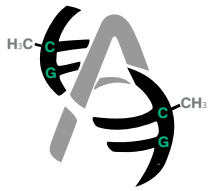
AutoMethyc

Documentation



AutoMethyc is an integrative pipeline to methylation analysis from raw sequences obtained from massive parallel bisulfite sequencing.

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.

Command 1: Download docker container

```
docker pull ambrizbiotech/automethyc
```

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

Command 2: Moving docker container automount script AutoMethyc

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

1.2 Local installation

Local installation requires installing all dependencies in the $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
- revelio

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

Command 3: Moving the scripts

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

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.

Command 4: Running automethyc

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

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

Command 5: Optional arguments

```
-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)
--read         # Read type in fastq (default=Paired)
```

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.

Command 6: Running automethyc in docker container

```
automethyc_docker -i [fastq_folder] -o [Output_folder] -r [reference genome file] [optional arguments]
# The output when executing this command is the "container ID" that will be running in the background.
    To see the execution progress use:
docker logs "container ID"
```

2.1 Format of bed file

The BED file must contain the regions of interest, in order to filter non-specific sequencing products or regions of non-interest. 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 Output and interpretation

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

3.1 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".

3.2 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.

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

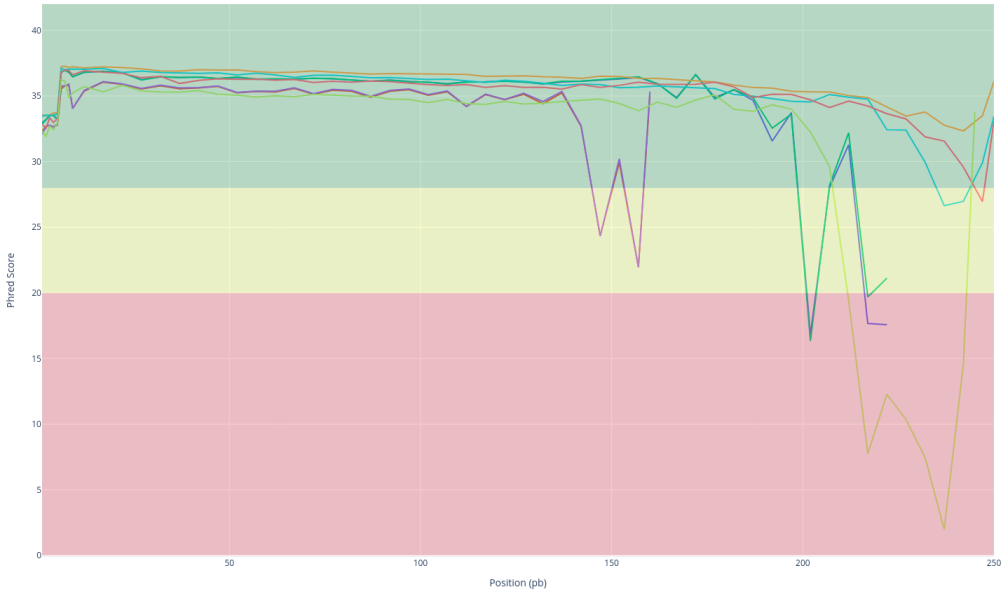


Figure 1: Quality score across all bases

3.3 Methylation percentage

To calculate the percentage of methylation, the conversion of the reference genome to bisulfite is performed using Bismark[1], followed by the use of Trim galore that automates the quality control and trimming of the adapter using Fastqc, Trimmomatic [2] and Cutadapt [3]. The alignment to the reference genome is done with bowtie2[4] and samtools[5] 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 4: 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	Sample	chr3	37034307	37034307	100.0	2383	0	2383
ISD202	Sample	chr3	37034316	37034316	0.463548	11	2362	2373

Table 5: Format of 'CSV/raw_data.csv'

3.4 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 [6].Therefore it is important to specify the genome used (default=hg19) with '-g'.

Command 7: Request UCSC

```
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'
    }
```

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 6: UCSC annotated regions

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

Table 7: Considering the BED with genes

3.5 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	Sample	chr3	37034307	37034307	100.0	2383	0	2383	MLH1
ISD202	Sample	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

3.6 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	-	-	Normal	Normal	Sample
Chr	Start	Gene	-	-	-
chr10	89619506	KLLN	98.65	97.50	97.95
chr10	89619510	KLLN	98.92	97.19	99.18

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

Subsequently, the mean per gene is calculated in a matrix

Gene	ISD202	ISD203	ISD203
Type	Normal	Normal	Sample
KLLN	96.76	96.66	98.65
ATM	0.29	0.10	0.85

Table 11: Format of 'CSV/matrix_mean_gene.csv'

3.7 CGI mapping

The CGI region mapping makes a request to the UCSC genome browser [6] and classifies each site according to distance from the nearest CpG island.

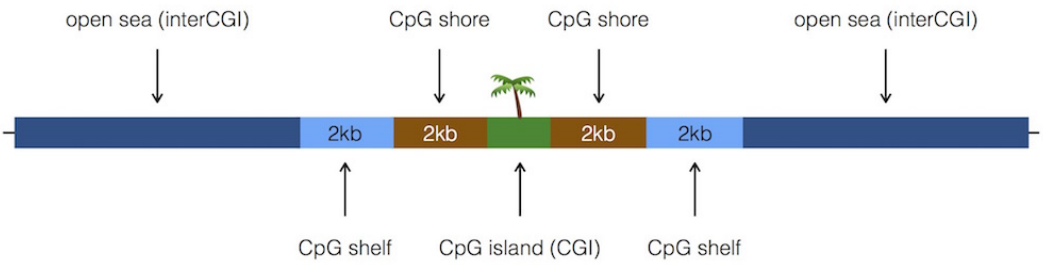


Figure 2: CpG island

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 12: Format of 'CSV/cgi_features.csv'

3.8 Normalization

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

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

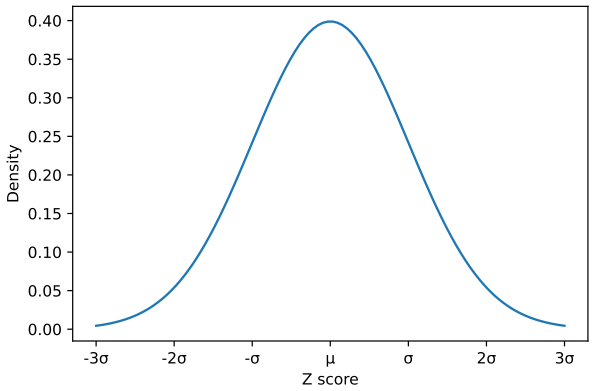


Figure 3: Normal distribution

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

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

Table 13: 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	Normal	chr7:6048966	-0.707107
ISD203	Sample	chr7:6048966	0.478456

Table 14: 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	Normal	-.707107	-0.707107
ISD203	Sample	3.421513	3.421513

Table 15: 'CSV/matrix_mean_gene_normalized.csv'

ID	Type	variable	value
ISD202	Normal	MSH2	0.707107
ISD203	Sample	MSH2	3.421513

Table 16: 'CSV/mean_gene_normalized.csv'

3.9 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.

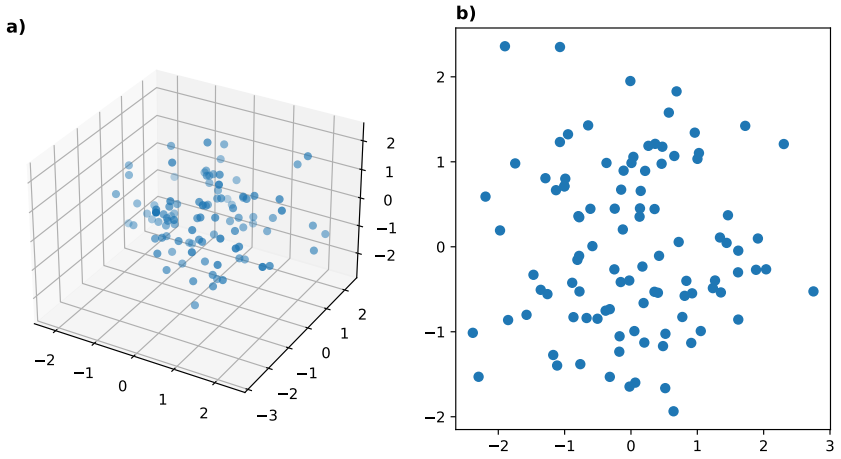


Figure 4: Dimensionality reduction by PCA

3.10 Variant calling in germline

Regarding the variant calling, the bam generated with Bismark [1] is ordered with samtools[5], as well as the tags MD and NM are calculated and the bam index is created. Subsequently revelio [7] is used for bisulfite-influenced base masking and with samtools [5] it is added a read group for the variant calling with HaplotypeCaller [8]. 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.

3.11 HTML report

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

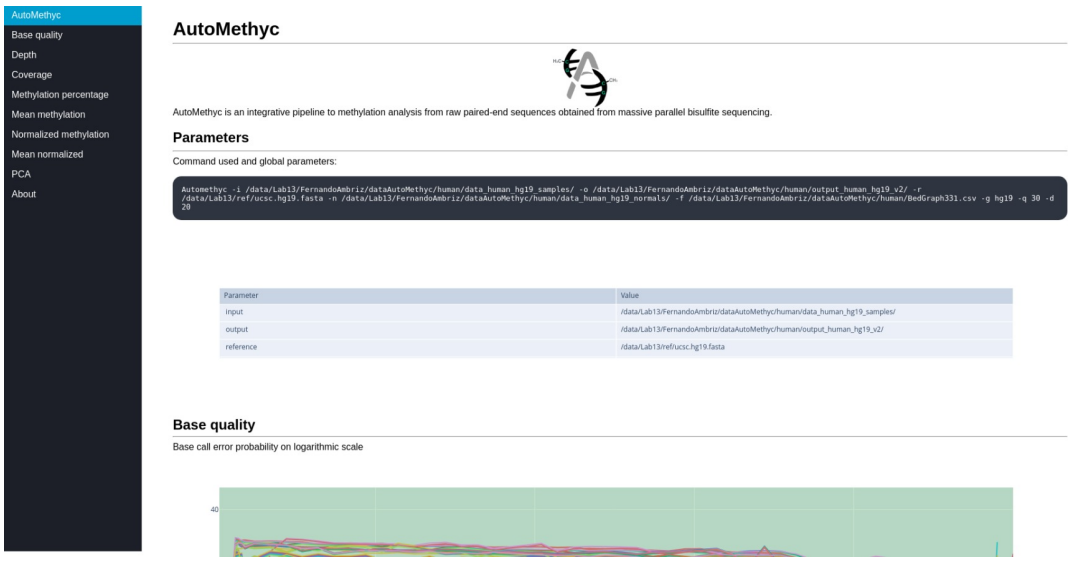


Figure 5: 'HTML/AutoMethyc_Report.html'

Figure 6: Output directory tree. The directories are shown in blue, while the files or folders that may or may not be normal depending on whether or not it is provided are in green. Finally, the rest of the files are shown as blank

References

[1] 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.

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

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

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

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

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

[7] 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.

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