# WSH - R package for Within-Sample Heterogeneity Scores

Michael Scherer

January 8, 2020

## Contents



## 1 Introduction

This vignette desribes the functionalities included in the WSH R package, which includes the Within-Sample Heterogeneity Scores FDRP, qFDRP, PDR, MHL, Epipolymorphism and Entropy. The package is able to compute each of those scores from bisulfite sequencing data. Input should be a *bam* file that contains reads aligned to a reference genome, preferably the human reference genome version 'hg38'. While PDR, qFDRP and FDRP are independent of the employed mapping tool, Epipolymorphism and Entropy only support bismark as the aligner. In addition to the aligned reads, the user needs to specify annotation of the sites for which the scores should be computed in either of two ways: `GRanges` (`http://bioconductor.org/packages/release/bioc/html/GenomicRanges.html`) or `RnBSet` (`https://rnbeads.org/`). Here, we only discuss how to use the package. A detailed description of each of the scores can be found in the corresponding publications.

## 2 Installation

The package is available from GitHub and can be installed by the following command, given that the devtools package is installed:

```
> if(!requireNamespace("devtools")) install.packages("devtools",repos = "https://cloud.r-pro
> devtools::install_github("MPIIComputationalEpigenetics/WSHPackage")
```

You can test if the package was properly installed by executing one of the examples in the package:

```
> library(WSH)
> qfdrp <- wsh.run.example()
```

```
2020-01-08 18:20:39     1.1  STATUS STARTED WSH score example
2020-01-08 18:20:39     1.1  STATUS     STARTED Removing Sex chromosomes
2020-01-08 18:20:39     1.1  STATUS     COMPLETED Removing Sex chromosomes
2020-01-08 18:20:40     1.1  STATUS     STARTED qFDRP calculation
2020-01-08 18:21:23     1.1  STATUS     COMPLETED qFDRP calculation
2020-01-08 18:21:23     1.1  STATUS COMPLETED WSH score example
```

# 3 Computing WSH scores

## 3.1 FDRP, qFDRP and PDR

FDRP, qFDRP and PDR do not require any additional tools or scripts and can be computed directly from your *bam* file. In this case, the `score` argument of the `compute.score` function needs to be one of "fdrp", "qfdrp" or "pdr". You need to specify the CpG sites for which the scores should be computed in either of two forms: `GRanges` or `RnBSet`.

1. `GRanges`: This object should contain the positions of the CpGs for which analysis is to be conducted. The `GRanges` object should contain a single entry for each CpG and only have length 1 for each of the entries. Then you can either run `compute.score.GRanges` directly or call the generic function `compute.score`.

   ```
   > example.bam <- system.file(file.path("extData","small_example.bam"),
   +                            package="WSH")
   > example.GRanges <- GRanges(Rle(rep("chr2",10)),
   +                            IRanges(start=c(2298361,2298554,2298732,
   +                                            2298743,2298787,2298792,
   +                                            2298827,2298884,2298915,2298921),
   +                                    end=c(2298361,2298554,2298732,
   +                                          2298743,2298787,2298792,
   +                                          2298827,2298884,2298915,2298921)+1))
   > pdr <- compute.score(bam.file=example.bam,example.GRanges,score="pdr")
   ```

   ```
   2020-01-08 18:21:23     1.2  STATUS STARTED PDR calculation
   2020-01-08 18:21:23     1.2  STATUS     STARTED Removing Sex chromosomes
   2020-01-08 18:21:23     1.2  STATUS     COMPLETED Removing Sex chromosomes
   2020-01-08 18:21:35     1.2  STATUS COMPLETED PDR calculation
   ```

   This returns a *data.frame*, with the CpG positions (chromosome, start, end) in the first columns and the corresponding score in the last column.

   ```
   > dim(pdr)
   ```

   ```
   [1] 10  4
   ```

```
> head(pdr)

  chromosome   start     end       PDR
1       chr2 2298361 2298362       NaN
2       chr2 2298554 2298555       NaN
3       chr2 2298732 2298733 0.2413793
4       chr2 2298743 2298744 0.2372881
5       chr2 2298787 2298788 0.2337662
6       chr2 2298792 2298793 0.2337662
```

2. **RnBSet:** In addition to `GRanges` objects, the WSH package supports `RnBSet` objects as input. Here, the annotation is inferred from the object's annotation. Furthermore, a coverage filtering step is executed, which filters sites with coverage lower than *coverage.threhold* in the `RnBSet` object, given such information is present. For more details on how to set options for analysis, see subsection 4.1.

```
> example.rnb.set <- system.file(file.path("extData","small_rnbSet.zip"),
+                                package="WSH")
> example.rnb.set <- load.rnb.set(example.rnb.set)
> set.option(coverage.threshold = 10)
> fdrp <- rnb.calculate.fdrp(example.rnb.set,example.bam)
> to.plot <- data.frame(qFDRP=qfdrp$qFDRP,FDRP=fdrp$FDRP)
> to.plot <- melt(to.plot)
> plot <- ggplot(to.plot,aes(x=value,y=..count..,fill=variable))+
+   geom_histogram()+facet_grid(variable~.)+theme_bw()
> plot
```

## 3.2 MHL

In contrast to the scores above, MHL requires a working version of `perl` installed on your machine. For Linux systems, this should in general be `/usr/bin/perl`, which is per default set in this package. In case you are using MacOS (we do not support Windows, see subsection 4.2), you first need to specify the option *perl.path*. Furthermore, a working version of `samtools` is required by the programs that compute MHL.

```
> set.option(perl.path = "/usr/bin/perl")
> set.option(samtools.path = "/usr/bin/")
> mhl <- compute.score.rnb(bam.file = example.bam,
+                          rnb.set = example.rnb.set, score="mhl")
```

## 3.3 Epipolymorphism and Entropy

Epipolymorphism and Entropy calculations depend on the methclone software (`https://code.google.com/archive/p/methclone/`) to compute epiallele counts and then uses R functions to compute the final scores. This package comes with an executable version of methclone and has been tested for several Debian versions. If you have trouble with the methclone version, please contact the author. In contrast to the scores discussed above, Epipolymorphism and Entropy do not require an annotation object (either `GRanges` or `RnBSet`), since methclone operates as a black box and produces scores at positions directly inferred from the *bam* file.

```
> epipoly <- compute.score(example.bam,score="epipolymorphism")
> entropy <- compute.score(example.bam,score="entropy")
> to.plot <- data.frame(Epipolymorphism=epipoly$Epipolymorphism,
+                       Entropy=entropy$Entropy)
> to.plot <- melt(to.plot)
> plot <- ggplot(to.plot,aes(x=value,y=..density..,color=variable))+
+   geom_density()+theme_bw()
> plot
```

# 4 Advanced Configuration

## 4.1 Option settings

The WSH package provides several options, which influence how the data is handled. This includes setting coverage thresholds on the annotation, distances between individual CpGs, or quality thresholds on reads to be considered in the calculation. For a detailed description of each of the options, see the R documentation or the reference manual.

```
> ?set.option
```

## 4.2 Windows troubleshooting

qFDRP, FDRP and PDR are the only scores supported on Windows machines, since they do not rely on external tools. In contrast, MHL depends on both `perl` and `samtools`, and since `samtools` is not easily installable on a Windows machine, we MHL cannot be computed on a Windows machine. Epipolymorphism and Entropy depend on the methclone software, which is not supported for Windows and we thus also not support it here.

## 4.3 Exporting results to Genome Browser tracks

We provide a function to export the results as a Genome Browser track. WSH scores can either be aggregated over genomic bins (parameters `bin.width`, default = 5kb) or exported in single CpG format. The package will output a BED file with the information on the output generated with `compute.score`.

```
> create.genomebrowser.track(score.output=qfdrp,
+                            bin.width=7500,sample.name="mySample")
> bed.file <- readLines("mySample_qFDRP.bed")
> head(bed.file)

[1] "browser position chr2:2298361-2305861"
[2] "track type=bed name=\"mySample\" description=\"qFDRP scores in 7500 tiles\" useScore=1"
[3] "chr2 2298361 2305861 '9%'  9.28"
[4] "chr2 2305861 2313361 '11%' 11.14"
[5] "chr2 2313361 2320861 '11%' 10.56"
[6] "chr2 2320861 2328361 '10%' 10.45"
```