

Package: ImprintCapASM (via r-universe)

July 1, 2026

Type Package

Title Allele-Specific Methylation Analysis for Imprinted DMR
Diagnostics

Version 0.1.1

Description Provides functions for SNP-phased allele-specific methylation (ASM) analysis across the 41 canonical human imprinted differentially methylated regions (DMRs). Reads are assigned to REF or ALT alleles based on bisulfite-aware SNP detection, enabling diagnosis of imprinting disorders from whole-genome bisulfite sequencing data.
<<https://github.com/19-saha/ImprintCapASM>>.

License MIT + file LICENSE

Encoding UTF-8

Imports data.table, readxl, writexl, vcfR, ggplot2, Rsamtools, tools

Suggests testthat (>= 3.0.0), knitr, rmarkdown

SystemRequirements samtools (>= 1.10)

VignetteBuilder knitr

URL <https://github.com/19-saha/ImprintCapASM>

BugReports <https://github.com/19-saha/ImprintCapASM/issues>

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libbz2-dev libicu-dev liblzma-dev xz-utils zlib1g-dev

Repository <https://19-saha.r-universe.dev>

Date/Publication 2026-06-24 17:22:14 UTC

RemoteUrl <https://github.com/19-saha/imprintcapasm>

RemoteRef HEAD

RemoteSha 0c05290e98b7828c5da63b99a1ef089bc7422a17

Contents

ASM	2
extract_bam_regions	3
prepare_cpg_snp_input	4
run_pipeline	6
Index	8

ASM

Run Allele-Specific Methylation Analysis

Description

Performs SNP-phased allele-specific methylation (ASM) analysis across the canonical human imprinted differentially methylated regions (DMRs). Reads are assigned to REF or ALT alleles based on bisulfite-aware SNP detection, and per-allele CpG methylation fractions are computed.

Usage

```
ASM(
  cpg_snp_file,
  sam_file,
  filter_cpgs_file,
  output_file = NULL,
  sample_type = c("control", "patient"),
  verbose = TRUE
)
```

Arguments

cpg_snp_file	Character. Path to the Excel output from prepare_cpg_snp_input .
sam_file	Character. Path to the extracted wide BAM file produced by extract_bam_regions .
filter_cpgs_file	Character. Path to the CpG filter/variability reference Excel file (filter_cpgs_ctrl.xlsx or filter_cpgs_pat.xlsx).
output_file	Character or NULL. Path for the ASM output .xlsx. Auto-named as asm_<sample_type>_<sample_id>.xlsx if NULL.
sample_type	Character. Either "control" or "patient".
verbose	Logical. If TRUE (default), progress messages are written via message(). Set to FALSE or wrap the call in suppressMessages() to silence all output.

Value

A named list with three elements:

asm Full read-level ASM results as a data table.
snp_cpg Per SNP x CpG summary table.
meth_summary Per CpG allele methylation summary table.

Examples

```
## Not run:
extdata      <- system.file("extdata", package = "ImprintCapASM")
cpg_snp_tmp  <- tempfile(fileext = ".xlsx")

prepare_cpg_snp_input(
  snp_file    = file.path(extdata, "example_snp.out"),
  meth_file   = file.path(extdata, "example_cgmeth.txt"),
  cpg_ref_file = file.path(extdata, "example_filter_cpgs.xlsx"),
  output_file = cpg_snp_tmp,
  sample_type = "control"
)

result <- ASM(
  cpg_snp_file = cpg_snp_tmp,
  sam_file     = file.path(extdata, "example_markdup.bam"),
  filter_cpgs_file = file.path(extdata, "example_filter_cpgs.xlsx"),
  output_file   = tempfile(fileext = ".xlsx"),
  sample_type   = "control"
)
head(result$snp_cpg)

## End(Not run)
```

extract_bam_regions *Extract BAM Regions Around SNP Windows*

Description

Calls samtools view and samtools sort to subset a whole-genome bisulfite BAM file to the genomic windows defined in a BED file, producing a smaller indexed BAM suitable for input to [ASM](#).

Usage

```
extract_bam_regions(
  bam_file,
  bed_file,
  output_dir = dirname(bam_file),
  overwrite = FALSE,
  sample_type = c("control", "patient"),
  verbose = TRUE
)
```

Arguments

bam_file Character. Path to the input _markdup.bam file (a .bai index must be present alongside it).

bed_file	Character. Path to the BED file produced by prepare_cpg_snp_input .
output_dir	Character. Directory for output files. Defaults to <code>dirname(bam_file)</code> .
overwrite	Logical. If FALSE (default), skips extraction when the output BAM already exists.
sample_type	Character. Either "control" or "patient".
verbose	Logical. If TRUE (default), progress messages are written via <code>message()</code> . Set to FALSE or wrap the call in <code>suppressMessages()</code> to silence all output.

Details

Requires `samtools` to be available on the system PATH. The function calls `samtools view -b`, `samtools sort`, and `samtools index` in sequence. If no `.bai` index is found alongside `bam_file`, [indexBam](#) is called automatically.

Value

The path to the output BAM file (returned invisibly).

Examples

```
if (nchar(Sys.which("samtools")) > 0L) {
  extdata <- system.file("extdata", package = "ImprintCapASM")

  out_bam <- extract_bam_regions(
    bam_file = file.path(extdata, "example_markdup.bam"),
    bed_file = file.path(extdata, "example_cpg_snp.bed"),
    output_dir = tempdir(),
    sample_type = "control"
  )
  file.exists(out_bam)
}
```

prepare_cpg_snp_input *Prepare CpG-SNP Input Table*

Description

Loads a bisulfite SNP `.out` file and a CpG methylation file, intersects heterozygous SNPs with the 41-DMR CpG panel, and returns a per-sample CpG-SNP pair table ready for input to [ASM](#).

Usage

```
prepare_cpg_snp_input(
  snp_file,
  meth_file,
  cpg_ref_file,
  output_file = NULL,
  min_depth = 20L,
  window_bp = 60L,
  sample_type = c("control", "patient"),
  verbose = TRUE
)
```

Arguments

snp_file	Character. Path to a VCF-like file ending in <code>_all.SNPs.out</code> .
meth_file	Character. Path to the gemBS/Bismark CpG methylation table (<code>.cov</code> or <code>.txt</code>).
cpg_ref_file	Character. Path to the CpG panel reference Excel file (<code>filter_cpgs_ctrl.xlsx</code> or <code>filter_cpgs_pat.xlsx</code>).
output_file	Character or NULL. Path for the output <code>.xlsx</code> . Auto-named as <code>cpg_snps_CG_<sample_type>_<sample_id>.xlsx</code> if NULL.
min_depth	Integer. Minimum total SNP read depth. Default 20L.
window_bp	Integer. Window in base pairs around each SNP used to search for CpG positions. Default 60L.
sample_type	Character. Either "control" or "patient".
verbose	Logical. If TRUE (default), progress messages are written via <code>message()</code> . Set to FALSE or wrap the call in <code>suppressMessages()</code> to silence all output.

Value

A data table of CpG-SNP pairs (returned invisibly). The `.xlsx` and `.bed` files are written as side effects.

Examples

```
extdata <- system.file("extdata", package = "ImprintCapASM")
cpg_snp_tmp <- tempfile(fileext = ".xlsx")

result <- prepare_cpg_snp_input(
  snp_file = file.path(extdata, "example_snp.out"),
  meth_file = file.path(extdata, "example_cgmeth.txt"),
  cpg_ref_file = file.path(extdata, "example_filter_cpgs.xlsx"),
  output_file = cpg_snp_tmp,
  sample_type = "control"
)
head(result)
```

run_pipeline

*Run the Full ImprintCapASM Pipeline***Description**

Convenience wrapper that executes the complete three-step pipeline: (1) [prepare_cpg_snp_input](#), (2) [extract_bam_regions](#), and (3) [ASM](#) for every sample in a directory. Controls and patients are always processed independently using their respective `filter_cpgs` reference files.

Usage

```
run_pipeline(
  bam_dir,
  snp_dir,
  meth_dir,
  cpg_ref_file,
  output_dir,
  sample_type = c("control", "patient"),
  bed_file,
  min_depth = 20L,
  window_bp = 60L,
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

<code>bam_dir</code>	Character. Path to the directory containing <code>.bam</code> files (and their <code>.bai</code> indices).
<code>snp_dir</code>	Character. Path to the directory containing SNP <code>.out</code> files produced by your bisulfite SNP caller.
<code>meth_dir</code>	Character. Path to the directory containing CpG methylation files (Bismark <code>.cov</code> or similar).
<code>cpg_ref_file</code>	Character. Path to the cohort-matched <code>filter_cpgs</code> reference <code>.xlsx</code> file. Use the file bundled in <code>inst/extdata/filter_cpgs_ctrl.xlsx</code> for controls and <code>inst/extdata/filter_cpgs_pat.xlsx</code> for patients.
<code>output_dir</code>	Character. Directory where all output files are written. Created automatically if it does not exist.
<code>sample_type</code>	Character. Either "control" or "patient". Must match the cohort of samples in <code>bam_dir</code> .
<code>bed_file</code>	Character. Path to the BED file defining the 41 DMR regions used by extract_bam_regions .
<code>min_depth</code>	Integer. Minimum read depth filter passed to prepare_cpg_snp_input . Default 20L.
<code>window_bp</code>	Integer. Window in base pairs around each CpG passed to prepare_cpg_snp_input . Default 60L.

overwrite	Logical. If TRUE, re-runs extraction even when output BAM files already exist. Default FALSE.
verbose	Logical. If TRUE, progress messages are printed to the console via message(). Default TRUE. Set to FALSE or wrap the call in suppressMessages() to silence all output.

Value

Invisibly returns a named list of ASM result objects, one per sample. Each element is the list returned by [ASM](#), containing \$asm, \$snp_cpg, and \$meth_summary.

Examples

```
## Not run:
extdata <- system.file("extdata", package = "ImprintCapASM")
cpg_ref <- file.path(extdata, "example_filter_cpgs.xlsx")
bed <- file.path(extdata, "example_cpg_snp.bed")
output_dir <- tempdir()

results <- run_pipeline(
  bam_dir = extdata,
  snp_dir = extdata,
  meth_dir = extdata,
  cpg_ref_file = cpg_ref,
  output_dir = output_dir,
  sample_type = "control",
  bed_file = bed
)
head(results[[1]]$asm)

## End(Not run)
```

Index

ASM, [2](#), [3](#), [4](#), [6](#), [7](#)

extract_bam_regions, [2](#), [3](#), [6](#)

indexBam, [4](#)

prepare_cpg_snp_input, [2](#), [4](#), [4](#), [6](#)

run_pipeline, [6](#)