqdm v4.64.1
  - IPython v8.4.0
  - pysam v0.19.1
- fastqc v0.11.9
- TrimGalore v0.6.6
- figlet v2.2.5
- multiqc v1.13
- git v2.34.1
- wget v1.21.2
- curl v7.81.0
- UnZip v6.0
- cutadapt v3.5
- java v11.0.18
- gatk v4.3.0.0
- R v4.1.2
  - gsalib v2.2.1
  - ggplot2 v3.4.2
  - reshape v0.8.9
  - gqplots v3.1.3
  - tidyverse v2.0.0
  - pROC v1.18.5
  - combiROC v 0.3.4
- revelio

And then move the files from the scr folder to the $PATH

```
git clone https://github.com/FerAmbriz/AutoMethyc.git && cd AutoMethyc/scr
sudo mv * /usr/bin/
```

Listing 3: Moving the scripts

2 Usage

We provide a series of default values for simplicity when running with a single command where the only mandatory parameters are the directory path where all the files with FASTQ (*.f*), the genome reference file and the output directory.

```
automethyc -i [fastq_folder] -o [Output_folder] -r [reference genome file] [optional arguments]
```

Listing 4: Running automethyc

On the other hand, greater flexibility is offered when running the program by establishing default parameters that can be modified by the user.

```
-t --threads       # Number of threads (default=4)
-n --normal        # Folder with fastq of normals (default=False)
-g --genome        # Genome used for request in UCSC (default=hg19)
-b --bed           # File with regions of interest (default=False)
-d --depth         # Minimum depth to consider (default=20)
-q --quality       # Minimum quality (default=30)
-c --combinations # Number of outliers considered to combinations in the evaluation for logistic
                   # regression (default=10)
-rb --run_background     # Run on background
--read             # Read type in fastq (default=Paired)
```

Listing 5: Optional arguments

In case you are using the version installed with docker, you have to mount the volume (-v) in the corresponding directory and run it in the background (-d) to avoid breaking the process in long execution times. For this, we provide an automount script with the possibility of using relative and absolute paths.

```
automethyc_docker -i [fastq_folder] -o [Output_folder] -r [reference genome file] [optional
    arguments]
```

Listing 6: Running automethyc in docker container

2.1 Format of bed file

The BED file must contain the regions of interest, to filter nonspecific sequencing products or regions of noninterest. The file format is comma separated values (CSV) with the chromosome, start and end, presenting different formats for greater versatility.

Chr	Start	End
chr10	89619506	89619580
chr11	22647545	22647849

Table 1: In range

Chr	Start	End
chr17	41277106	41277106
chr17	41277115	41277115

Table 2: Specific-site

Chr	Start	End	Gene
chr10	89619506	89619580	KLLN
chr11	22647545	22647849	FANCF

Table 3: With gene

3 Example usage

In this trial, we conducted a comprehensive analysis of 10 samples (5 cases and 5 controls) from these previously generated datasets. The raw fastq files for bioinformatic analysis are accessible at SRR25023301, SRR25023302, SRR25023303, SRR25023304, SRR25023305 for cases and SRR25023039, SRR25023040, SRR25023041, SRR25023042, SRR25023043 for controls [1].

```
git clone https://github.com/FerAmbriz/AutoMethycTest.git
cd AutoMethycTest && mkdir output
automethyc_docker -i cases -n controls -r [hg19_reference_genome_file] -b BedGraph331.csv -o
    output
```

Listing 7: Example usage

4 Output and interpretation

The output is organized in 4 folders (Bismark, CSV, HTML, VCF).

4.1 ID Assignment

For greater data cleanliness, the ID assignment will be the file name considering the above to '%_S*'. For example: if the original name of the file is: 'ISD202_S152_L001_R1_001.fastq.gz' its ID will be "ISD202".

4.2 Base call error probability

Base call error probability on logarithmic scale is calculated using phred score wich are found in: 'CSV/fastqc_raw_data.csv' using FASTQC.

$$Q = -10log_{10}P \tag{1}$$

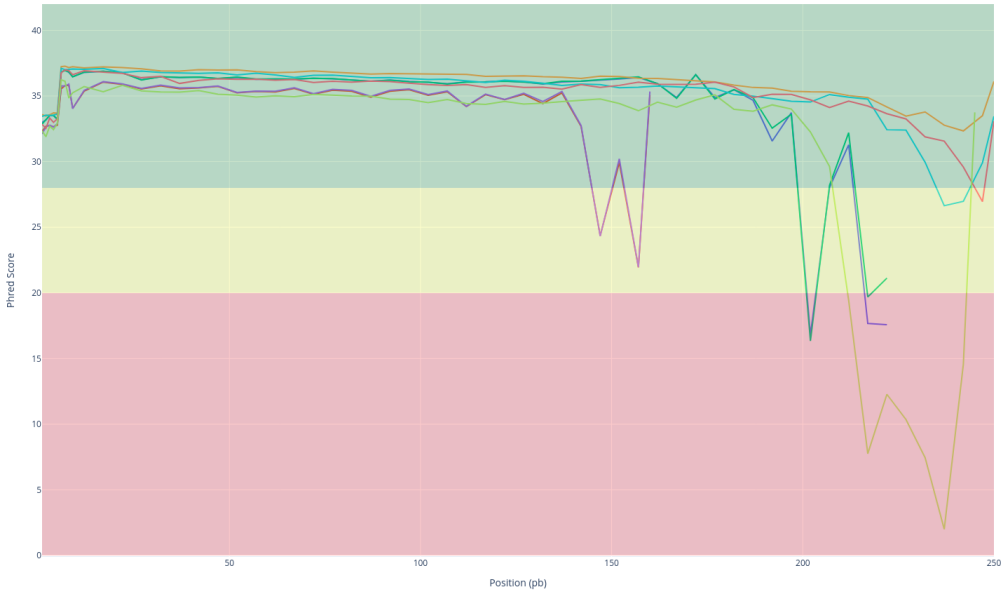


Figure 1: Quality score across all bases

To improve this and remove low quality sequences trim galore is used using a default Q>30. The output is provided in 'CSV/quality_trimming_metrics.csv'

4.3 Non conversion BS

In addition, an estimate of the conversion rate by Bisulfite is incorporated in 'CSV/non_conversion_metrics.csv', where the metrics show the equences removed because of apparent non-bisulfite conversion (at least 3 non-CG calls per read).

4.4 Alignment quality

To evaluate the alignment quality, information is extracted and compiled into a file to facilitate subsequent reading and analysis of alignment metrics, such as mapping efficiency, among others in the file 'CSV/quality_alignment_metrics.csv'

4.5 Depth

Additionally, an additional depth filter is added that discards sites with a depth less than established (by default >20 readings), where the metrics are compiled in 'CSV/count_depth_1_pass.csv'

4.6 Annotator

Regions unique to the raw_data will be annotated for their relationship to their corresponding gene or regions specified in the BED file using a request to UCSC genome browser [2].Therefore it is important to specify the genome used (default=hg19) with '-g'.

```
session = requests.Session()
params = {
        'hgsid': '1442153227_FWCo6wJtrFjEzVt07A5mEs5LeL3m',
        'db': genome,
        'hgta_group': 'genes',
        'hgta_track': 'refSeqComposite',
        'hgta_table': 'ncbiRefSeq',
        'hgta_regionType': 'genome',
        'hgta_outputType': 'primaryTable',
        'boolshad.sendToGalaxy': '0',
        'boolshad.sendToGreat': '0',
        'boolshad.sendToGenomeSpace': '0',
        'hgta_outFileName': '',
        'hgta_compressType': 'none',
        'hgta_doTopSubmit': 'get output'
    }
```

Listing 8: Request UCSC

The output will be a file in 'CSV/annotated_regions.csv' containing the annotated regions or in which case a BED file has been provided with the specified gene it will simply save the BED file as well.

Chr	Start	End	Gene	Strand	AccessName
chr7	6048904	6048904	AIMP2	+	NM_0013266*
chr3	37034316	37034316	EPM2AIP1	-	NM_014805.4

Table 4: UCSC annotated regions

Chr	Start	End	Gene
chr10	89619506	89619580	KLLN
chr11	22647545	22647849	FANCF

Table 5: Considering the BED with genes

GenomicRanges for CpG Site Annotation and Functional Mapping
In genomic analyses, the `GenomicRanges` package from Bioconductor provides a robust infrastructure for representing and manipulating genomic intervals. These objects enable efficient operations such as overlap detection, filtering, and merging of genomic regions—critical steps in functional genomics workflows. In our algorithm, we utilize `GenomicRanges` to map experimentally identified CpG methylation sites to annotated genomic features (e.g., genes, exons, promoters) based on the hg19 reference genome. This integration facilitates the functional interpretation of methylation events in the context of known biological elements.

ID	Chr	Start	End	Symbol
100124536	chr17	65736786	65736917	SNORA38B
100126313	chr17	11985216	11985313	MIR744
100126356	chr17	29902430	29902540	MIR365B
100128288	chr17	8261371	8263859	LOC100128288
100128977	chr17	43920722	43972879	MAPT-AS1
9931	chr17	65066554	65241319	HELZ
9953	chr17	14204056	14249492	HS3ST3B1
9957	chr17	13399066	13505244	HS3ST3A1
996	chr17	45195311	45266665	CDC27
999	chr17	60019966	60124643	MED13

Table 6: Annotated genes on chromosome 17 (GRanges object)

Chr	Start	End	Symbol
chr17	41196312	41322240	BRCA1

Table 7: Overlapping gene on chromosome 17 (GRanges mapping)

4.7 Filter target

Once the previously mentioned 'CSV/raw_data' is obtained, it will be filtered by the regions specified in the BED file o and the corresponding gene of each site previously annotated in 'CSV/annotated_regions.csv' will be added and saved as: 'filtered_target.csv'

ID	Type	Chr	Start	End	Met_perc	Cyt_Met	Cyt_NoMet	Depth	Gene
ISD202	cases	chr3	37034307	37034307	100.0	2383	0	2383	MLH1
ISD202	cases	chr3	37034316	37034316	0.463548	11	2362	2373	MLH1

Table 8: Format of 'CSV/filtered_target.csv'

In addition, a total count of the sites is made after filtering (targets)

-	ID
ISD202	337
ISD203	283

Table 9: Format of 'CSV/count_targets.csv

4.8 CGI mapping

The CGI region mapping makes a request to the UCSC genome browser [2] and classifies each site according to distance from the nearest CpG island.

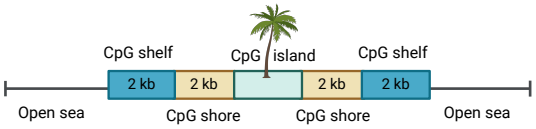


Figure 2: CpG island

The output of this mapping will be saved in: 'CSV/cgi_features.csv' with the information of the nearest CpG island and the mapped site.

#bin	chrom	chromStart	chromEnd	...	Site	DistCpGIsland	Type
1268	chr10	89621772	89624128	...	89619506	2266	CpG shelf
631	chr7	6048396	6049255	...	6048968	-	CpG island

Table 10: Format of 'CSV/cgi_features.csv'

4.9 Methylation percentage

To calculate the percentage of methylation, the conversion of the reference genome to bisulfite is carried out using Bismark[3], followed by the use of Trim galore, which automates quality control and trimming of the adapter using Fastqc, Trimmomatic [4] and Cutadapt [5]. The alignment to the reference genome is done with bowtie2[6] and samtools[7] to finally call the percentage of methylation. Subsequently, filtering by depth (default depth>20) is performed to reduce sequencing errors, which are collected for a data summary in 'CSV/count_depth_[depth (default=20)]_pass.csv'.

ID	unfiltered	filtered	depth_mean	depth_std
ISD202	672	347	572.08	723.23447
ISD203	490	225	709.924528	935.77306

Table 11: Format of 'CSV/count_depth_[depth (default=20)]_pass.csv

To simplify data analysis, we merge the COV files with the methylation percentages of each sample into a single file called: 'CSV/raw_data.csv', however, if you want to know more about the files generated in the 'Bismark' folder, we recommend reading their documentation.

ID	Type	Chr	Start	End	Met_perc	Cyt_Met	Cyt_NoMet	Depth
ISD202	cases	chr3	37034307	37034307	100.0	2383	0	2383
ISD202	cases	chr3	37034316	37034316	0.463548	11	2362	2373

Table 12: Format of 'CSV/raw_data.csv'

4.10 Matrix construction

From the filtered and annotated regions, a matrix of the regions is constructed to optimize the normalization of the data.

ID	-	-	ISD202	ISD203	ISD203
Type	-	-	controls	controls	cases
Chr	Start	Gene	-	-	-
chr10	89619506	KLLN	98.65	97.50	97.95
chr10	89619510	KLLN	98.92	97.19	99.18

Table 13: Format of 'CSV/matrix_filtered_target.csv'

Subsequently, the mean per gene is calculated in a matrix

Gene	ISD202	ISD203	ISD203
Type	controls	controls	cases
KLLN	96.76	96.66	98.65
ATM	0.29	0.10	0.85

Table 14: Format of 'CSV/matrix_mean_gene.csv'

4.11 Normalization

Normalization is calculated from the mean and standard deviation of the normals provided, following equation 2.

$$Z_{ij} = \frac{x_{ij} - \overline{x_j}}{S_j} \tag{2}$$

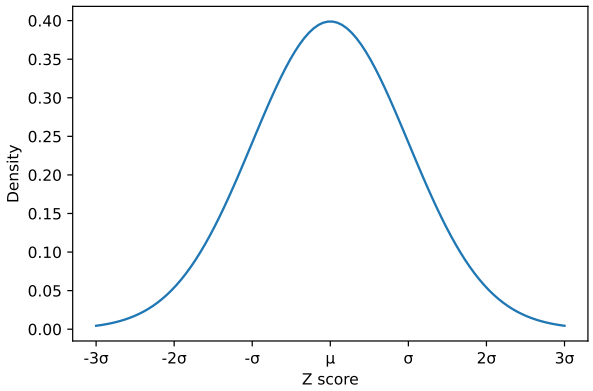


Figure 3: Normal distribution

The normalization output will be saved in: 'CSV/matrix_filtered_target_normalized.csv'

ID	Type	chr7:6048966	chr2:47596942	chr11:108093572
ISD202	controls	-0.707107	-0.539522	0.723362
ISD203	cases	0.478456	3.377785	-0.707107

Table 15: Format of 'CSV/matrix_filtered_target_normalized.csv'

However, the long format of the normalized matrix is also performed in:

ID	Type	variable	value
ISD202	controls	chr7:6048966	-0.707107
ISD203	cases	chr7:6048966	0.478456

Table 16: Format of 'CSV/filtered_target_normalized.csv'

Subsequently, the mean per gene is calculated in a matrix and the long format is also performed.

ID	Type	MSH2	BRIP1
ISD202	controls	-.707107	-0.707107
ISD203	cases	3.421513	3.421513

Table 17: 'CSV/matrix_mean_gene_normalized.csv'

ID	Type	variable	value
ISD202	controls	MSH2	0.707107
ISD203	cases	MSH2	3.421513

Table 18: 'CSV/mean_gene_normalized.csv'

4.12 PCA

To reduce the dimensionality of the data, we did an analysis of principal components, see the axes of greatest variation and see if there is a differential grouping between the samples and normals. The output is in 'CSV/pca_vectors.csv0

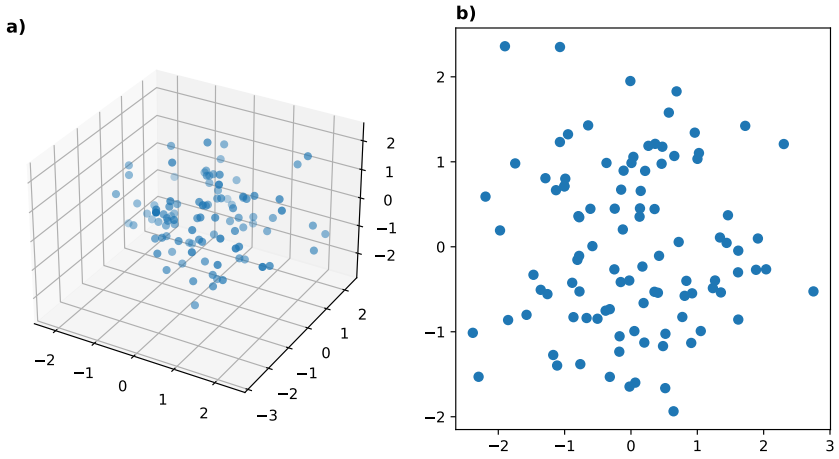


Figure 4: Dimensionality reduction by PCA

4.13 ROC

For Receiver Operating Characteristic (ROC) analysis, the best combination of sites that allows separation between controls and cases is identified in an unsupervised manner, where possible combinations between the sites with the highest number of outliers are performed, followed by the prediction evaluation using a logistic regression model. Finally, the ROC curve analysis is performed, evaluating the best combination.

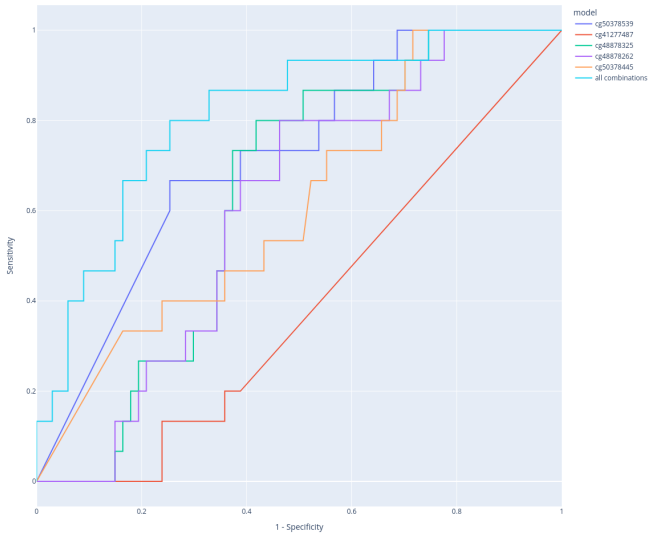


Figure 5: ROC curves of sites with better accuracy to classification

4.14 Variant calling

Regarding the variant calling, the bam generated with Bismark [3] is ordered with samtools[7], as well as the tags MD and NM are calculated and the bam index is created. Subsequently revelio [8] is used for bisulfite-influenced base masking and with samtools [7] it is added a read group for the variant calling with HaplotypeCaller [9]. The output will be laid out in 'VCF/*_mask_haplotype2.vcf', therefore, we recommend reading their official documentation for a correct interpretation and subsequent analysis.

4.15 Differential methylation

Differential methylation was made on the comparison of cases and controls, with a implementation of shapiro wilk test, and t-student or The Mann-Whitney U test in each site.

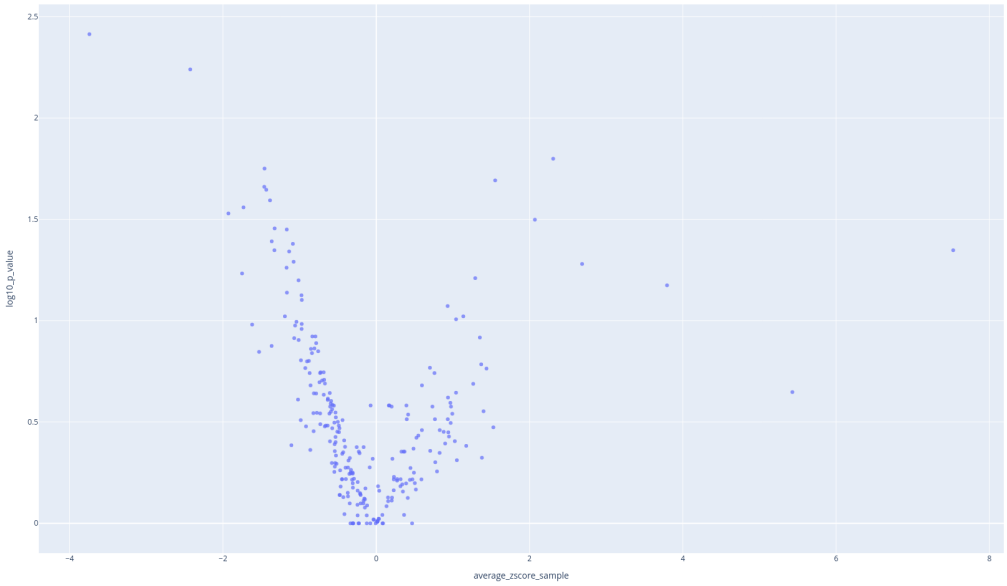


Figure 6: Differential methylation

4.16 HTML report

For greater ease in the interpretation and visualization of general data, we compile the information obtained in an interactive HTML report.

Figure 7: 'HTML/AutoMethyc_Report.html'

5 Step-by-Step Execution

To implement the process step-by-step, first create an output folder along with its subdirectories. Next, initiate the Bismark implementation, specifying the type of sample. If you have both cases and controls, run the implementation

twice to accommodate your requirements.

```
bismark_rounded $input $output $ref_folder $thr $quality $read_fastq cases
```

Listing 9: Bismark

Next, it filters out shallow sites in both cases and controls (optional).

```
filter_depth $output/Bismark/cases/bedGraph $output/Bismark/cases $depth cases
```

Listing 10: Depth

Finally, merge all the files into one.

```
bindcov $output/Bismark/cases/bedGraph $output/Bismark/cases 'cases'
```

Listing 11: Merge

To create the final HTML report, we extract the metrics from FastQC and Bismark and combine them into a single file.

```
fastqc_extract $output/Bismark/cases/fastq_trimmed $output/Bismark/cases
extract_statistics_alignment $output/Bismark/cases/fastq_trimmed $output/Bismark/cases/aligned
    $output/Bismark/cases/deduplicated cases $output/Bismark/cases
```

Listing 12: FastQC

Optionally, we run MultiQC to view the quality metrics in separate, more detailed reports. However, AutoMethyc already provides the main quality metrics in its report.

```

multiqc $output/Bismark/controls/fastq_trimmed/*
mv multiqc_report.html $output/HTML/multiqc_report_controls.html
```

Listing 13: MultiQC

If you ran the flow for the cases folder and then the controls, merge them into a single file and save it in the 'output/CSV' directory.

```
awk '(NR == 1) || (FNR > 1)' $output/Bismark/controls/raw_data.csv $output/Bismark/cases/raw_data.
    csv > $output/CSV/raw_data.csv

awk '(NR == 1) || (FNR > 1)' $output/Bismark/controls/count_depth_${depth}_pass.csv $output/
    Bismark/cases/count_depth_${depth}_pass.csv > $output/CSV/count_depth_${depth}_pass.csv

awk '(NR == 1) || (FNR > 1)' $output/Bismark/controls/fastqc_raw_data.csv $output/Bismark/cases/
    fastqc_raw_data.csv > $output/CSV/fastqc_raw_data.csv

awk '(NR == 1) || (FNR > 1)' $output/Bismark/controls/quality_trimming_metrics.csv $output/Bismark
    /cases/quality_trimming_metrics.csv > $output/CSV/quality_trimming_metrics.csv

awk '(NR == 1) || (FNR > 1)' $output/Bismark/controls/quality_alignment_metrics.csv $output/
    Bismark/cases/quality_alignment_metrics.csv > $output/CSV/quality_alignment_metrics.csv

awk '(NR == 1) || (FNR > 1)' $output/Bismark/controls/non_conversion_metrics.csv $output/Bismark/
    cases/non_conversion_metrics.csv > $output/CSV/non_conversion_metrics.csv

awk '(NR == 1) || (FNR > 1)' $output/Bismark/controls/duplicated_metrics.csv $output/Bismark/cases
    /duplicated_metrics.csv > $output/CSV/duplicated_metrics.csv
```

Listing 14: Merge with awk

Annotation is performed by querying the genomes available in the UCSC Genome Browser.

```
region_annotator $filtro $genome $output/CSV $thr
```

Listing 15: Annotation

Optionally, filter the regions of interest provided by the BED file.

```

filter_target $output/CSV/raw_data.csv $output/CSV/annotated_regions.csv $output/CSV
```

Listing 16: Filter target

To have greater control over the normalization process, matrices of the sites of interest are constructed and then unpivoted.

```

matrix_normalizer $output/CSV/matrix_filtered_target.csv $output/CSV/matrix_mean_gene.csv $output/
    CSV

make_vectors_pca $output/CSV/matrix_filtered_target_normalized.csv $output/CSV

unpivot_matrix_normalized $output/CSV/matrix_filtered_target_normalized.csv $output/CSV $output/
    CSV/matrix_mean_gene_normalized.csv
```

Listing 17: Normalization

For island classification, mapping is performed based on the CpG islands reported in the genomes available from the UCSC Genome Browser

```
cgi_mapping $output/CSV/matrix_filtered_target.csv $genome $output/CSV
```

Listing 18: CGI mapping

For multivariate analysis using PCA, vectors are extracted from the normalized data.

```
make_vectors_pca $output/CSV/matrix_filtered_target_normalized.csv $output/CSV
```

Listing 19: PCA

A differential expression analysis is then performed using a volcano plot.

```
volcano $output/CSV/filtered_target_normalized.csv $output/CSV/
```

Listing 20: Volcano

To identify the hypermethylated sites with the highest number of samples, an unsupervised analysis was conducted to evaluate the top 10 sites with the most hypermethylated samples. A comparative analysis of classification prediction using logistic regression was then performed. The combination with the highest accuracy in the validation test (defined by the 30% of data hidden from training) was subsequently selected for combined ROC analysis.

```
co_methylation $output/CSV/matrix_filtered_target_normalized.csv $output/CSV/
    filtered_target_normalized.csv $output/CSV/ $combinations

Rscript /usr/bin/combi_roc.R $output/CSV
```

Listing 21: Co methylation

For single nucleotide variation (SNV) analysis, the base is masked using Revelio, and the variants are called using HaplotypeCaller. The number of identified variants is then counted, and if controls are used, they are merged into a single file.

```
revelio_haplotype $output/Bismark/cases/aligned $ref $output/VCF/cases $thr

snv_count $output/VCF/cases $output/VCF/cases cases
awk '(NR == 1) || (FNR > 1)' $output/VCF/controls/snv_count.csv $output/VCF/cases/snv_count.csv >
    $output/CSV/snv_count.csv
```

Listing 22: Revelio and HaplotypeCaller

Finally, generate the HTML report, which provides an interactive summary of the entire analysis.

```
html_report $output $output/HTML True $depth
```

Listing 23: HTML generation

References

[1] Miguel Ruiz-De La Cruz et al. "Methylation marks in blood DNA reveal breast cancer risk in patients fulfilling hereditary disease criteria". In: *npj Precision Oncology* 8.1 (June 2024). ISSN: 2397-768X.

[2] Donna Karolchik et al. "The UCSC Table Browser data retrieval tool". In: *Nucleic acids research* 32.suppl_1 (2004), pp. D493–D496.

[3] Felix Krueger and Simon R Andrews. "Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications". In: *bioinformatics* 27.11 (2011), pp. 1571–1572.

[4] Anthony M Bolger, Marc Lohse, and Bjoern Usadel. "Trimmomatic: a flexible trimmer for Illumina sequence data". In: *Bioinformatics* 30.15 (2014), pp. 2114–2120.

[5] Marcel Martin. "Cutadapt removes adapter sequences from high-throughput sequencing reads". In: *EMBnet. journal* 17.1 (2011), pp. 10–12.

[6] Ben Langmead et al. "Scaling read aligners to hundreds of threads on general-purpose processors". In: *Bioinformatics* 35.3 (2019), pp. 421–432.

[7] Petr Danecek et al. "Twelve years of SAMtools and BCFtools". In: *Gigascience* 10.2 (2021), giab008.

[8] Adam Nunn et al. "Manipulating base quality scores enables variant calling from bisulfite sequencing alignments using conventional Bayesian approaches". In: *BMC genomics* 23.1 (2022), p. 477.

[9] Ryan Poplin et al. "Scaling accurate genetic variant discovery to tens of thousands of samples". In: *BioRxiv* (2017), p. 201178.