

test__fisher

John Stansfield

December 14, 2017

```
library(readr)
library(data.table)
library(HiCcompare)
library(dplyr)
library(pROC)

make_changes <- function(hic.table, N = 300, FC = 2, quant = 0.25) {
  # introduce changes
  changes <- sample(1:nrow(hic.table), size = N)
  whichIF = ifelse(hic.table[changes, ]$M < 0, -1, 1)
  newIF1 = FC^whichIF * hic.table[changes,]$IF2
  newIF1 = (round(newIF1))
  newIF1[newIF1 < 0.01] <- 1
  hic.table[changes,]$IF1 = newIF1
  hic.table = hic.table[, M := log2(IF2/IF1)]

  # make truth vector
  truth <- rep(0, nrow(hic.table))
  truth[changes] <- 1
  hic.table[, truth := truth]

  # normalize
  hic.table <- hic_loess(hic.table, Plot = FALSE)
  hic.table <- hic_diff(hic.table, Plot = TRUE)

  # roc_result <- roc(response = result$truth, predictor = result$p.val)
  # return(roc_result)
  return(hic.table)
}
```

Test fisher test

N = 300 FC = 1.5

```
hic.table <- dplfc1_2[[1]]
backup.table <- hic.table

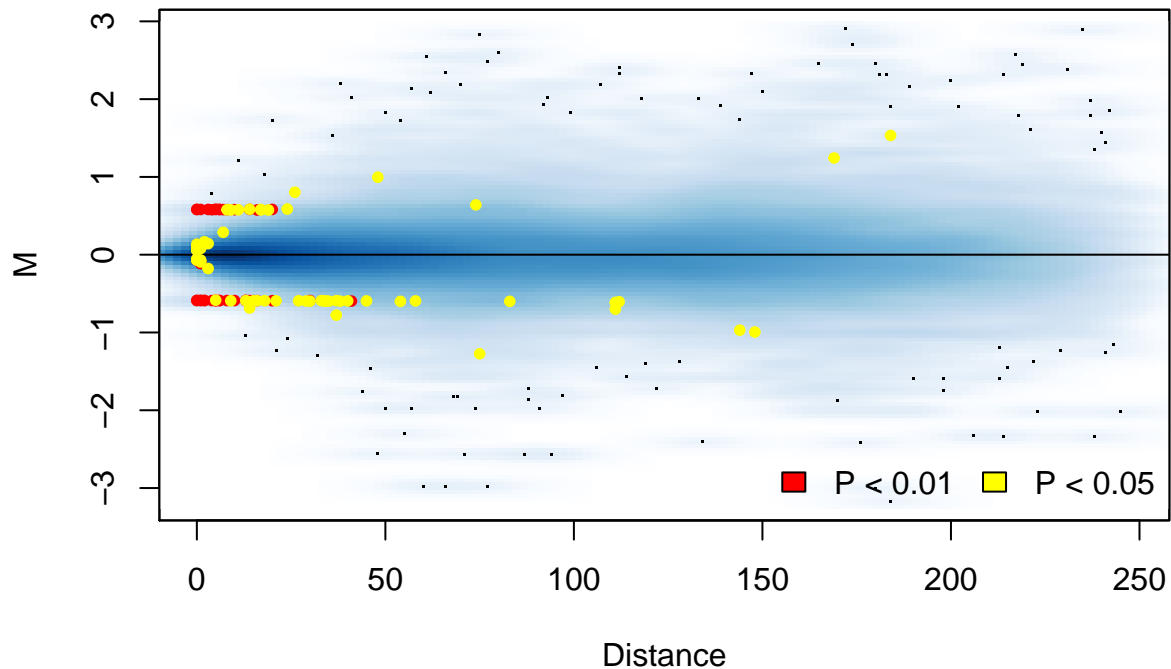
hic.table <- make_changes(hic.table, N = 300, FC = 1.5, quant = 0.25)

## Warning in `[<-.data.table`(`*tmp*`, changes, , value = structure(list(chr1
## = c("chr1", : Coerced 'double' RHS to 'integer' to match the column's
## type; may have truncated precision. Either change the target column to
## 'double' first (by creating a new 'double' vector length 25836 (nrows of
```

```
## entire table) and assign that; i.e. 'replace' column), or coerce RHS to
## 'integer' (e.g. 1L, NA_[real|integer]_, as.*, etc) to make your intent
## clear and for speed. Or, set the column type correctly up front when you
## create the table and stick to it, please.

## Span for loess: 0.899930807098073
## GCV for loess: 6.05376944280131e-06
## AIC for loess: -0.854961351681373
```

MD Plot

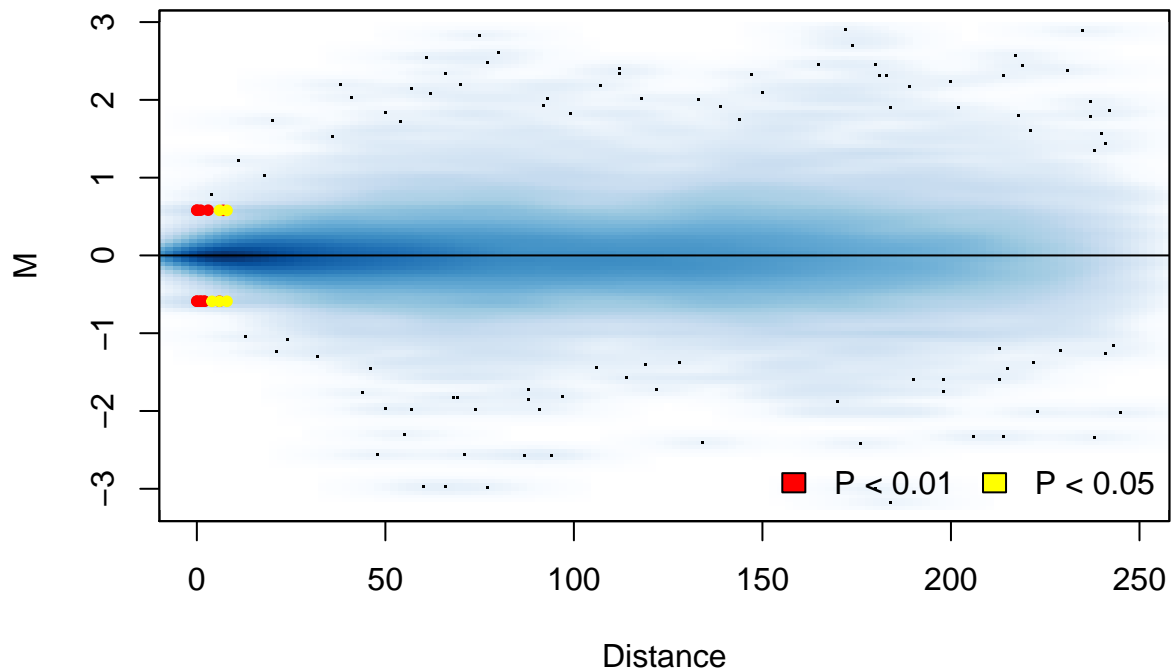


```
## True positive rate:
sum(hic.table$truth == 1 & hic.table$p.val < 0.05) / 300
```

```
## [1] 0.2266667
```

```
# test p-value adjustments
hic.table[, p.adj := p.adjust(p.val, method = 'fdr')]
MD.plot2(hic.table$adj.M, hic.table$D, hic.table$p.adj)
```

MD Plot



```
## True positive rate:
sum(hic.table$truth == 1 & hic.table$p.adj < 0.05) / 300

## [1] 0.05666667
```

$N = 300$ $FC = 2$

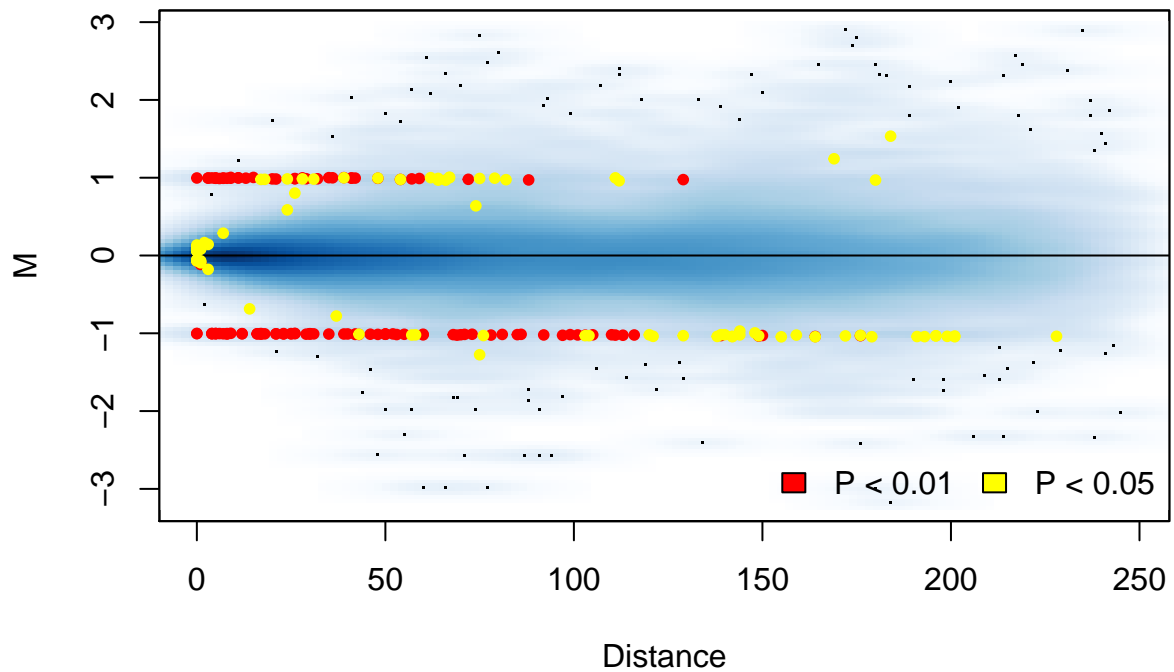
```
hic.table <- backup.table

hic.table <- make_changes(hic.table, N = 300, FC = 2, quant = 0.25)
```

```
## Warning in `[<-data.table`(`*tmp*`, changes, , value = structure(list(chr1
## = c("chr1", : Coerced 'double' RHS to 'integer' to match the column's
## type; may have truncated precision. Either change the target column to
## 'double' first (by creating a new 'double' vector length 25836 (nrows of
## entire table) and assign that; i.e. 'replace' column), or coerce RHS to
## 'integer' (e.g. 1L, NA_[real|integer]_, as.*, etc) to make your intent
## clear and for speed. Or, set the column type correctly up front when you
## create the table and stick to it, please.

## Span for loess: 0.899933350516034
## GCV for loess: 6.34564940422931e-06
## AIC for loess: -0.807873033740112
```

MD Plot



```
## True positive rate:
```

```
sum(hic.table$truth == 1 & hic.table$p.val < 0.05) / 300
```

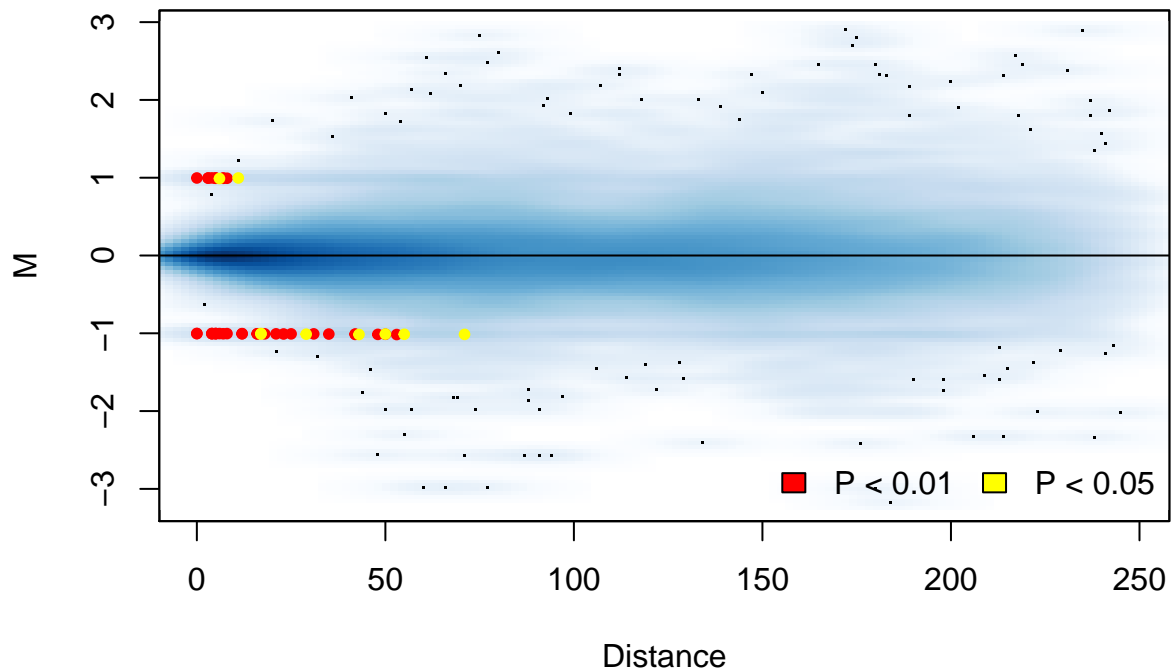
```
## [1] 0.5366667
```

```
# test p-value adjustments
```

```
hic.table[, p.adj := p.adjust(p.val, method = 'fdr')]
```

```
MD.plot2(hic.table$adj.M, hic.table$D, hic.table$p.adj)
```

MD Plot



```
## True positive rate:
sum(hic.table$truth == 1 & hic.table$p.adj < 0.05) / 300

## [1] 0.15
```

$N = 300$ $FC = 3$

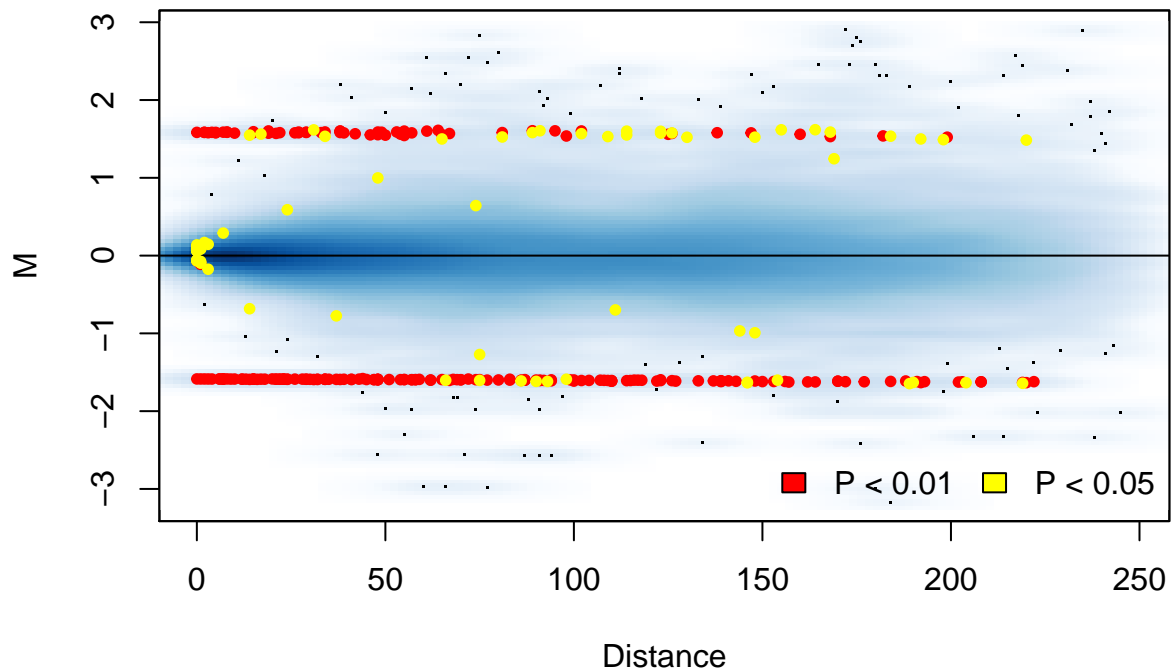
```
hic.table <- backup.table

hic.table <- make_changes(hic.table, N = 300, FC = 3, quant = 0.25)
```

```
## Warning in `[<-data.table`(`*tmp*`, changes, , value = structure(list(chr1
## = c("chr1", : Coerced 'double' RHS to 'integer' to match the column's
## type; may have truncated precision. Either change the target column to
## 'double' first (by creating a new 'double' vector length 25836 (nrows of
## entire table) and assign that; i.e. 'replace' column), or coerce RHS to
## 'integer' (e.g. 1L, NA_[real|integer]_, as.*, etc) to make your intent
## clear and for speed. Or, set the column type correctly up front when you
## create the table and stick to it, please.

## Span for loess: 0.899922082753364
## GCV for loess: 7.03438453248832e-06
## AIC for loess: -0.704832276032049
```

MD Plot



```
## True positive rate:
```

```
sum(hic.table$truth == 1 & hic.table$p.val < 0.05) / 300
```

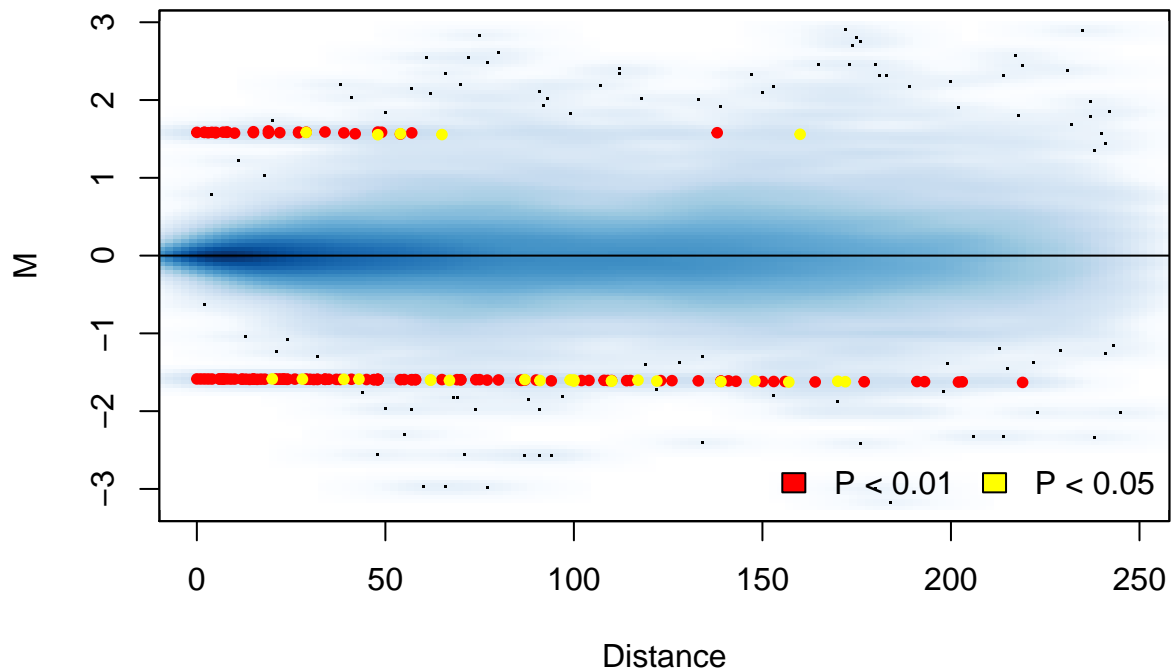
```
## [1] 0.81
```

```
# test p-value adjustments
```

```
hic.table[, p.adj := p.adjust(p.val, method = 'fdr')]
```

```
MD.plot2(hic.table$adj.M, hic.table$D, hic.table$p.adj)
```

MD Plot



```
## True positive rate:
sum(hic.table$truth == 1 & hic.table$p.adj < 0.05) / 300

## [1] 0.5033333
```

$N = 300$ $FC = 4$

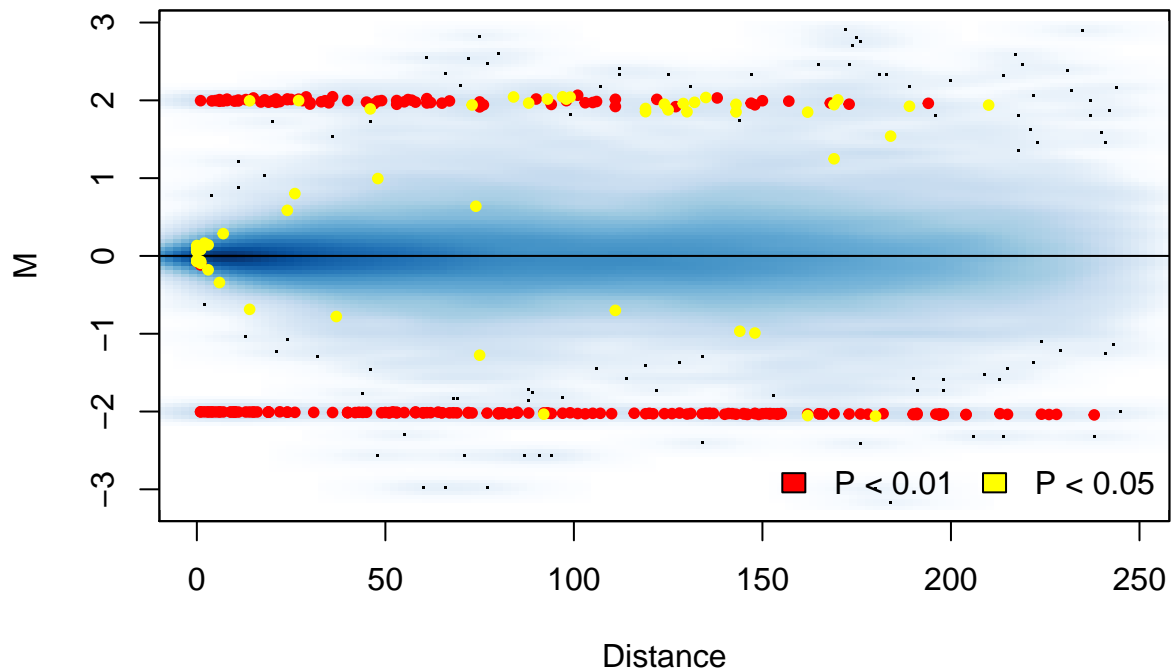
```
hic.table <- backup.table

hic.table <- make_changes(hic.table, N = 300, FC = 4, quant = 0.25)
```

```
## Warning in `[<-data.table`(`*tmp*`, changes, , value = structure(list(chr1
## = c("chr1", : Coerced 'double' RHS to 'integer' to match the column's
## type; may have truncated precision. Either change the target column to
## 'double' first (by creating a new 'double' vector length 25836 (nrows of
## entire table) and assign that; i.e. 'replace' column), or coerce RHS to
## 'integer' (e.g. 1L, NA_[real|integer]_, as.*, etc) to make your intent
## clear and for speed. Or, set the column type correctly up front when you
## create the table and stick to it, please.

## Span for loess: 0.899678144589143
## GCV for loess: 7.66856809163022e-06
## AIC for loess: -0.618512516530877
```

MD Plot



```
## True positive rate:
sum(hic.table$truth == 1 & hic.table$p.val < 0.05) / 300
```

```
## [1] 0.83
```

$N = 50$ $FC = 3$

```
hic.table <- backup.table
```

```
hic.table <- make_changes(hic.table, N = 50, FC = 4, quant = 0.25)
```

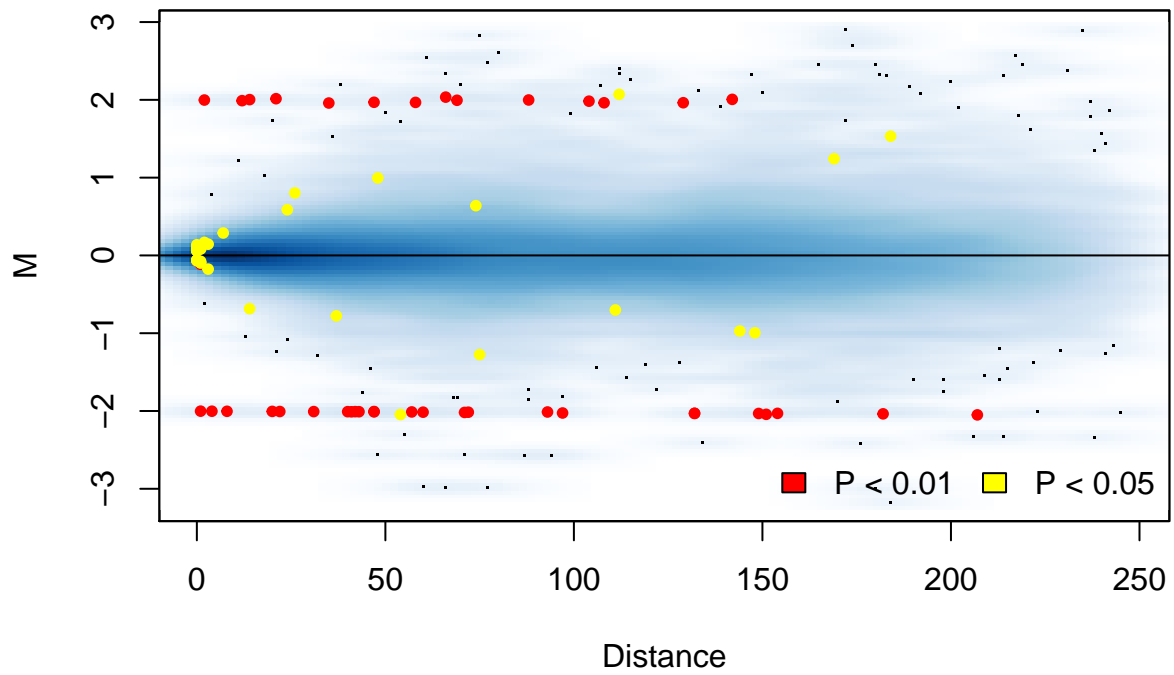
```
## Warning in `[<-data.table`(`*tmp*`, changes, , value = structure(list(chr1
## = c("chr1", : Coerced 'double' RHS to 'integer' to match the column's
## type; may have truncated precision. Either change the target column to
## 'double' first (by creating a new 'double' vector length 25836 (nrows of
## entire table) and assign that; i.e. 'replace' column), or coerce RHS to
## 'integer' (e.g. 1L, NA_[real|integer]_, as.*, etc) to make your intent
## clear and for speed. Or, set the column type correctly up front when you
## create the table and stick to it, please.
```

```
## Span for loess: 0.899945506167573
```

```
## GCV for loess: 6.25019095077518e-06
```

```
## AIC for loess: -0.823030465790362
```


MD Plot



```
sum(hic.table$truth == 1 & hic.table$p.val < 0.05) / 50
```

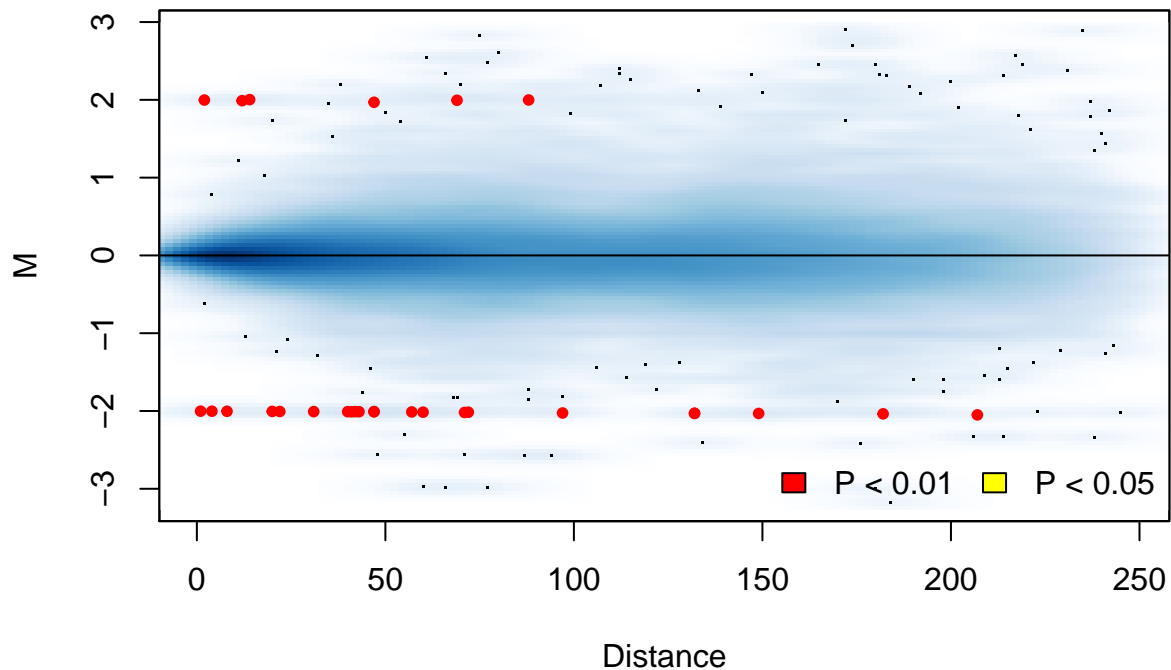
```
## [1] 0.82
```

```
# test p-value adjustments
```

```
hic.table[, p.adj := p.adjust(p.val, method = 'fdr')]
```

```
MD.plot2(hic.table$adj.M, hic.table$D, hic.table$p.adj)
```

MD Plot



```
## True positive rate:
sum(hic.table$truth == 1 & hic.table$p.adj < 0.05) / 300

## [1] 0.09333333
```

N = 50 FC = 4

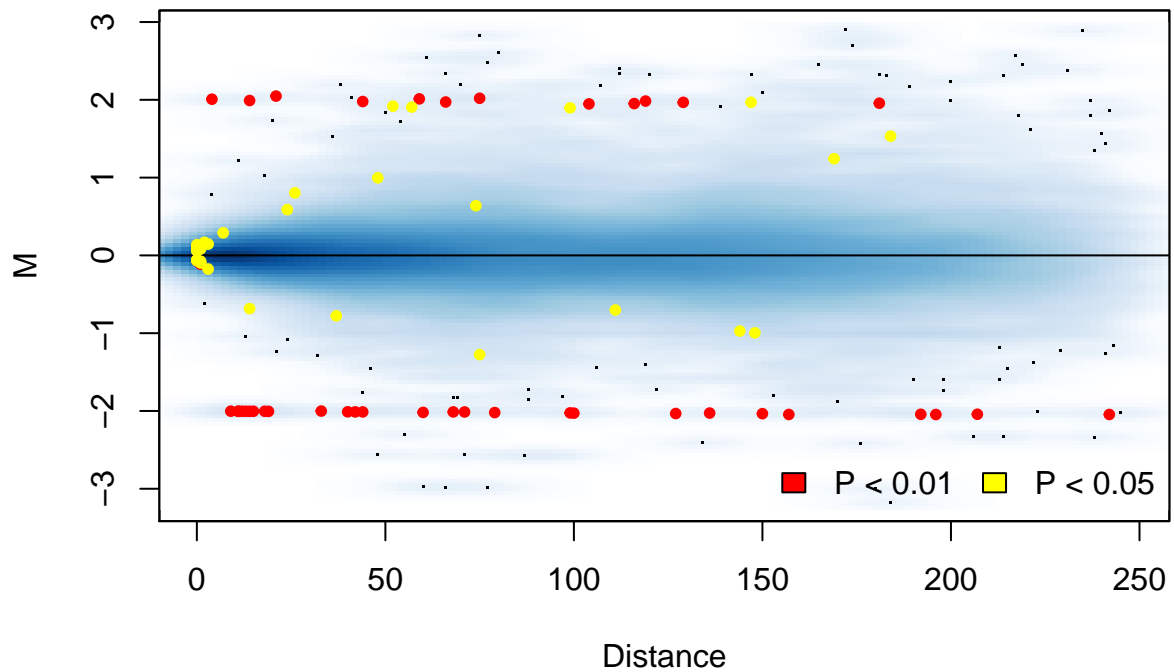
```
hic.table <- backup.table

hic.table <- make_changes(hic.table, N = 50, FC = 4, quant = 0.25)
```

```
## Warning in `[<-data.table`(`*tmp*`, changes, , value = structure(list(chr1
## = c("chr1", : Coerced 'double' RHS to 'integer' to match the column's
## type; may have truncated precision. Either change the target column to
## 'double' first (by creating a new 'double' vector length 25836 (nrows of
## entire table) and assign that; i.e. 'replace' column), or coerce RHS to
## 'integer' (e.g. 1L, NA_[real|integer]_, as.*, etc) to make your intent
## clear and for speed. Or, set the column type correctly up front when you
## create the table and stick to it, please.

## Span for loess: 0.899929958808223
## GCV for loess: 6.2426346271636e-06
## AIC for loess: -0.824240168763802
```

MD Plot

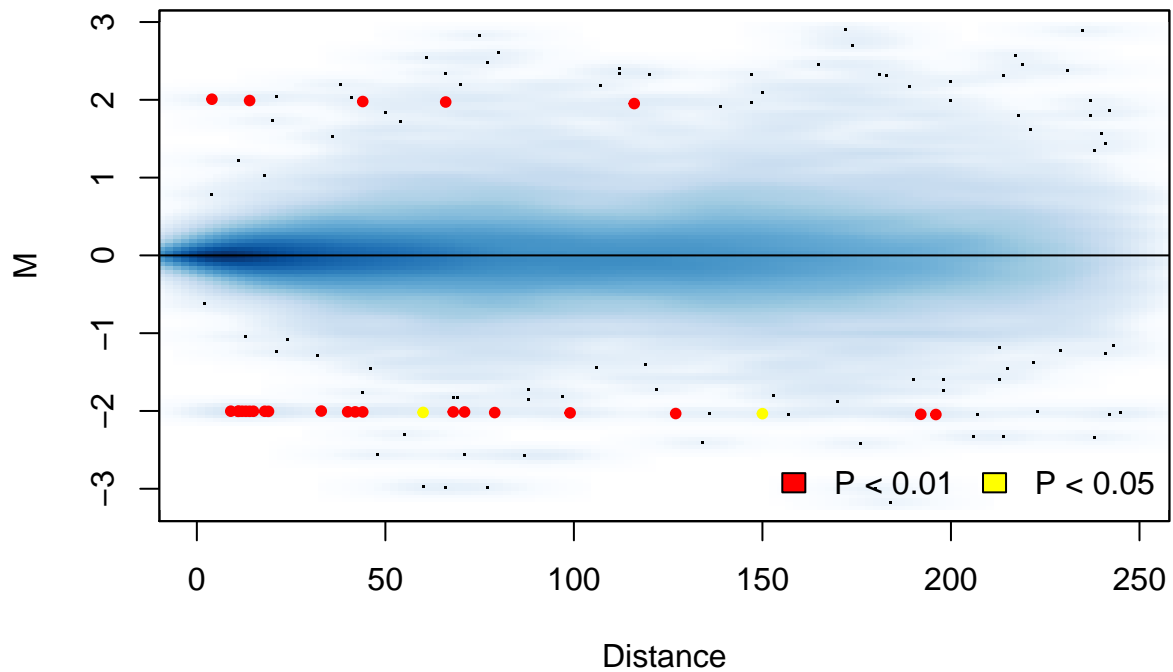


```
## True positive rate:  
sum(hic.table$truth == 1 & hic.table$p.val < 0.05) / 50
```

```
## [1] 0.86
```

```
# test p-value adjustments  
hic.table[, p.adj := p.adjust(p.val, method = 'fdr')]  
MD.plot2(hic.table$adj.M, hic.table$D, hic.table$p.adj)
```

MD Plot



```
## True positive rate:
sum(hic.table$truth == 1 & hic.table$p.adj < 0.05) / 300

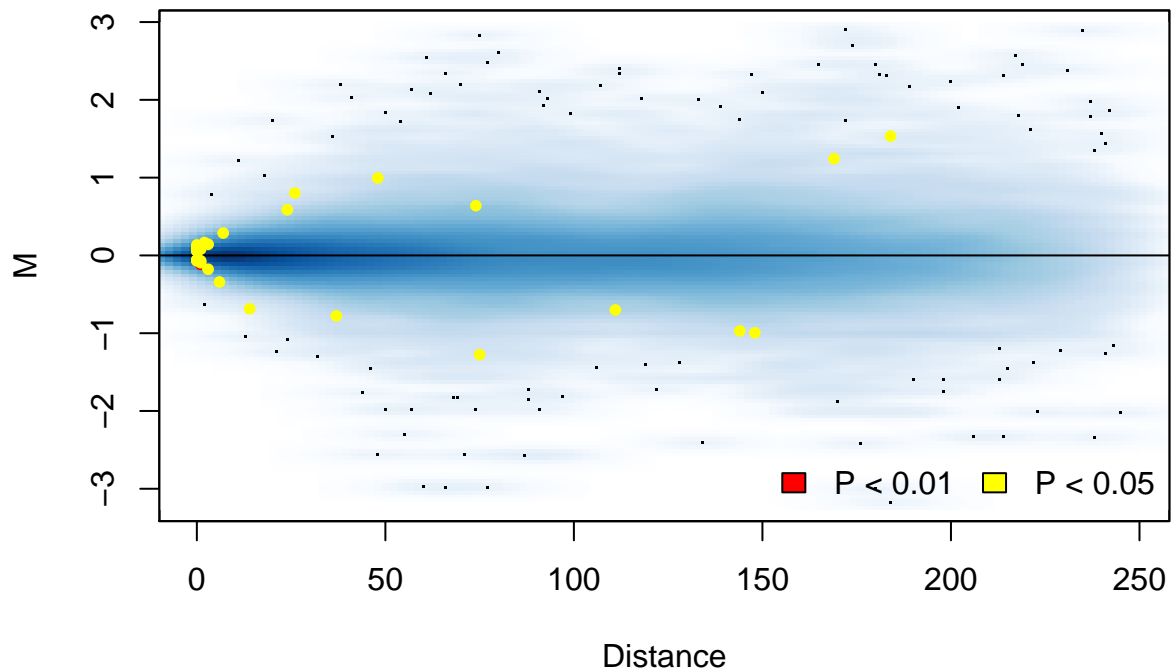
## [1] 0.09
```

No changes

```
hic.table <- backup.table
hic.table <- hic_loess(hic.table, Plot = FALSE)

## Span for loess: 0.89992480205479
## GCV for loess: 5.9608868807595e-06
## AIC for loess: -0.870423201341588
hic.table <- hic_diff(hic.table, Plot = TRUE)
```

MD Plot

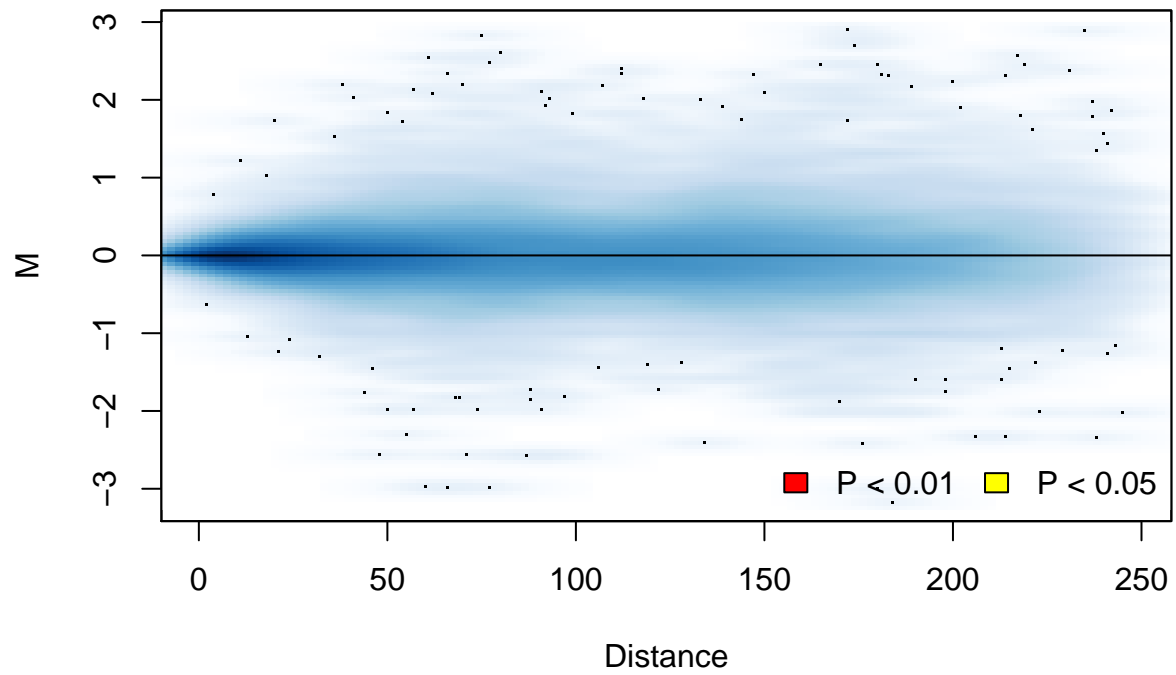


```
## False positives
sum(hic.table$p.val < 0.05)
```

```
## [1] 45
```

```
# test p-value adjustments
hic.table[, p.adj := p.adjust(p.val, method = 'fdr')]
MD.plot2(hic.table$adj.M, hic.table$D, hic.table$p.adj)
```

MD Plot



```
## False positives  
sum(hic.table$p.adj < 0.05)
```

```
## [1] 0
```