

zscore_Athreshold

John Stansfield

December 11, 2017

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

.calc_z <- function(hic.table, quant = 0.25) {
  threshold <- quantile((hic.table$A), quant, na.rm = TRUE)
  Z <- (hic.table$adj.M - mean(hic.table$adj.M)) / sd(hic.table$adj.M)
  # set z-scores where A < threshold to 0
  Z[hic.table$A < threshold] <- 0
  hic.table[, Z := Z]
  hic.table[, p.val := 2*pnorm(abs(Z), lower.tail = FALSE)]
  # hic.table[, p.adj := p.adjust(p.val, method = 'holm')]
  MD.plot2(hic.table$adj.M, hic.table$D, hic.table$p.val)
  # MD.plot2(hic.table$adj.M, hic.table$D, hic.table$p.adj)
  return(hic.table)
}

# pvalue adjustment
holm_adjust <- function(hic.table) {
  # split table up for each distance
  temp_list <- S4Vectors::split(hic.table, hic.table$D)
  # combined top 15% of distances into single data.table
  all_dist <- sort(unique(hic.table$D))
  dist_85 <- ceiling(0.85 * length(all_dist))
  temp_list2 <- temp_list[1:dist_85]
  temp_list2[[dist_85+1]] <- data.table::rbindlist(temp_list[(dist_85+1):length(temp_list)])
  temp_list <- temp_list2
  rm("temp_list2")
  # rank M by distance
  temp_list <- lapply(temp_list, function(x) {
    x[, p.adj := p.adjust(p.val, method = 'holm')]
    return(x)
  })
  # recombine into one table
  hic.table <- rbindlist(temp_list)
  MD.plot2(hic.table$adj.M, hic.table$D, hic.table$p.adj)
  return(hic.table)
}

make_roc <- 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 = FALSE)

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

```

build ROC

Varying fold change

Fold change = 1.5

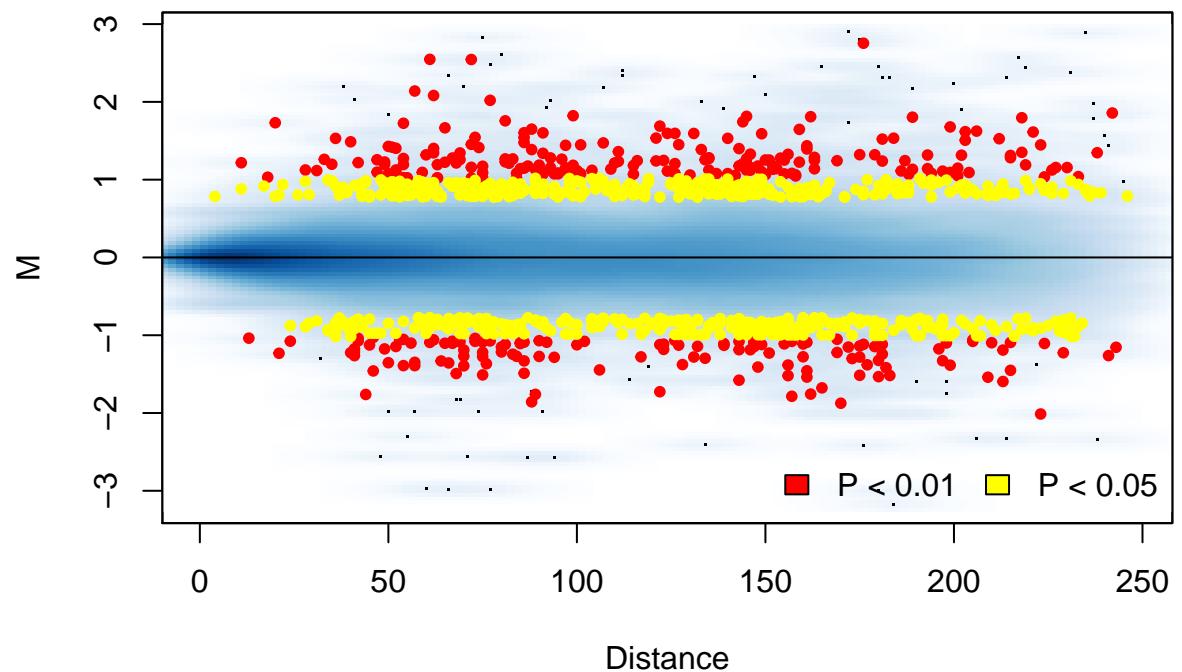
```

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

quant5 <- make_roc(hic.table, N = 300, FC = 1.5, quant = 0.05)

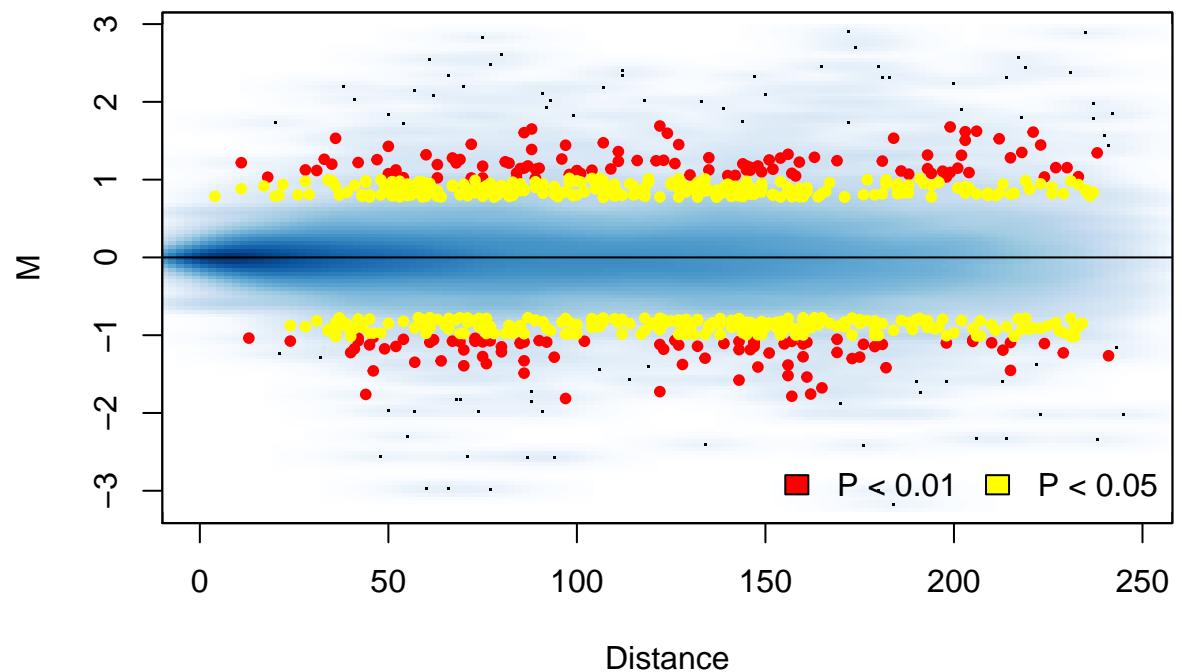
```

MD Plot



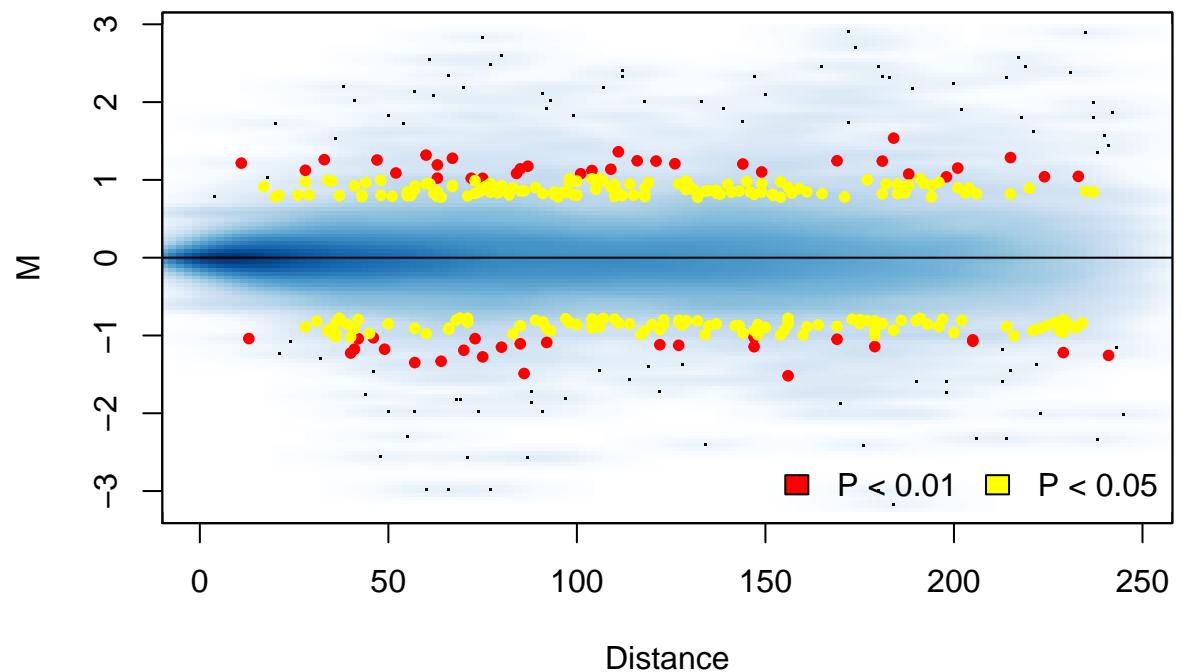
```
rm("hic.table")
hic.table <- backup.table
quant10 <- make_roc(hic.table, N = 300, FC = 1.5, quant = 0.1)
```

MD Plot



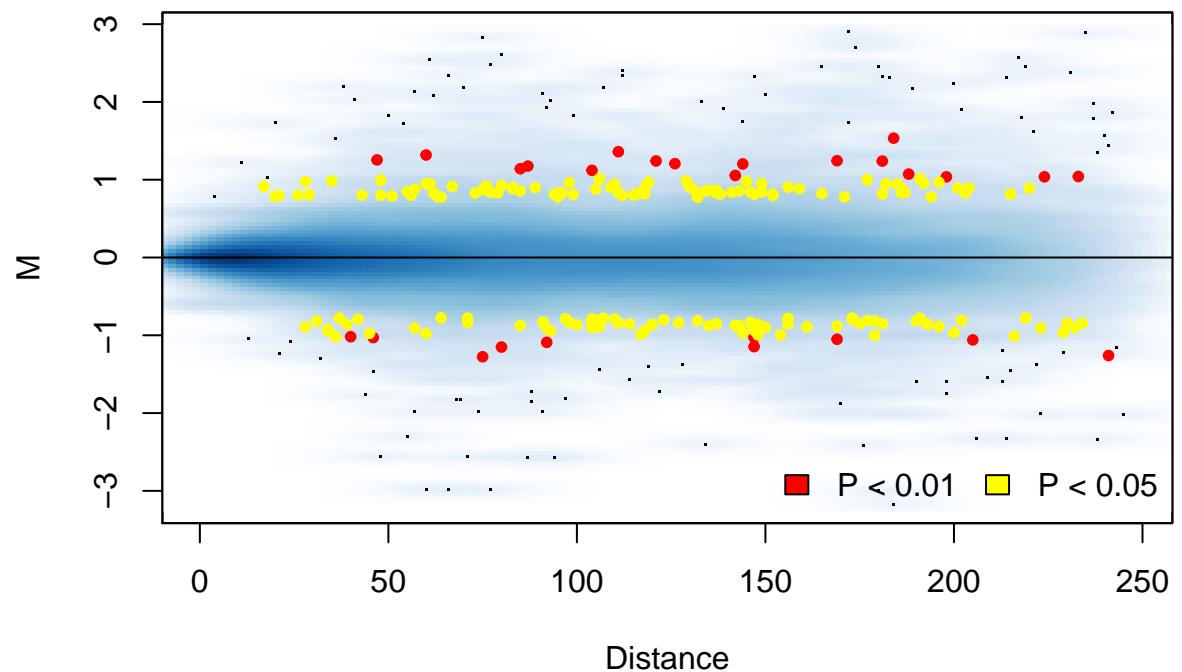
```
rm("hic.table")
hic.table <- backup.table
quant20 <- make_roc(hic.table, N = 300, FC = 1.5, quant = 0.2)
```

MD Plot



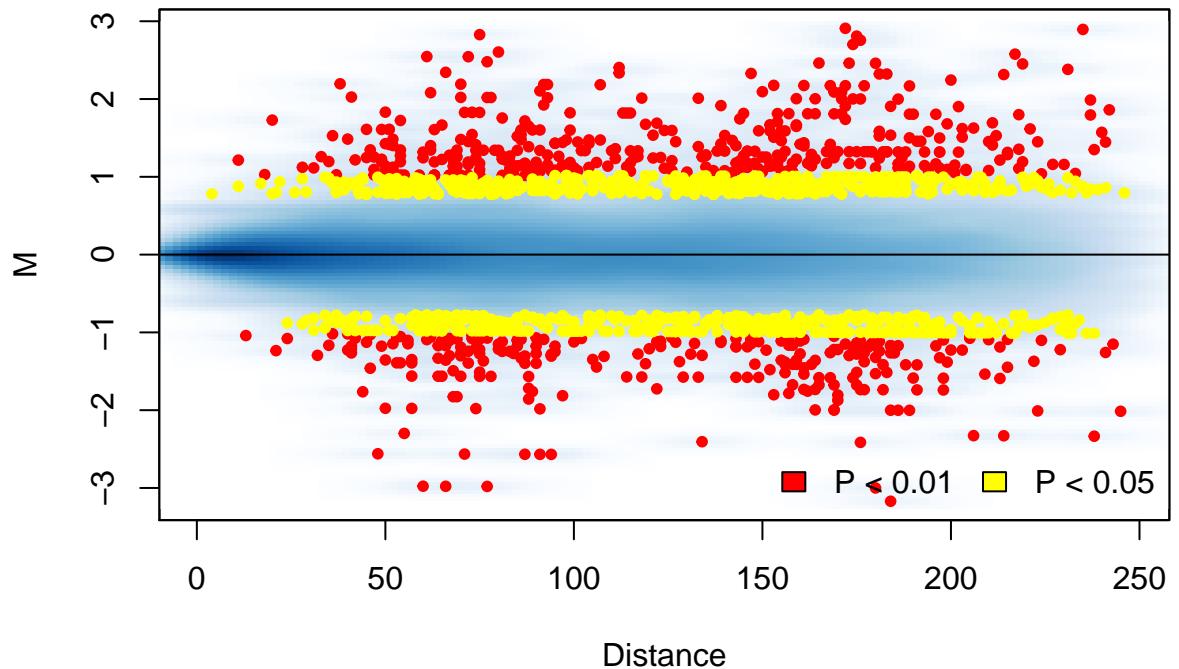
```
rm("hic.table")
hic.table <- backup.table
quant25 <- make_roc(hic.table, N = 300, FC = 1.5, quant = 0.25)
```

MD Plot

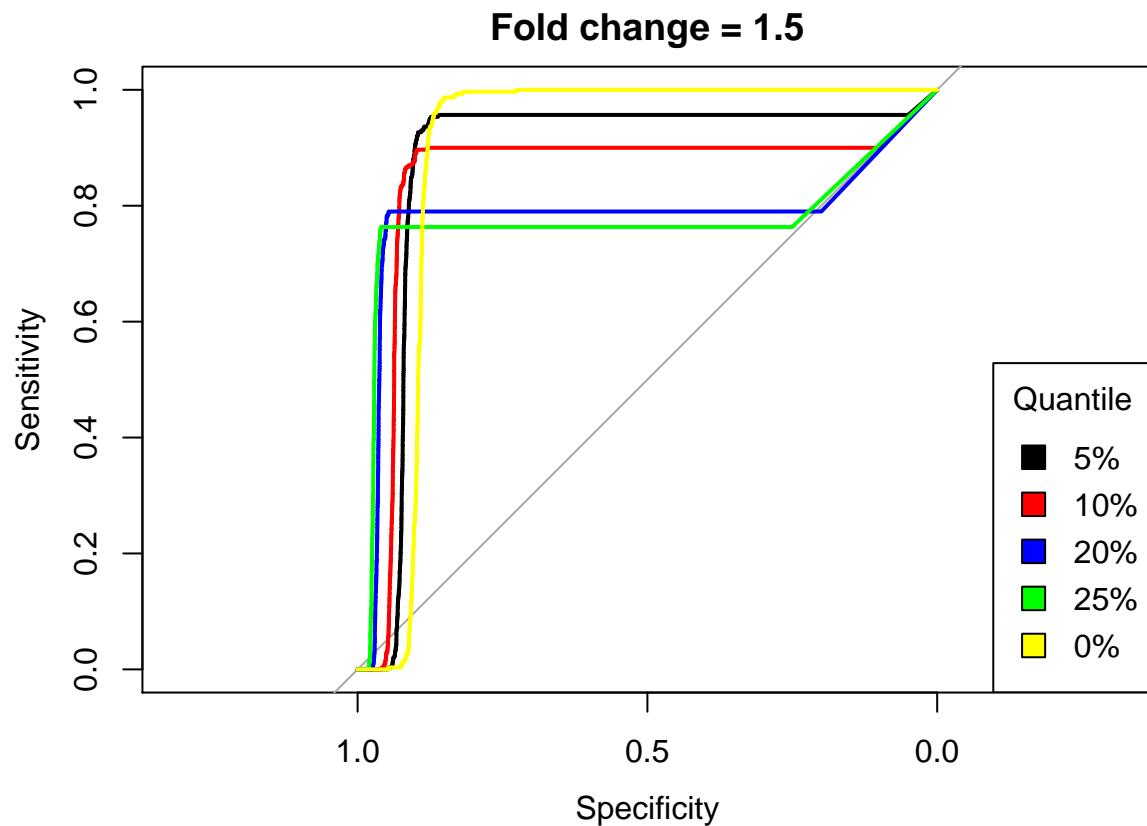


```
rm("hic.table")
hic.table <- backup.table
quant0 <- make_roc(hic.table, N = 300, FC = 1.5, quant = 0)
```

MD Plot



```
plot.colors <- c('black', 'red', 'blue', 'green', 'yellow', 'orange', 'purple', 'lightblue')
plot(quant5, main = paste0('Fold change = ', 1.5))
plot(quant10, add = TRUE, col = plot.colors[2])
plot(quant20, add = TRUE, col = plot.colors[3])
plot(quant25, add = TRUE, col = plot.colors[4])
plot(quant0, add = TRUE, col = plot.colors[5])
legend('bottomright', inset = 0, legend = c('5%', '10%', '20%', '25%', '0%'), title = 'Quantile', fill ...
```

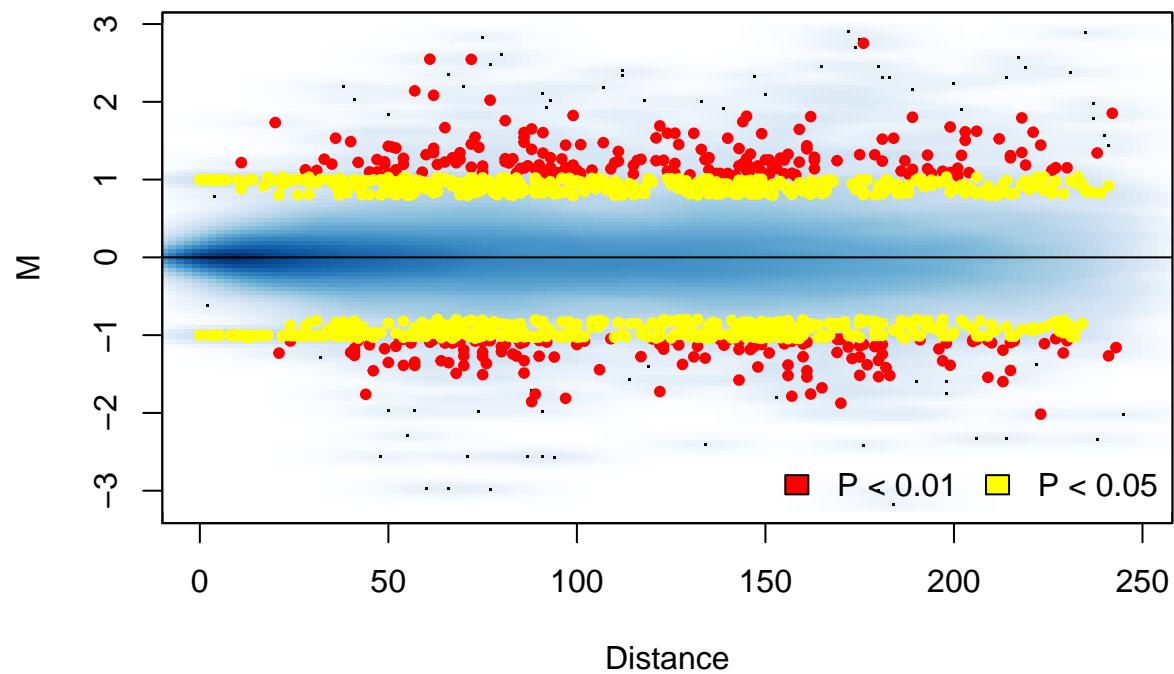


Fold change = 2

```
rm("hic.table")
hic.table <- dplfc1_2[[1]]
backup.table <- hic.table

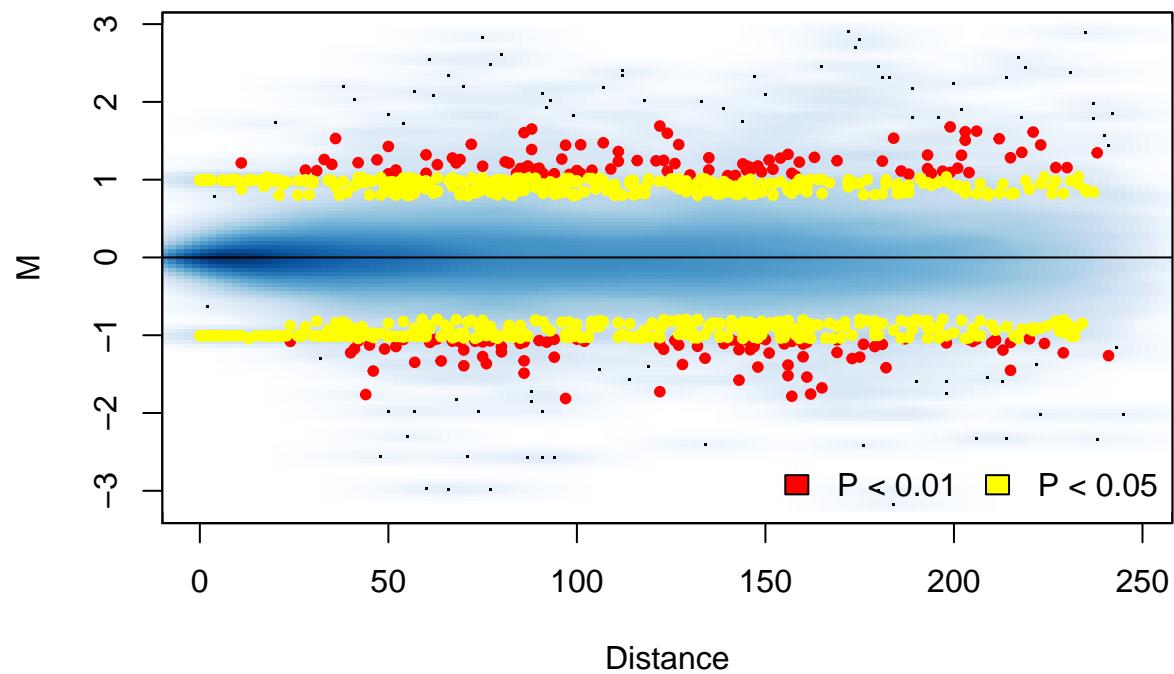
quant5 <- make_roc(hic.table, N = 300, FC = 2, quant = 0.05)
```

MD Plot



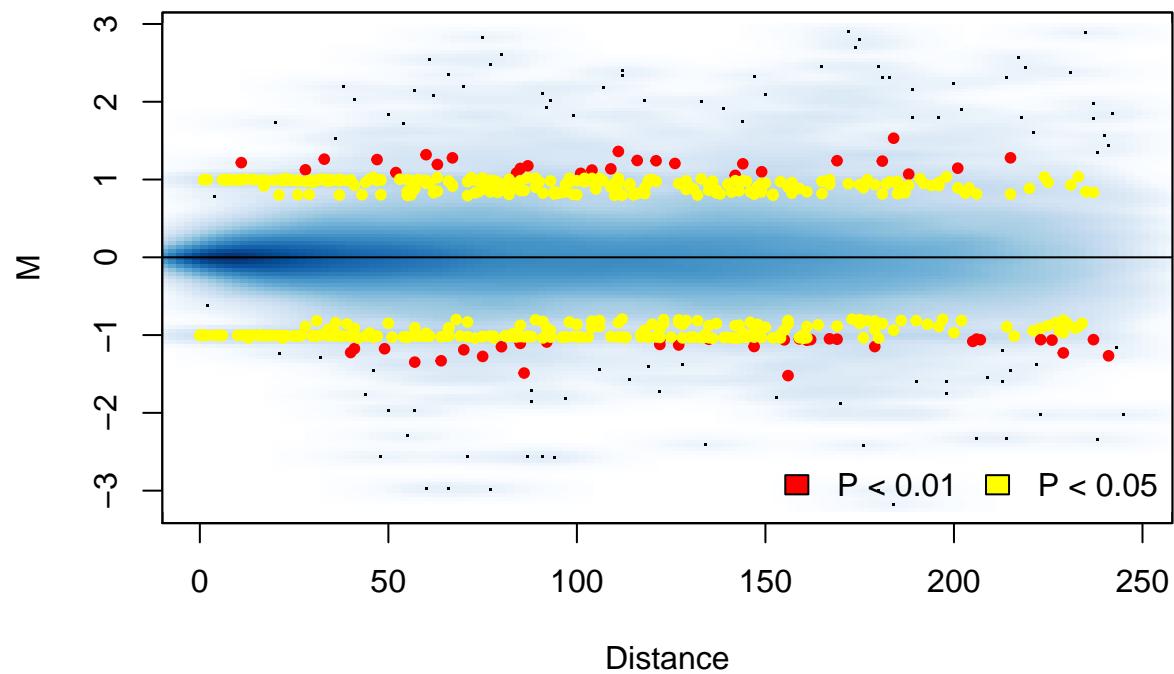
```
rm("hic.table")
hic.table <- backup.table
quant10 <- make_roc(hic.table, N = 300, FC = 2, quant = 0.1)
```

MD Plot



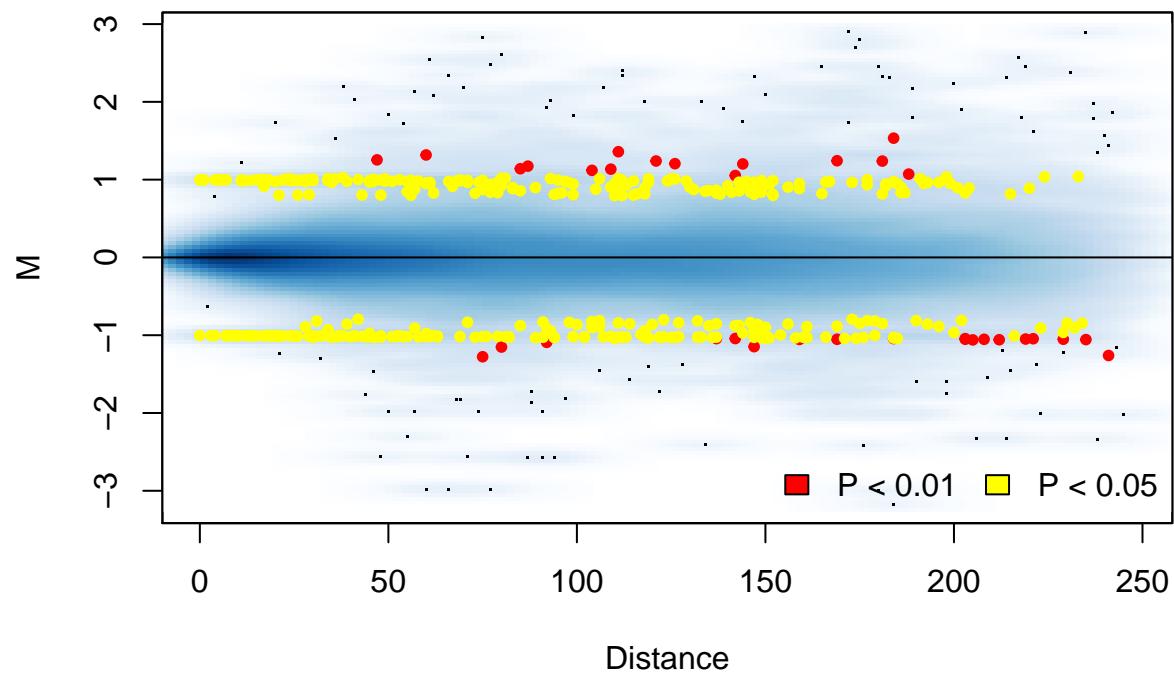
```
rm("hic.table")
hic.table <- backup.table
quant20 <- make_roc(hic.table, N = 300, FC = 2, quant = 0.2)
```

MD Plot



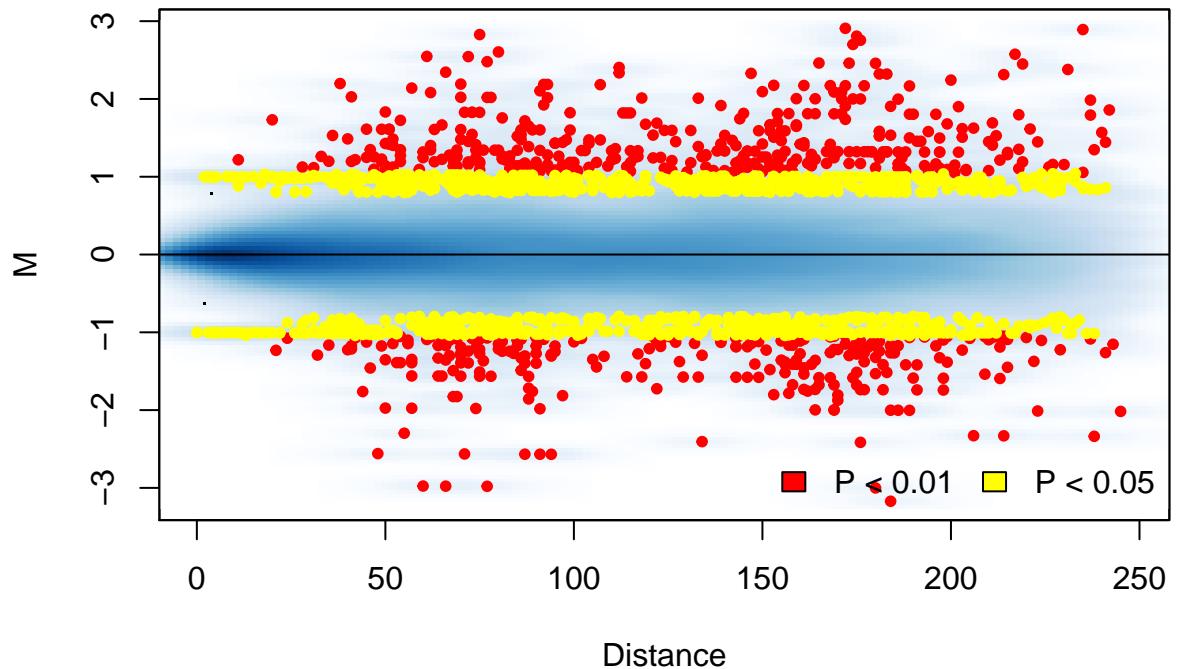
```
rm("hic.table")
hic.table <- backup.table
quant25 <- make_roc(hic.table, N = 300, FC = 2, quant = 0.25)
```

MD Plot

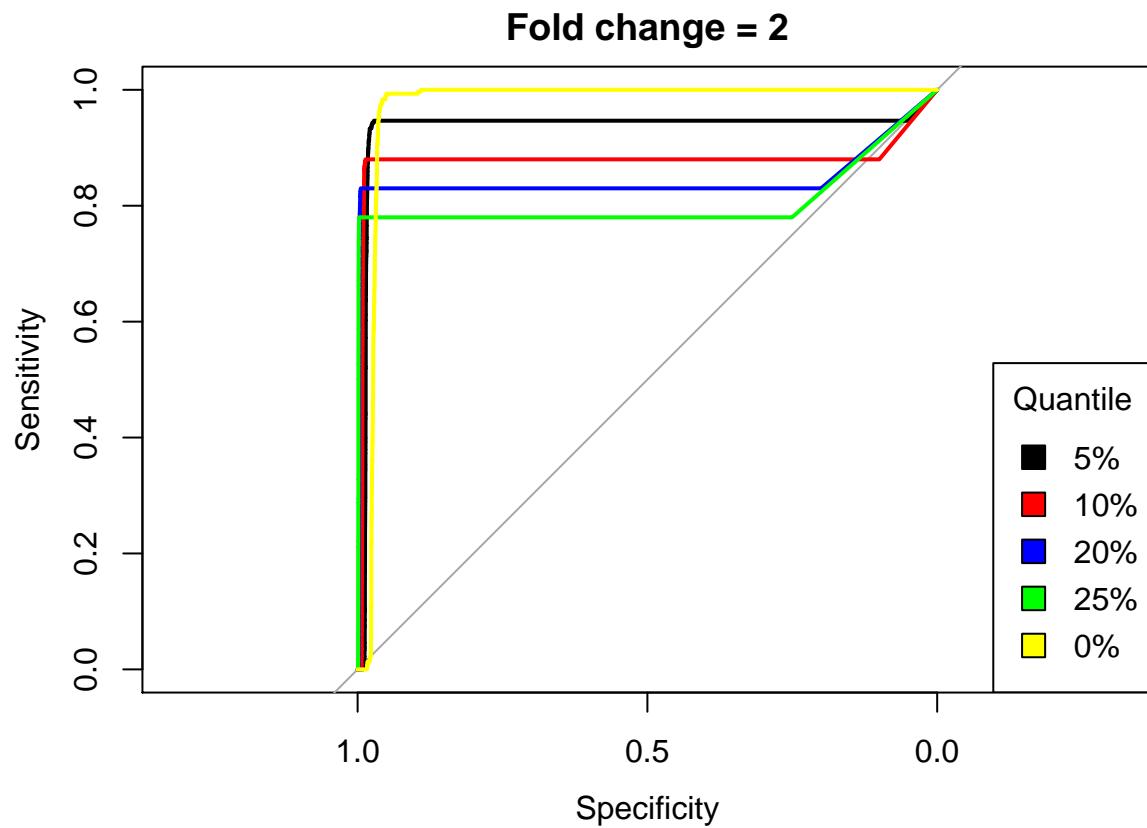


```
rm("hic.table")
hic.table <- backup.table
quant0 <- make_roc(hic.table, N = 300, FC = 2, quant = 0)
```

MD Plot



```
plot.colors <- c('black', 'red', 'blue', 'green', 'yellow', 'orange', 'purple', 'lightblue')
plot(quant5, main = paste0('Fold change = ', 2))
plot(quant10, add = TRUE, col = plot.colors[2])
plot(quant20, add = TRUE, col = plot.colors[3])
plot(quant25, add = TRUE, col = plot.colors[4])
plot(quant0, add = TRUE, col = plot.colors[5])
legend('bottomright', inset = 0, legend = c('5%', '10%', '20%', '25%', '0%'), title = 'Quantile', fill ...
```

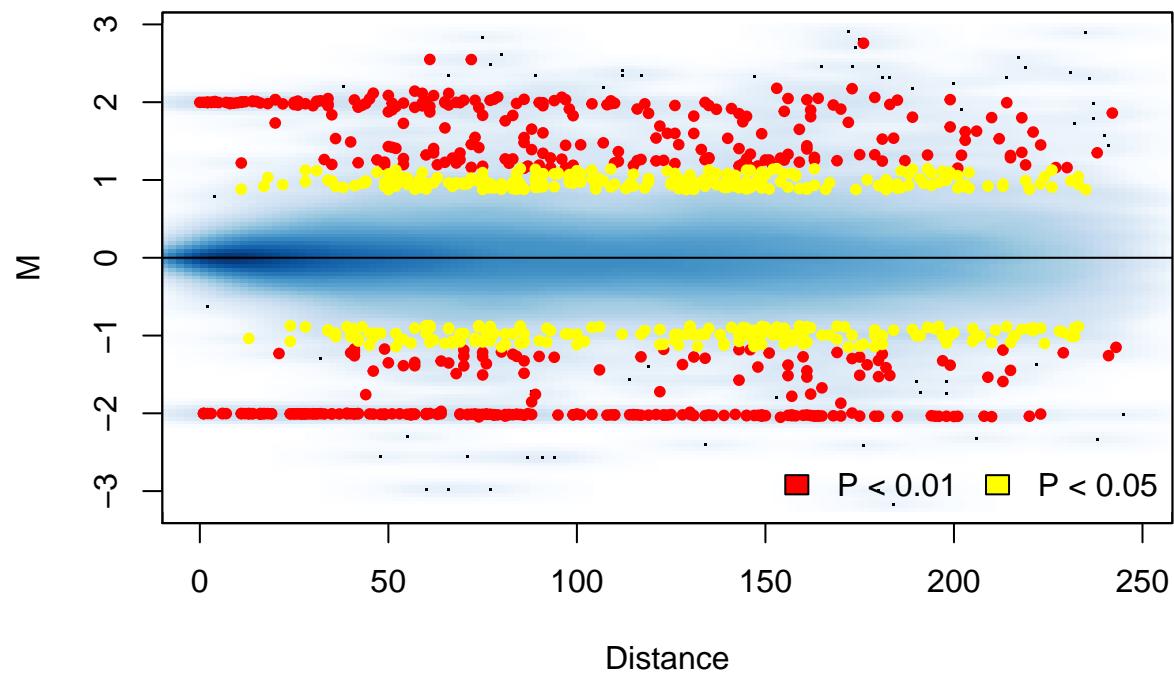


Fold change = 4

```
rm("hic.table")
hic.table <- dplfc1_2[[1]]
backup.table <- hic.table

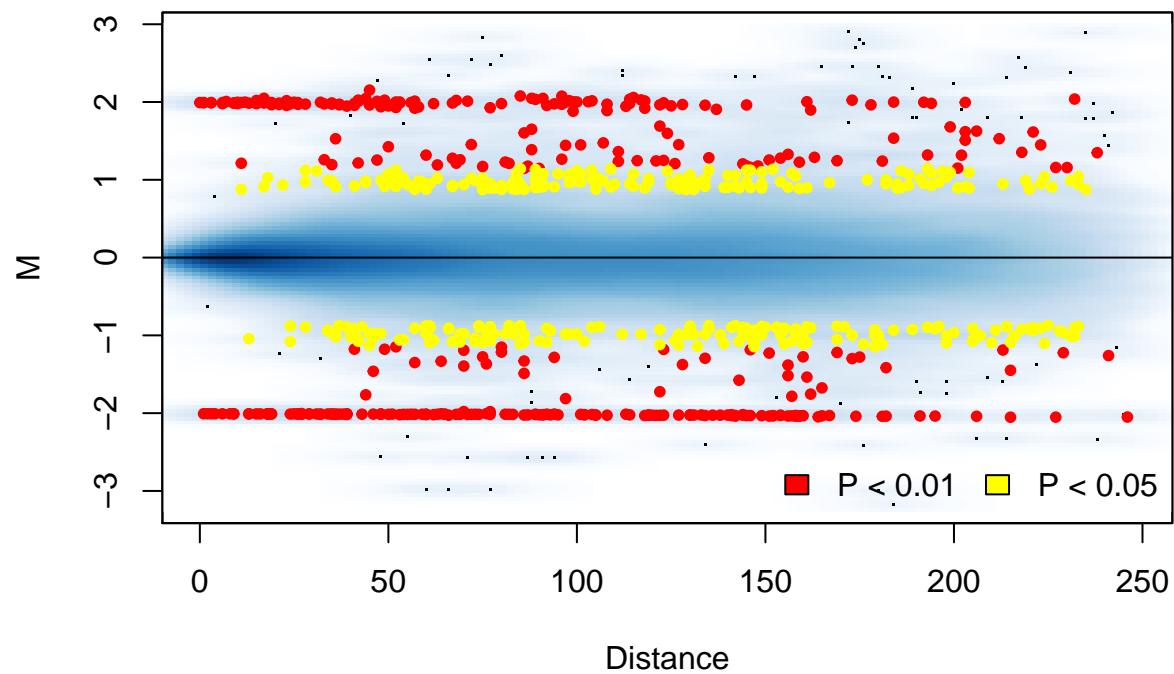
quant5 <- make_roc(hic.table, N = 300, FC = 4, quant = 0.05)
```

MD Plot



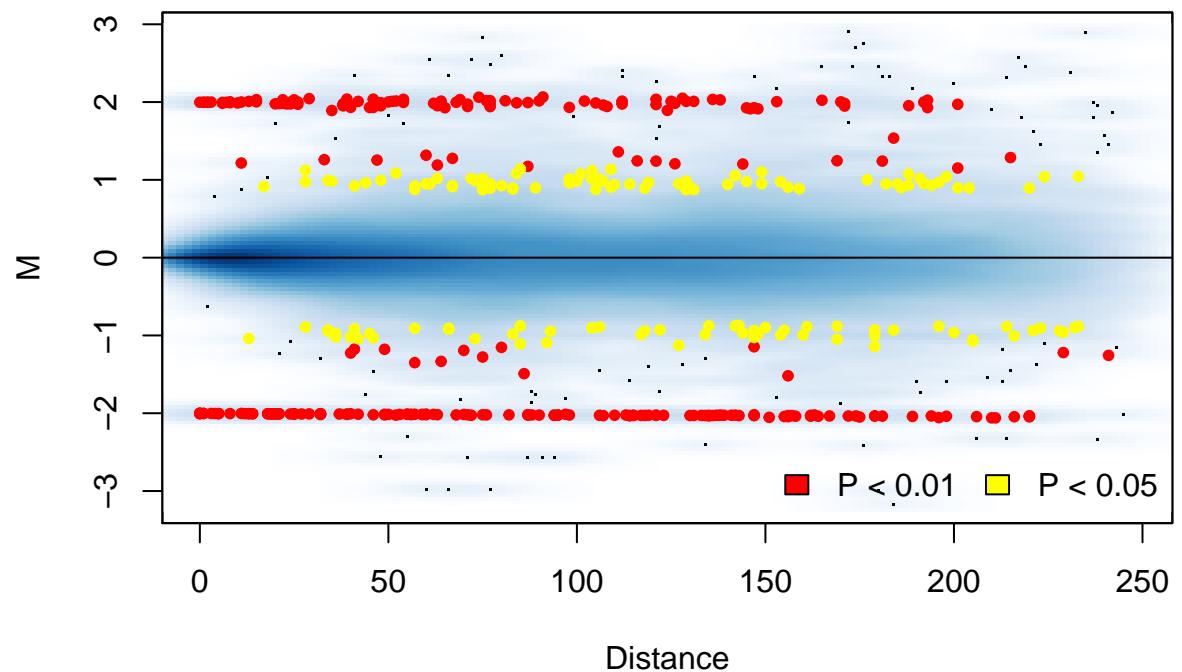
```
rm("hic.table")
hic.table <- backup.table
quant10 <- make_roc(hic.table, N = 300, FC = 4, quant = 0.1)
```

MD Plot



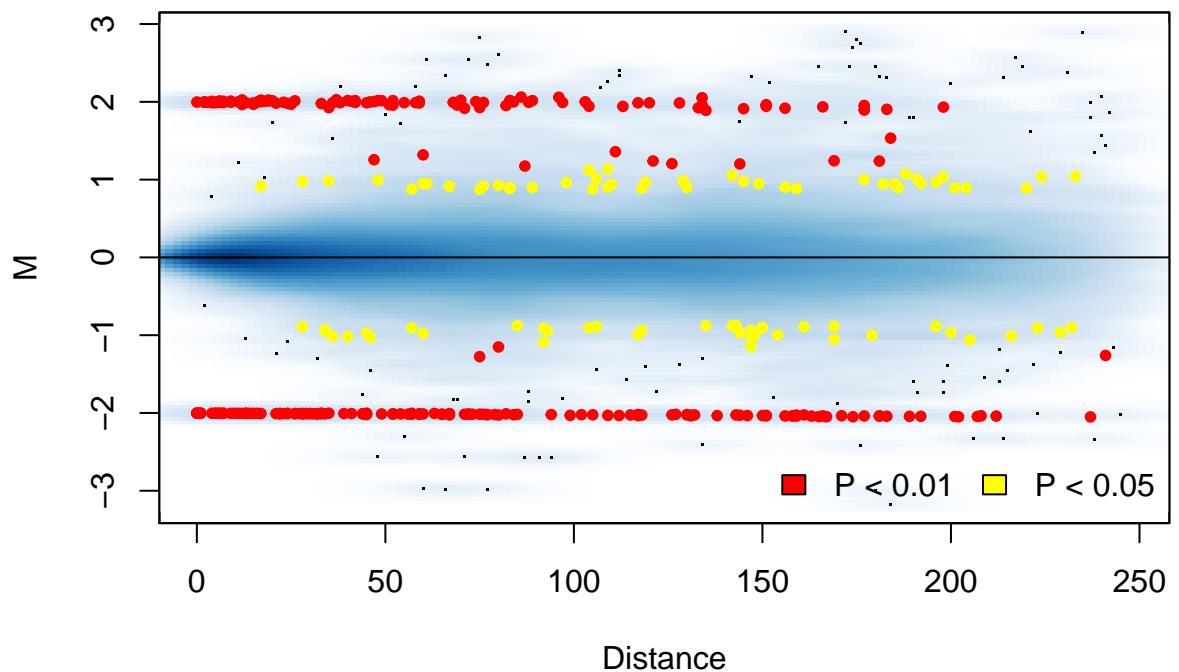
```
rm("hic.table")
hic.table <- backup.table
quant20 <- make_roc(hic.table, N = 300, FC = 4, quant = 0.2)
```

MD Plot



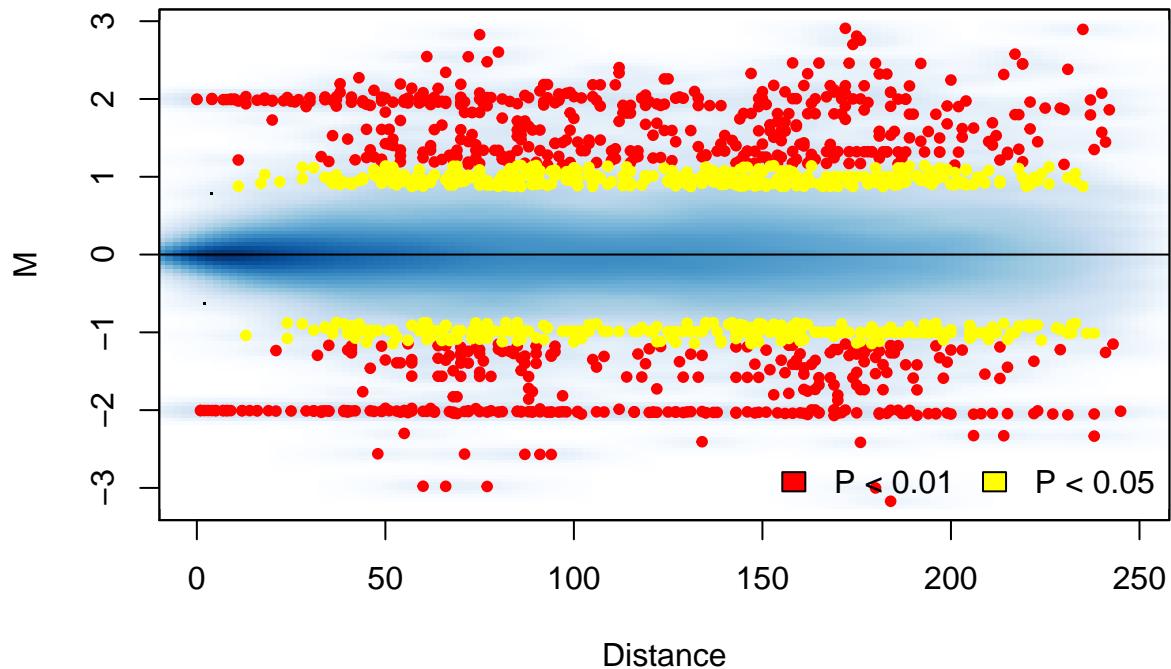
```
rm("hic.table")
hic.table <- backup.table
quant25 <- make_roc(hic.table, N = 300, FC = 4, quant = 0.25)
```

MD Plot

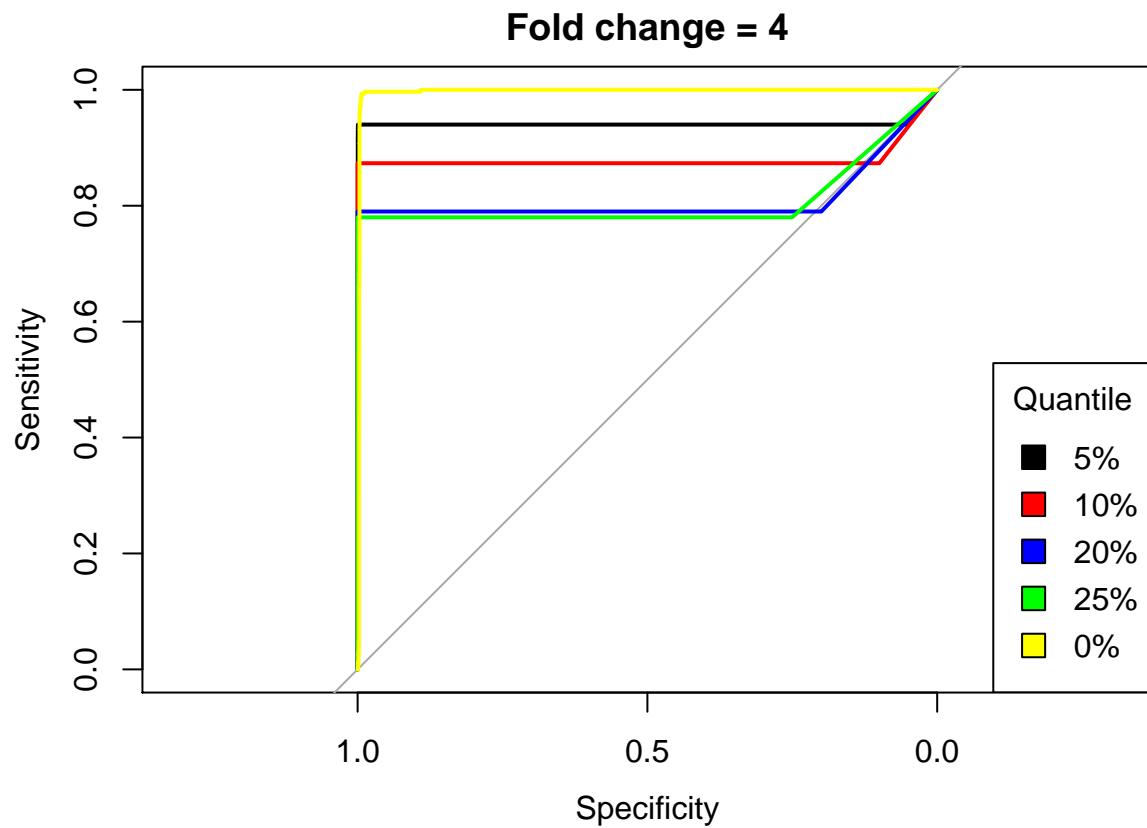


```
rm("hic.table")
hic.table <- backup.table
quant0 <- make_roc(hic.table, N = 300, FC = 4, quant = 0)
```

MD Plot



```
plot.colors <- c('black', 'red', 'blue', 'green', 'yellow', 'orange', 'purple', 'lightblue')
plot(quant5, main = paste0('Fold change = ', 4))
plot(quant10, add = TRUE, col = plot.colors[2])
plot(quant20, add = TRUE, col = plot.colors[3])
plot(quant25, add = TRUE, col = plot.colors[4])
plot(quant0, add = TRUE, col = plot.colors[5])
legend('bottomright', inset = 0, legend = c('5%', '10%', '20%', '25%', '0%'), title = 'Quantile', fill ...
```



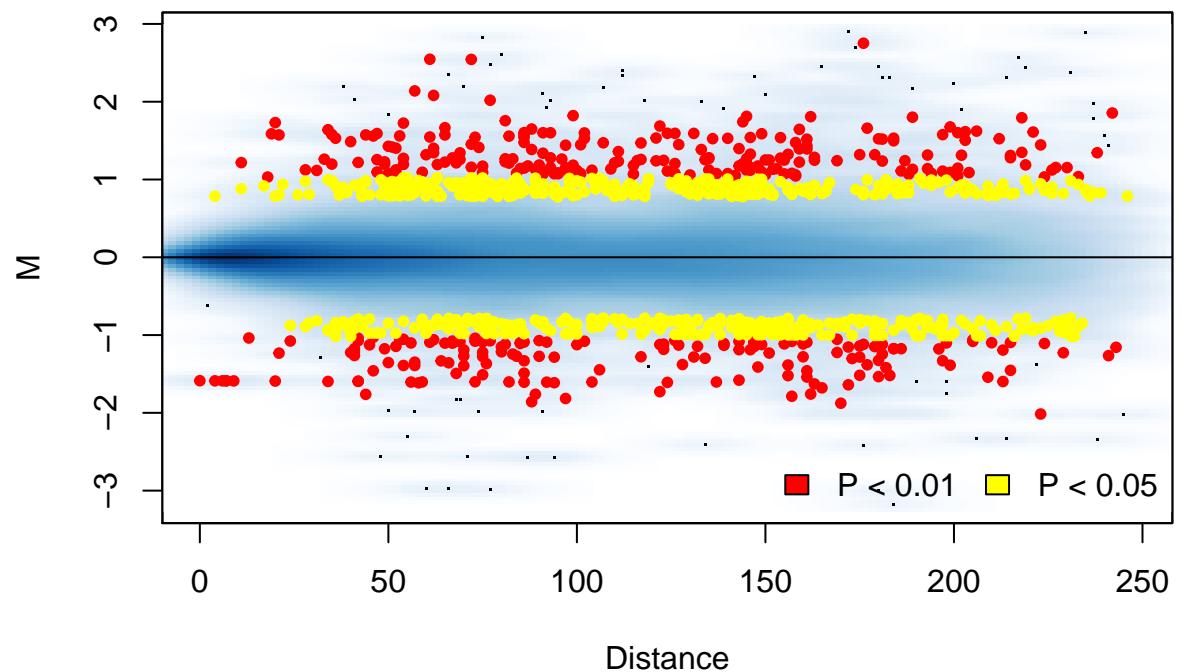
Varying number differences

50 changes 3 FC

```
rm("hic.table")
hic.table <- dplfc1_2[[1]]
backup.table <- hic.table

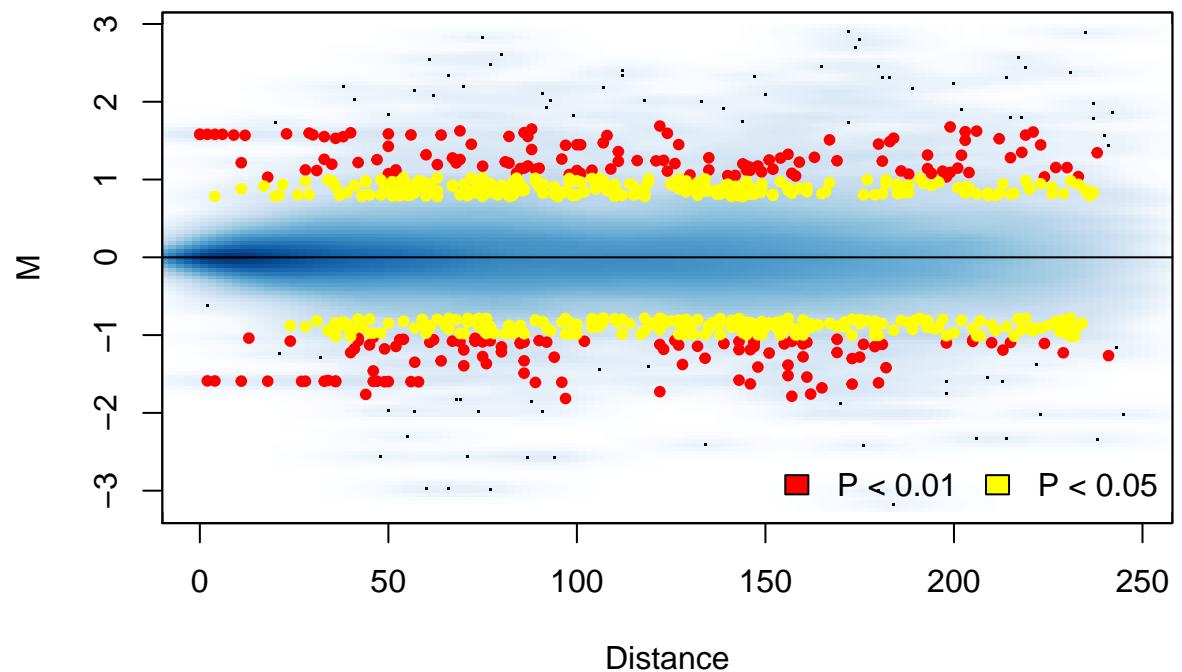
quant5 <- make_roc(hic.table, N = 50, FC = 3, quant = 0.05)
```

MD Plot



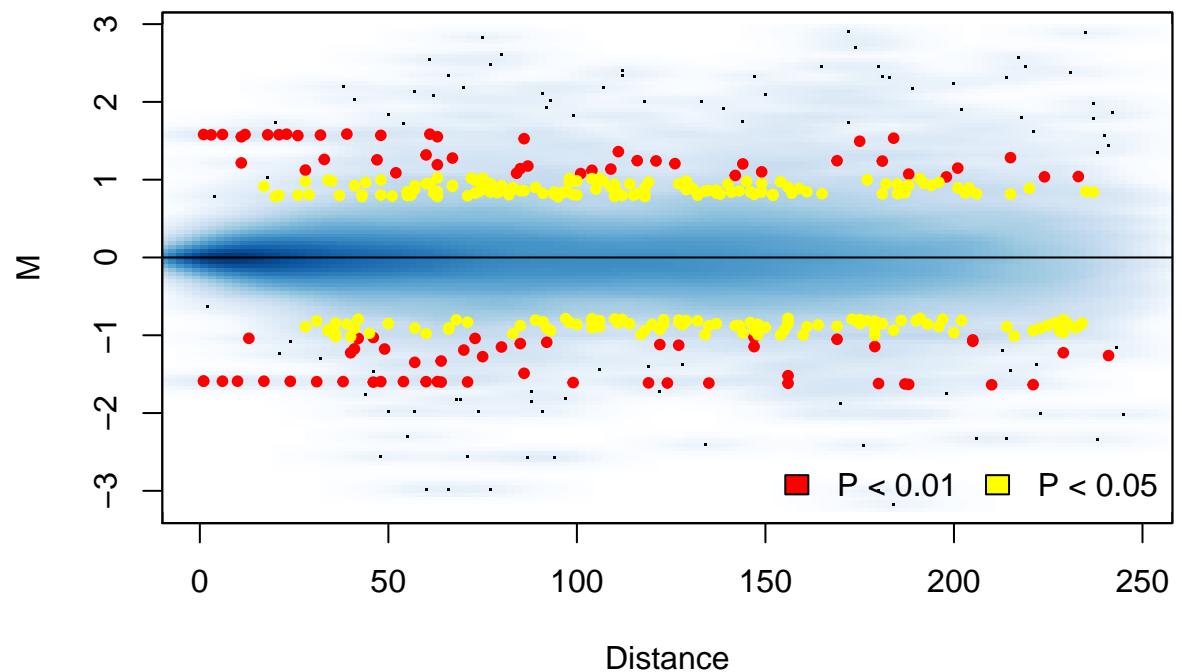
```
rm("hic.table")
hic.table <- backup.table
quant10 <- make_roc(hic.table, N = 50, FC = 3, quant = 0.1)
```

MD Plot



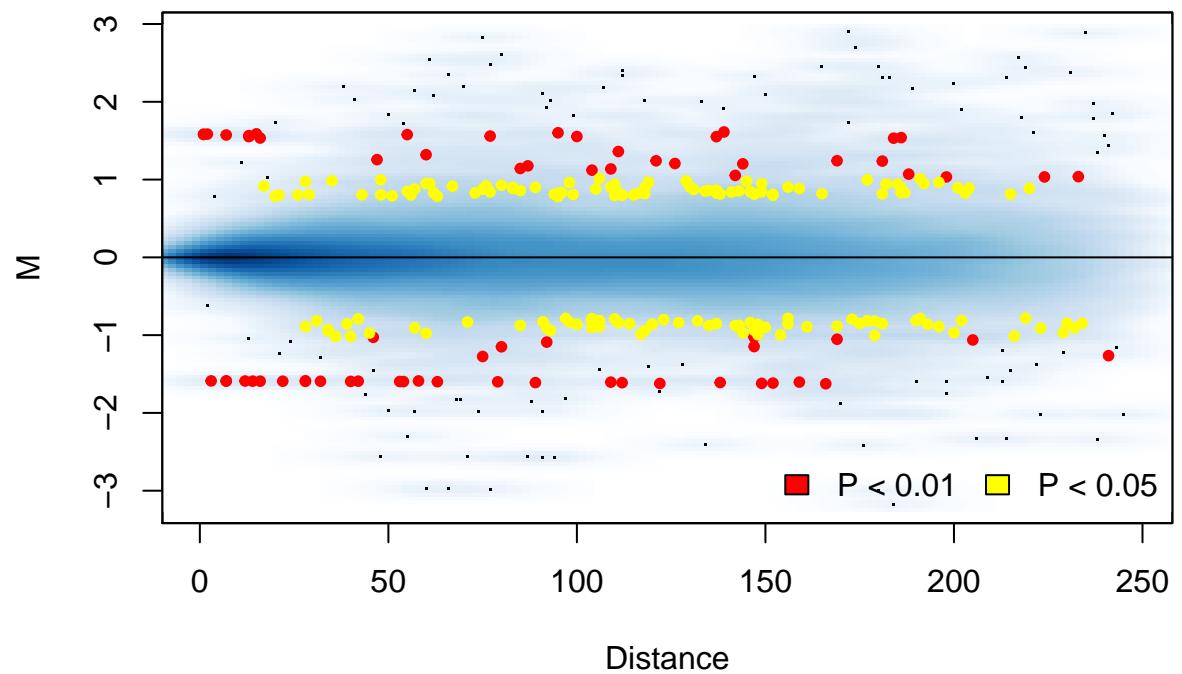
```
rm("hic.table")
hic.table <- backup.table
quant20 <- make_roc(hic.table, N = 50, FC = 3, quant = 0.2)
```

MD Plot



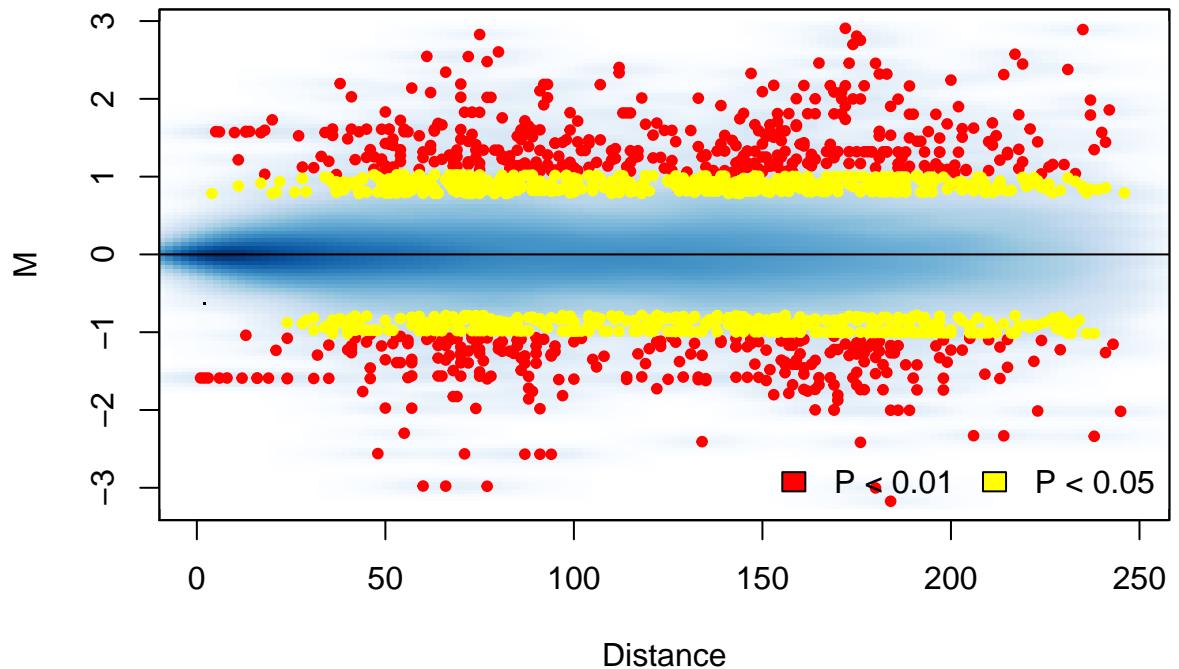
```
rm("hic.table")
hic.table <- backup.table
quant25 <- make_roc(hic.table, N = 50, FC = 3, quant = 0.25)
```

MD Plot

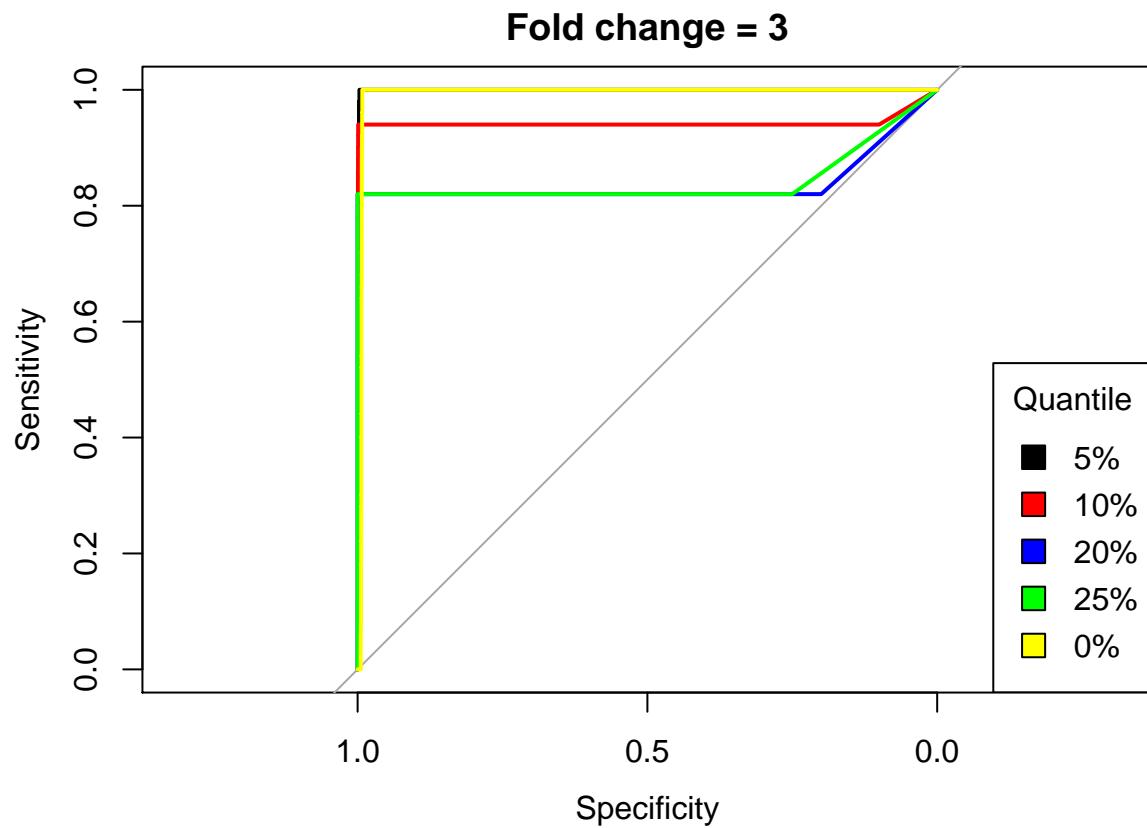


```
rm("hic.table")
hic.table <- backup.table
quant0 <- make_roc(hic.table, N = 50, FC = 3, quant = 0)
```

MD Plot



```
plot.colors <- c('black', 'red', 'blue', 'green', 'yellow', 'orange', 'purple', 'lightblue')
plot(quant5, main = paste0('Fold change = ', 3))
plot(quant10, add = TRUE, col = plot.colors[2])
plot(quant20, add = TRUE, col = plot.colors[3])
plot(quant25, add = TRUE, col = plot.colors[4])
plot(quant0, add = TRUE, col = plot.colors[5])
legend('bottomright', inset = 0, legend = c('5%', '10%', '20%', '25%', '0%'), title = 'Quantile', fill ...
```

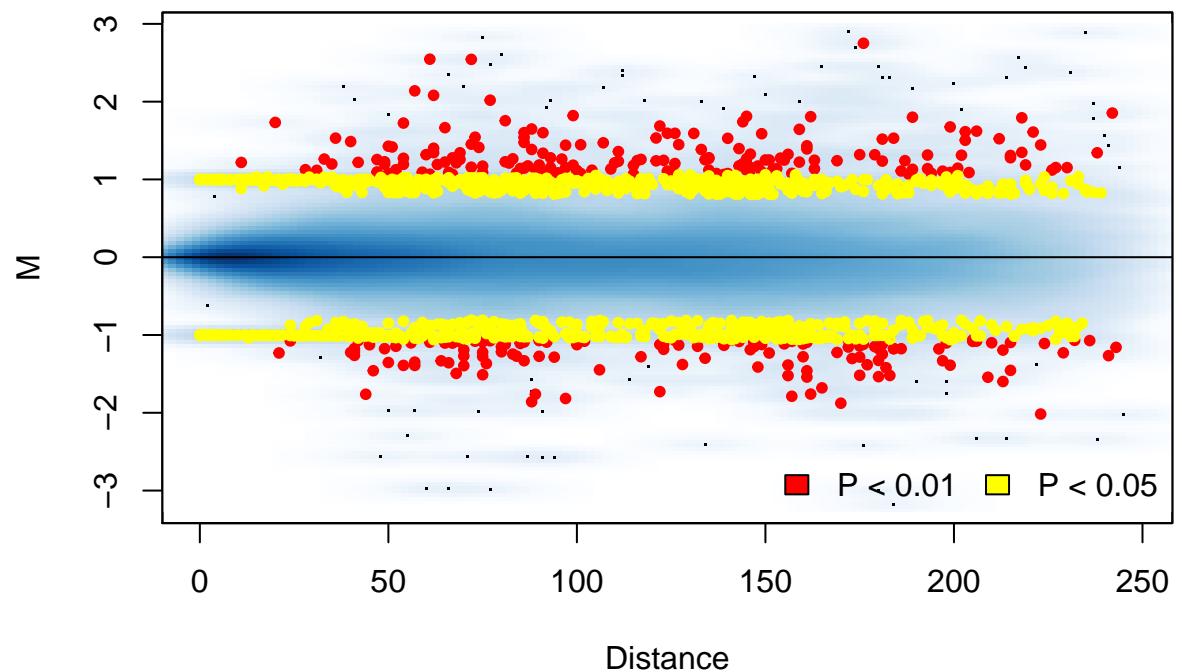


500 changes 2 FC

```
rm("hic.table")
hic.table <- dplfc1_2[[1]]
backup.table <- hic.table

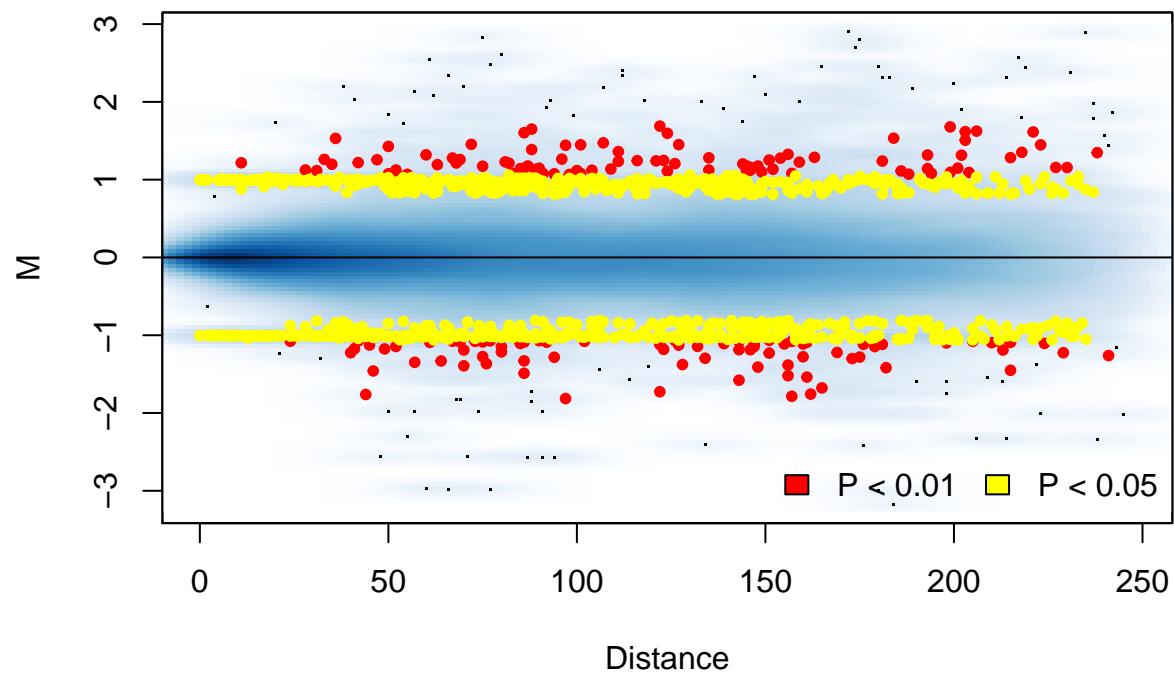
quant5 <- make_roc(hic.table, N = 500, FC = 2, quant = 0.05)
```

MD Plot



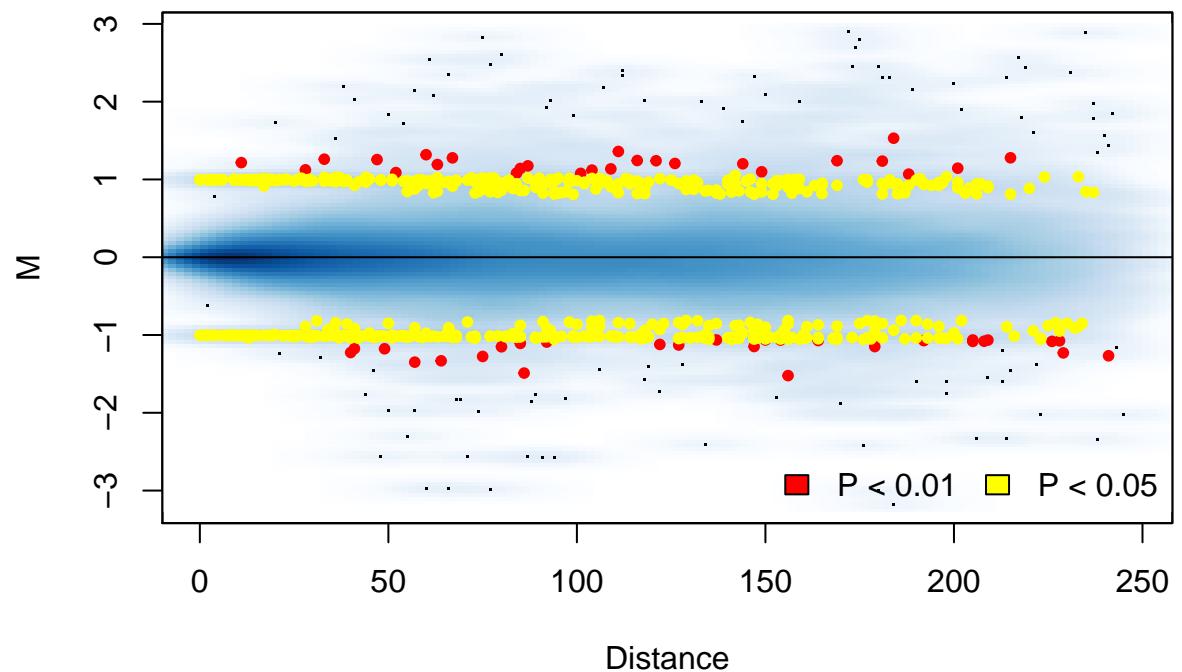
```
rm("hic.table")
hic.table <- backup.table
quant10 <- make_roc(hic.table, N = 500, FC = 2, quant = 0.1)
```

MD Plot



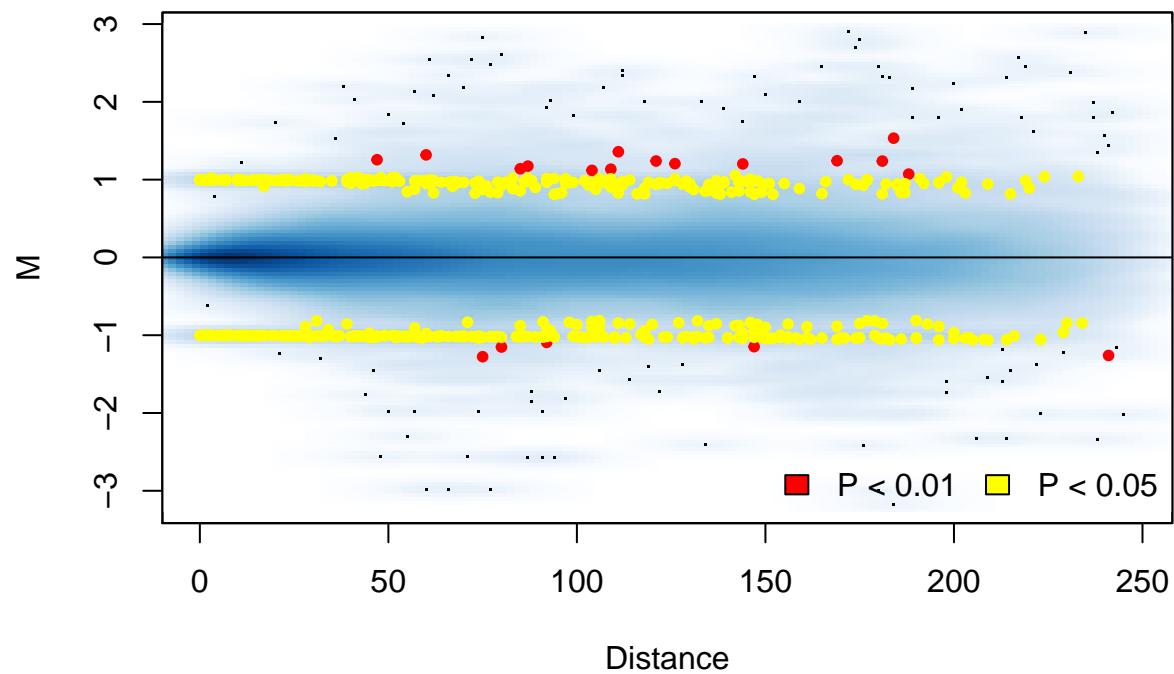
```
rm("hic.table")
hic.table <- backup.table
quant20 <- make_roc(hic.table, N = 500, FC = 2, quant = 0.2)
```

MD Plot



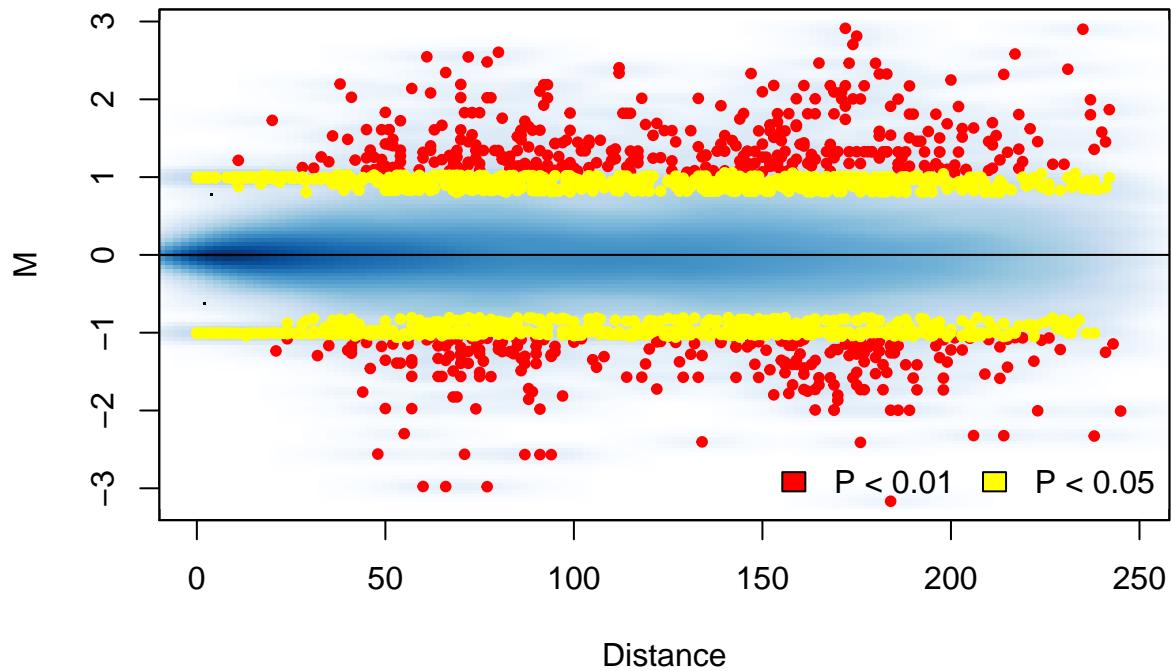
```
rm("hic.table")
hic.table <- backup.table
quant25 <- make_roc(hic.table, N = 500, FC = 2, quant = 0.25)
```

MD Plot

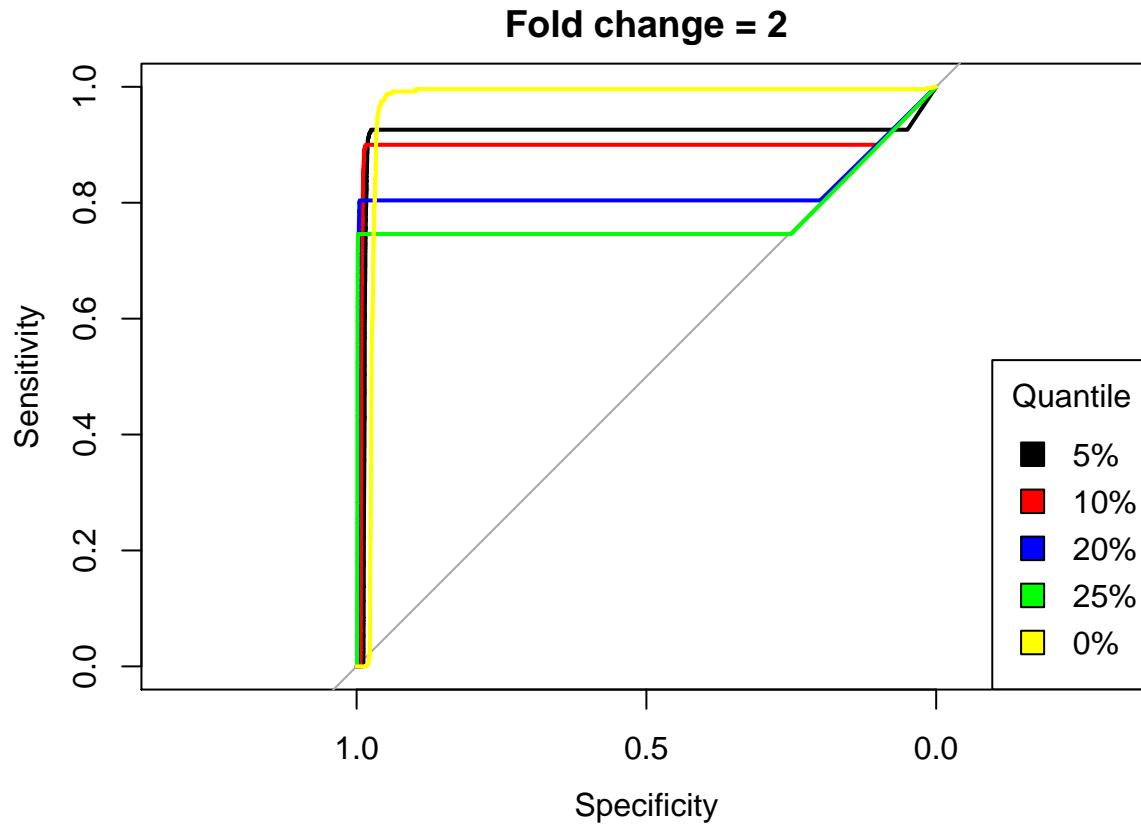


```
rm("hic.table")
hic.table <- backup.table
quant0 <- make_roc(hic.table, N = 500, FC = 2, quant = 0)
```

MD Plot



```
plot.colors <- c('black', 'red', 'blue', 'green', 'yellow', 'orange', 'purple', 'lightblue')
plot(quant5, main = paste0('Fold change = ', 2))
plot(quant10, add = TRUE, col = plot.colors[2])
plot(quant20, add = TRUE, col = plot.colors[3])
plot(quant25, add = TRUE, col = plot.colors[4])
plot(quant0, add = TRUE, col = plot.colors[5])
legend('bottomright', inset = 0, legend = c('5%', '10%', '20%', '25%', '0%'), title = 'Quantile', fill ...
```



Test holm p-value adjustment by distance

```

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 = FALSE)

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

```

```

    # return(roc_result)
    return(result)
}

```

N = 300 FC = 1.5

```

hic.table <- backup.table

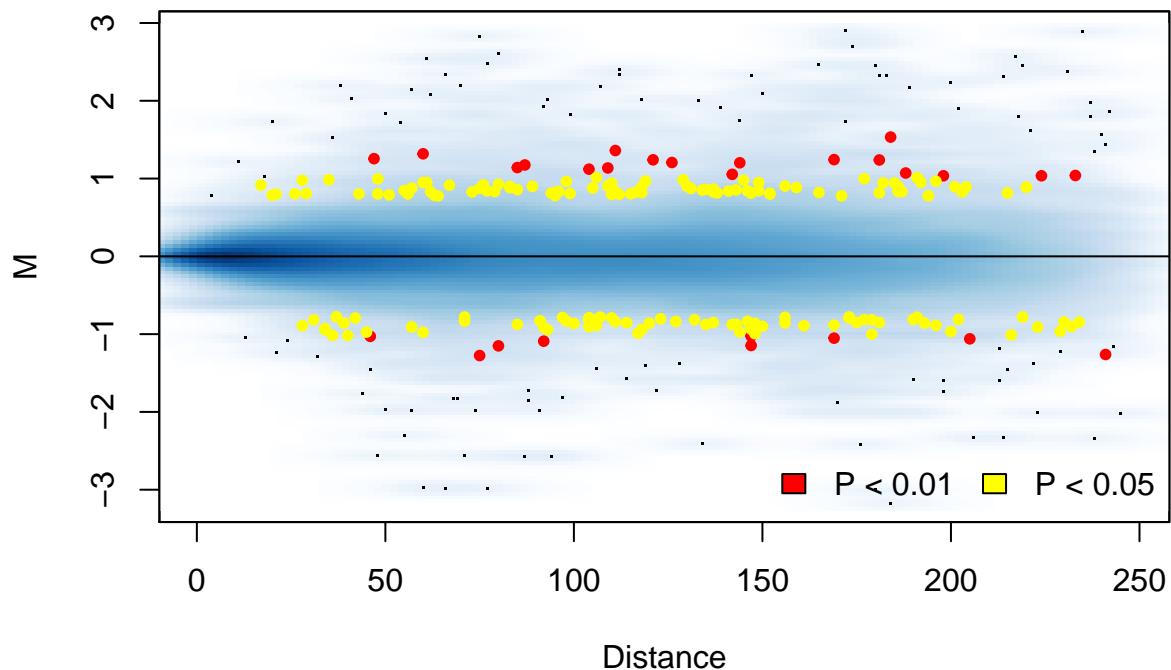
hic.table <- make_changes(hic.table, N = 300, FC = 1.5, quant = 0.25)

## 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.

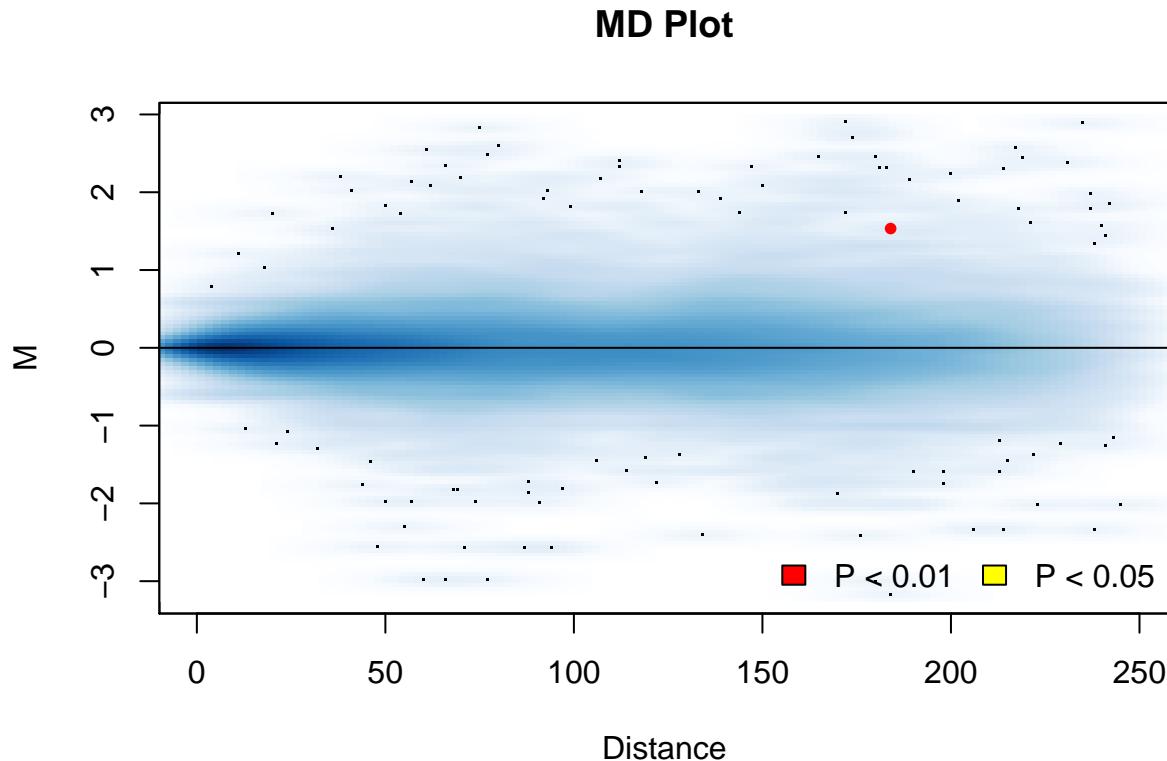
## Span for loess: 0.899918985683651
## GCV for loess: 6.06422104146962e-06
## AIC for loess: -0.853236376517183

```

MD Plot



```
hic.table <- holm_adjust(hic.table)
```



N = 300 FC = 2

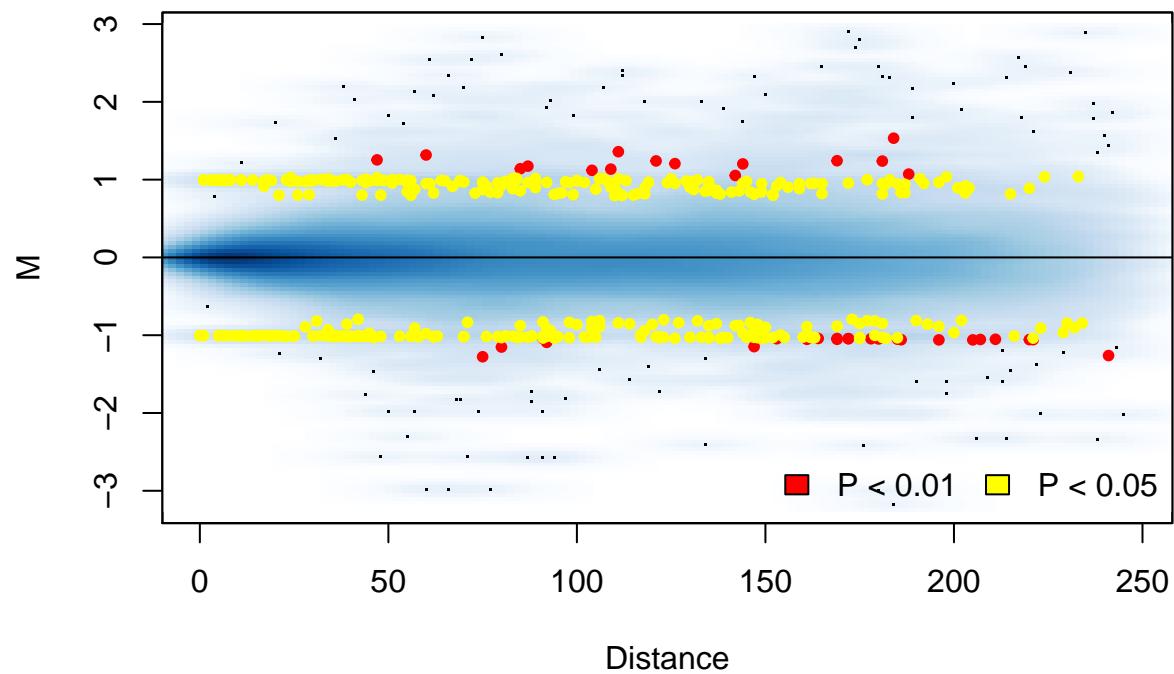
```
hic.table <- backup.table

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.

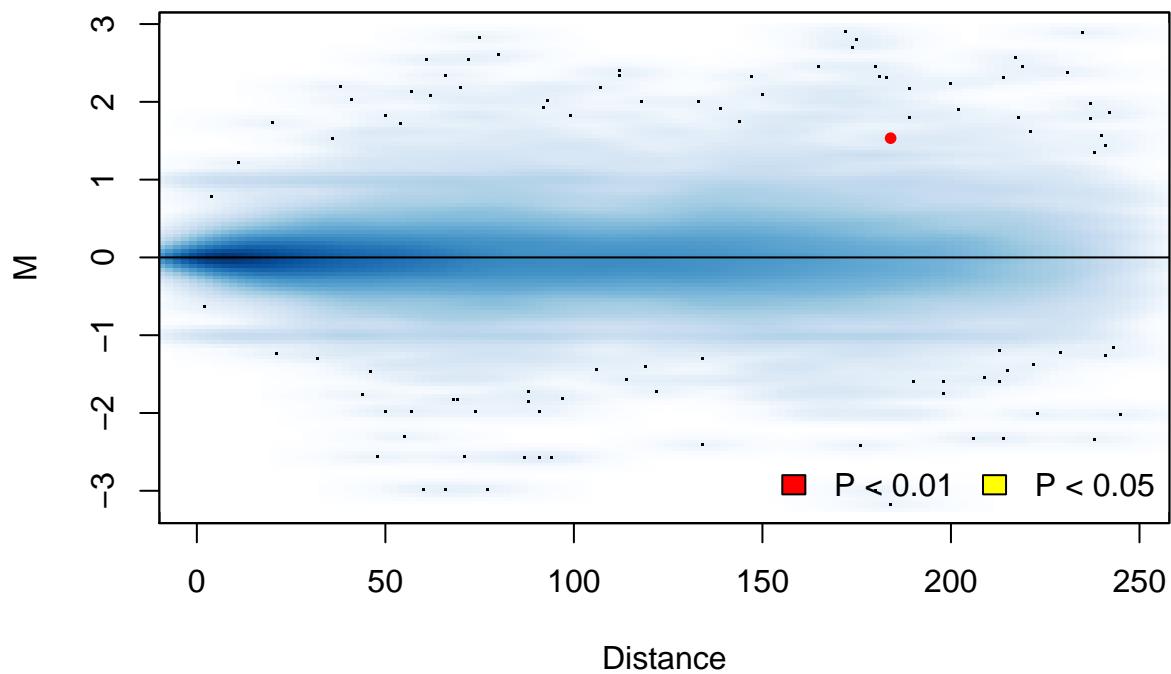
## Span for loess: 0.899698836045662
## GCV for loess: 6.32757620673592e-06
## AIC for loess: -0.810725172244512
```

MD Plot



```
hic.table <- holm_adjust(hic.table)
```

MD Plot



$N = 300$ $FC = 3$

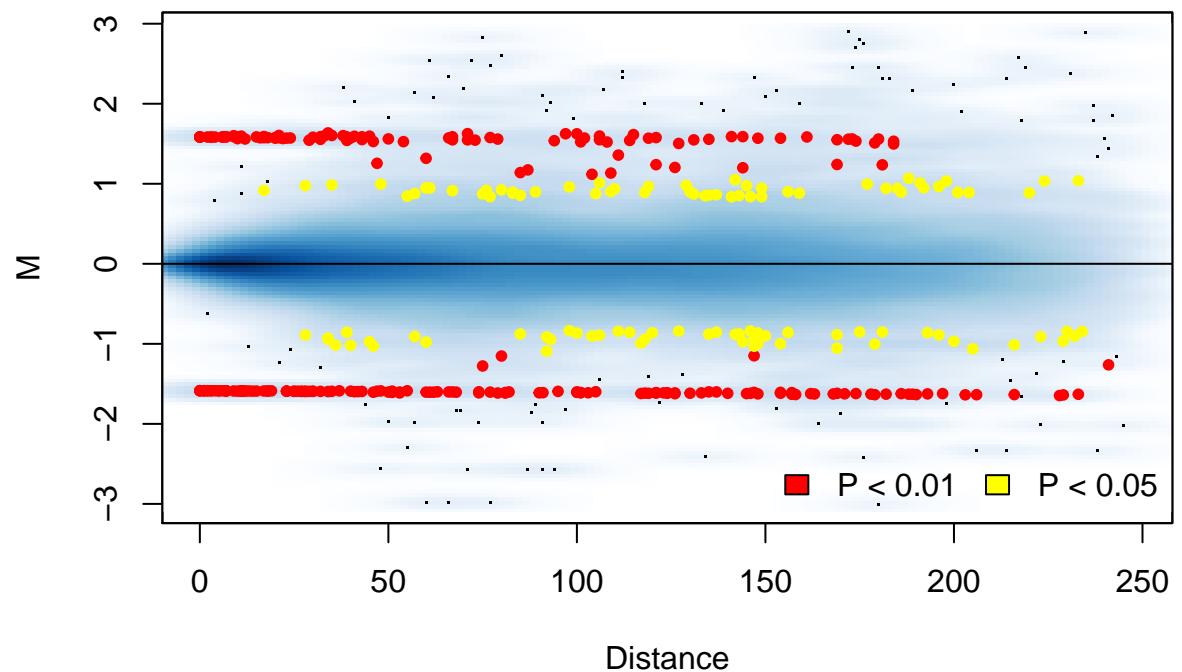
```
hic.table <- backup.table

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.

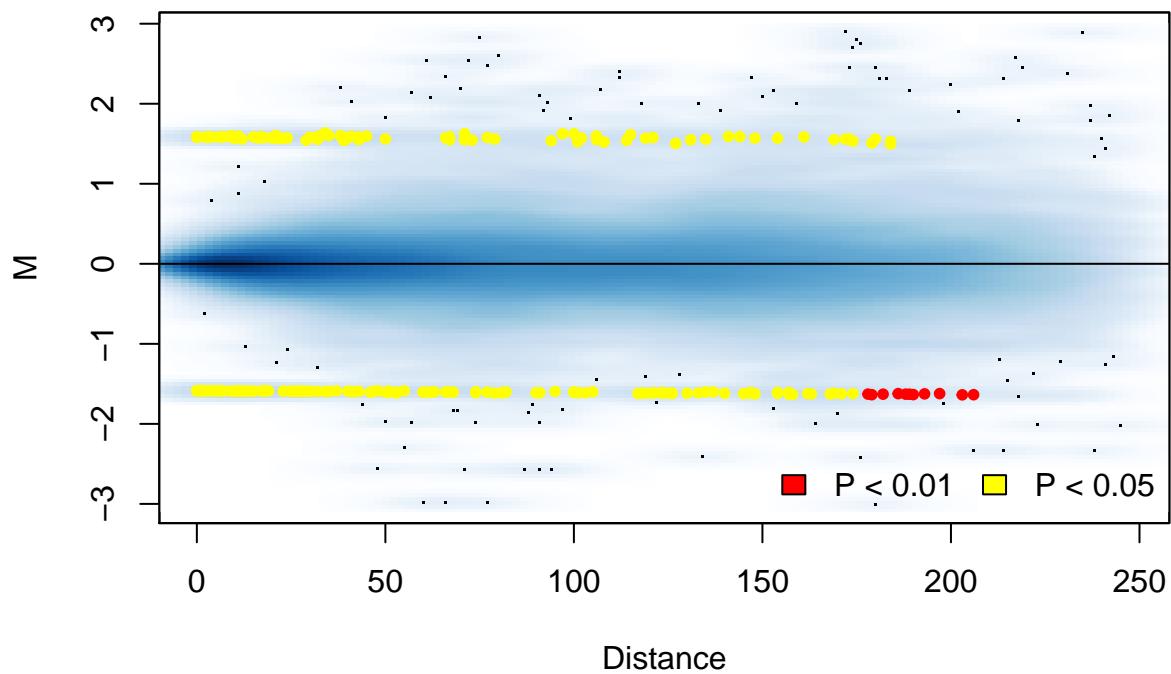
## Span for loess: 0.899945506167573
## GCV for loess: 7.00761199221638e-06
## AIC for loess: -0.70864549550216
```

MD Plot



```
hic.table <- holm_adjust(hic.table)
```

MD Plot



$N = 300 \text{ FC} = 4$

```

hic.table <- backup.table

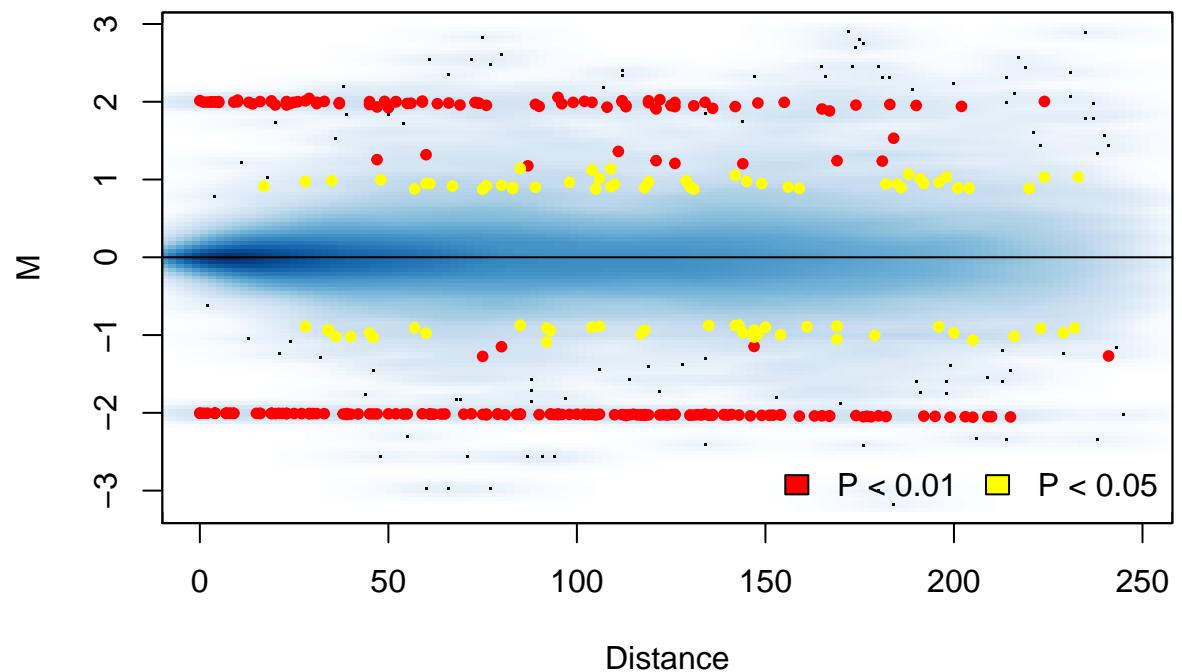
hic.table <- make_changes(hic.table, N = 300, FC = 4, quant = 0.25)

## 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: 0.899713488511402
## GCV for loess: 7.65533686917287e-06
## AIC for loess: -0.620239397860361

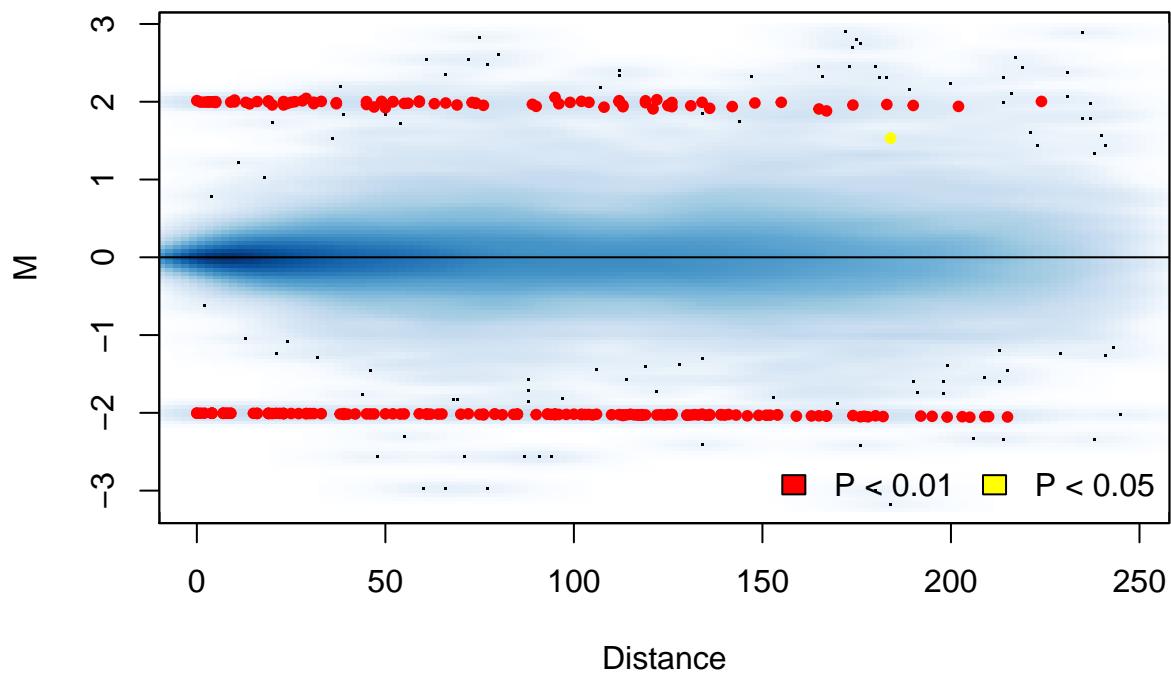
```

MD Plot



```
hic.table <- holm_adjust(hic.table)
```

MD Plot



N = 50 FC = 3

```

hic.table <- backup.table

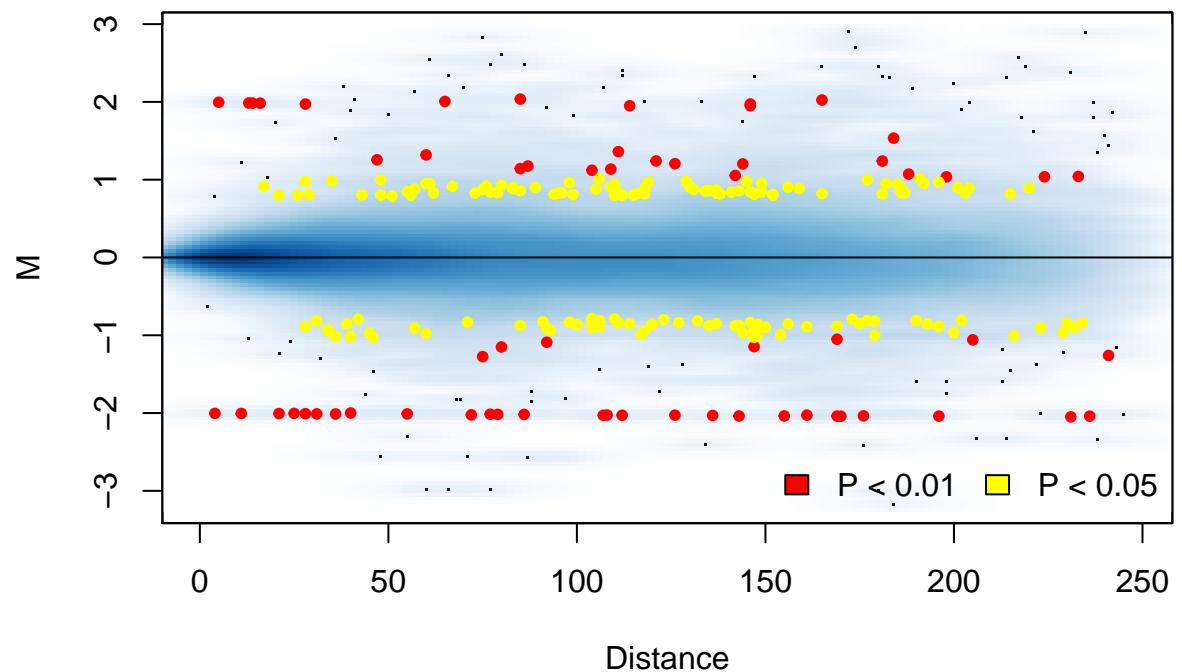
hic.table <- make_changes(hic.table, N = 50, FC = 4, quant = 0.25)

## 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: 0.899929958808223
## GCV for loess: 6.24910864241823e-06
## AIC for loess: -0.823203641564267

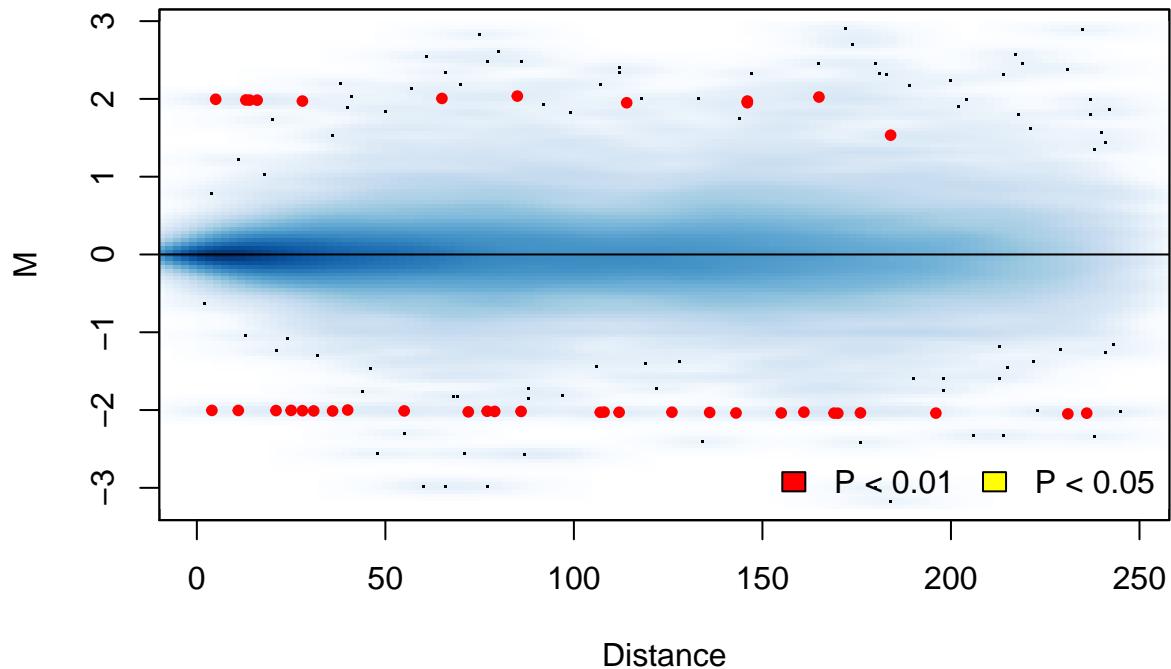
```

MD Plot



```
hic.table <- holm_adjust(hic.table)
```

MD Plot



$N = 50$ $FC = 4$

```

hic.table <- backup.table

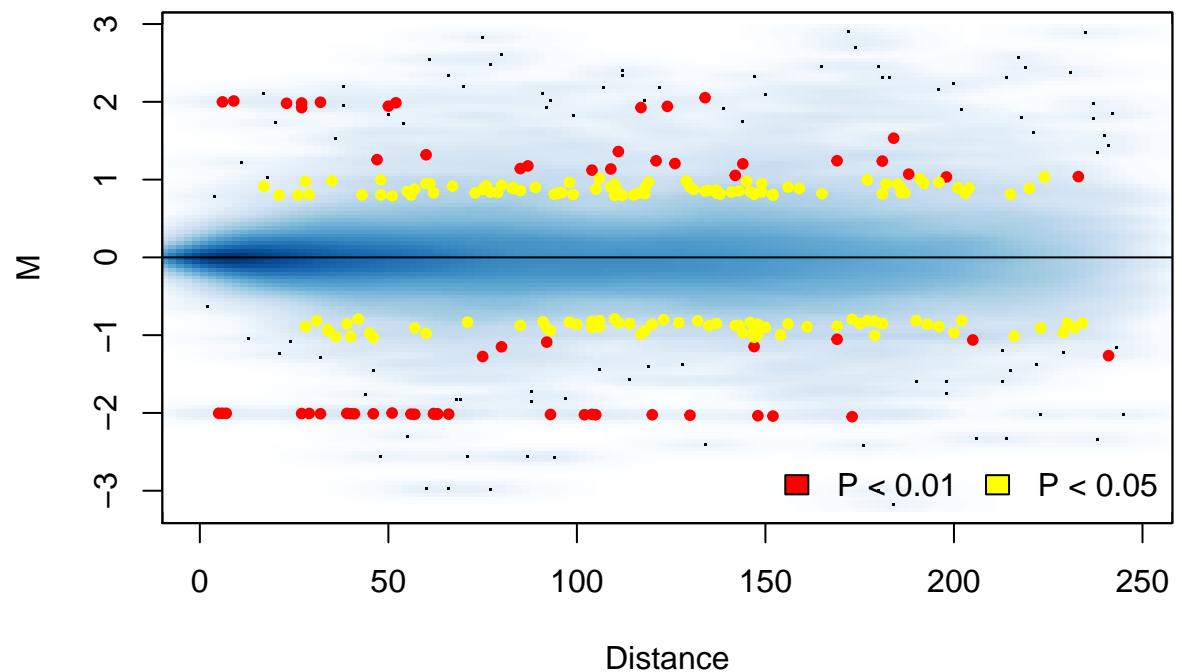
hic.table <- make_changes(hic.table, N = 50, FC = 4, quant = 0.25)

## 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: 0.899931087702247
## GCV for loess: 6.24723480931513e-06
## AIC for loess: -0.823503542828459

```

MD Plot



```
hic.table <- holm_adjust(hic.table)
```

MD Plot

