

# Package ‘borealis’

June 22, 2025

**Type** Package

**Title** Bisulfite-seq Outlier mEthylation At singLe-sIte reSolution

**Version** 1.12.0

**Depends** R (>= 4.2.0), Biobase

**Imports** doParallel, snow, purrr, plyr, foreach, gamlss, gamlss.dist,  
bsseq, methods, DSS, R.utils, utils, stats, ggplot2, cowplot,  
dplyr, rlang, GenomicRanges

**Description** Borealis is an R library performing outlier analysis for  
count-based bisulfite sequencing data. It detects outlier  
methylated CpG sites from bisulfite sequencing  
(BS-seq). The core of Borealis is modeling Beta-Binomial distributions. This  
can be useful for rare disease diagnoses.

**License** GPL-3

**Encoding** UTF-8

**Suggests** BiocStyle, knitr, rmarkdown, RUnit, BiocGenerics, annotatr,  
tidyr, TxDb.Hsapiens.UCSC.hg19.knownGene, org.Hs.eg.db

**VignetteBuilder** knitr

**biocViews** Sequencing, Coverage, DNAMethylation,  
DifferentialMethylation

**git\_url** <https://git.bioconductor.org/packages/borealis>

**git\_branch** RELEASE\_3\_21

**git\_last\_commit** 35b22b9

**git\_last\_commit\_date** 2025-04-15

**Repository** Bioconductor 3.21

**Date/Publication** 2025-06-22

**Author** Garrett Jenkinson [aut, cre] (ORCID:  
<<https://orcid.org/0000-0003-2548-098X>>)

**Maintainer** Garrett Jenkinson <[gargar934@gmail.com](mailto:gargar934@gmail.com)>

Contents

borealis-package . . . . .	2
plotCpGsite . . . . .	2
runBorealis . . . . .	3
runSingleNewSample . . . . .	4
<b>Index</b>	<b>6</b>

---

borealis-package	<i>Bisulfite-seq Outlier mEthylation At singLe-sIte reSolution</i>
------------------	--

---

Description

Borealis is an R package performing outlier analysis for count-based bisulfite sequencing data. It detects outlier methylated CpG sites from bisulfite sequencing (BS-seq). The core of Borealis is modeling Beta-Binomial distributions. This can be useful for rare disease diagnoses.

Details

See packageDescription('borealis')

Author(s)

Maintainer: Garrett Jenkinson <gargar934@gmail.com>

---

plotCpGsite	<i>Generate a plot of the model and raw data at one or more CpG sites</i>
-------------	---

---

Description

Generate plots of model and results. The top panel of the plot will be the beta distribution in the beta-binomial model estimated for the cohort. The bottom panel will be the 95 percent confidence intervals around the percent methylation in each sample at that CpG site.

Usage

```
plotCpGsite(cpgSites, sampleOfInterest=NA, modelFile="CpG_model.csv",
            methCountFile="CpG_model_rawMethCount.tsv",
            totalCountFile="CpG_model_rawTotalCount.tsv")
```

**Arguments**

cpgSites	A character vector of CpG sites specified as "chr1:71732" representing the chromosome and start position of the CpG site. A separate plot will be generated for each site specified.
sampleOfInterest	(optional) character(1) Name of sample of interest which will be colored differently than the rest of the samples in the cohort. If NA then all samples will be plotted with same color.
modelFile	character(1) The mode file (including full path if not current working directory) with beta-binomial parameter estimates produced by runBorealis.
methCountFile	character(1) File name (including full path if not current working directory) for the methylated count file produced by runBorealis.
totalCountFile	character(1) File name (including full path if not current working directory) for the total count file produced by runBorealis.

**Value**

Returns a list with each element indexed by the provided cpgSites and storing a ggplot/cowplot object.

**Examples**

```
extdata <- system.file("extdata", package="borealis")
plots <- plotCpGsite("chr14:24780288",
  sampleOfInterest="patient_72",
  modelFile=file.path(extdata,"CpG_model_chr14.csv"),
  methCountFile=file.path(extdata,"CpG_model_rawMethCount_chr14.tsv"),
  totalCountFile=file.path(extdata,"CpG_model_rawTotalCount_chr14.tsv"))
```

---

runBorealis

*Run the full borealis pipeline*


---

**Description**

Run the full borealis pipeline. It will load in bismark data and save out to disk matrix-based methylation and total count files, then it will build the beta-binomial statistical models for the cohort at each CpG site and save the parameters of this model to disk, and finally provide outlier p-values and summary statistics for each sample in the cohort at each CpG site.

**Usage**

```
runBorealis(inDir,
  suffix = "_merged.cov.gz.CpG_report.merged_CpG_evidence.cov.gz",
  nThreads = 8, minDepth = 4, minSamps = 5, timeout = 10,
  laplaceSmooth = TRUE,
  chrs = c(paste0("chr",seq_len(22)), "chrX", "chrY"),
  outprefix = "borealis_", modelOutPrefix = "CpG_model")
```

**Arguments**

<code>inDir</code>	character(1) Directory path to bismark results. NOTE: this assumes following pattern for full paths to bismark coverage gz files: <code>\${inDir}/\${sampleName}/\${sampleName}\${suffix}</code> .
<code>suffix</code>	(optional) character(1) File suffix for the bismark coverage files.
<code>nThreads</code>	(optional) numeric(1) Number of compute threads to be used in multithreading computations.
<code>minDepth</code>	(optional) numeric(1) The minimum depth of coverage for sample to go into modeling.
<code>minSamps</code>	(optional) numeric(1) The minimum number of samples with <code>minDepth</code> coverage required to build a model at a given CpG site.
<code>timeout</code>	(optional) numeric(1) The maximum time in seconds to spend trying to build a model at a given CpG site (if it takes longer, we skip the site).
<code>laplaceSmooth</code>	(optional) logical(1) Whether or not to do Laplace (i.e., add one) smoothing on the counts.
<code>chrs</code>	(optional) A character vector listing the chromosomes to be loaded.
<code>outprefix</code>	(optional) character(1) The sample output file prefix (can include a full file path if current working directory is not desired output location).
<code>modelOutPrefix</code>	(optional) character(1) The cohort modeling output file prefix (can include a full file path if current working directory is not desired output location).

**Value**

Returns an object of "BSseq" class with raw dataset loaded and used for modeling purposes.

**Examples**

```
extdata <- system.file("extdata", "bismark", package="borealis")
outdir <- tempdir()
results <- runBorealis(extdata, nThreads=2, chrs="chr14", suffix=".gz",
                      outprefix = file.path(outdir, "borealis_"),
                      modelOutPrefix = file.path(outdir, "CpG_model"))
```

---

<code>runSingleNewSample</code>	<i>Run a single new sample after modeling complete</i>
---------------------------------	--

---

**Description**

Run a single new sample after modeling using `runBorealis` has already been completed in a cohort of samples. It will not rebuild the models and only predict using previously estimated model specified by `modelFile`.

**Usage**

```
runSingleNewSample(inFile, outFile, minObsDepth=10, modelFile="CpG_model.csv")
```

**Arguments**

<code>inFile</code>	character(1) File name (including full path if not current working directory) to the bismark coverage file.
<code>outFile</code>	character(1) File name (including full path if not current working directory) for the sample's modeling outputs. If NULL is provided, no outputs will be written to disk.
<code>minObsDepth</code>	(optional) numeric(1) Minimum depth of coverage in this sample for a modeling output/p-value to be produced at a given CpG.
<code>modelFile</code>	(optional) character(1) File name (including full path if not current working directory) for the model files (built by running <code>runBorealis</code> function).

**Value**

Returns a `GRanges` object with modeling results.

**Examples**

```
extdata <- system.file("extdata", package="borealis")
outdir <- tempdir()
gr <- runSingleNewSample(file.path(extdata, 'bismark', 'patient_72',
                                   'patient_72.gz'), file.path(outdir, 'output.txt'),
                        modelFile=file.path(extdata, 'CpG_model_chr14.csv'))
```

# Index

## \* **package**

borealis-package, [2](#)

borealis (borealis-package), [2](#)

borealis-package, [2](#)

plotCpGsite, [2](#)

runBorealis, [3](#)

runSingleNewSample, [4](#)