

# 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[, 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)
}

# pvalue adjustment - FDR by distance
fdr_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 = 'fdr')]
    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)
}

```

```

calc_stats1 <- function(hic.table) {
  TP <- sum(hic.table$truth == 1 & hic.table$p.val < 0.05)
  TN <- sum(hic.table$truth == 0 & hic.table$p.val > 0.05)
  FP <- sum(hic.table$truth == 0 & hic.table$p.val < 0.05)
  FN <- sum(hic.table$truth == 1 & hic.table$p.val > 0.05)
  TPR <- TP / (TP + FN)
  TNR <- TN / (TN + FP)
  PPV <- TP / (TP + FP)
  NPV <- TN / (TN + FN)
  FPR <- FP / (FP + TN)
  FNR <- FN / (TP + FN)
  FDR <- FP / (TP + FP)
  ACC <- (TP + TN) / (TP + FP + FN + TN)
  res <- data.frame(TPR = TPR, TNR = TNR, PPV = PPV, NPV = NPV, FPR = FPR, FNR = FNR, FDR = FDR, ACC = ACC)
  knitr::kable(res)
}

```

```

calc_stats <- function(hic.table) {
  TP <- sum(hic.table$truth == 1 & hic.table$p.adj < 0.05)
  TN <- sum(hic.table$truth == 0 & hic.table$p.adj > 0.05)
  FP <- sum(hic.table$truth == 0 & hic.table$p.adj < 0.05)
  FN <- sum(hic.table$truth == 1 & hic.table$p.adj > 0.05)
  TPR <- TP / (TP + FN)
  TNR <- TN / (TN + FP)
  PPV <- TP / (TP + FP)
  NPV <- TN / (TN + FN)
  FPR <- FP / (FP + TN)
  FNR <- FN / (TP + FN)
  FDR <- FP / (TP + FP)
  ACC <- (TP + TN) / (TP + FP + FN + TN)
  res <- data.frame(TPR = TPR, TNR = TNR, PPV = PPV, NPV = NPV, FPR = FPR, FNR = FNR, FDR = FDR, ACC = ACC)
  knitr::kable(res)
}

```

## 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)

```

```

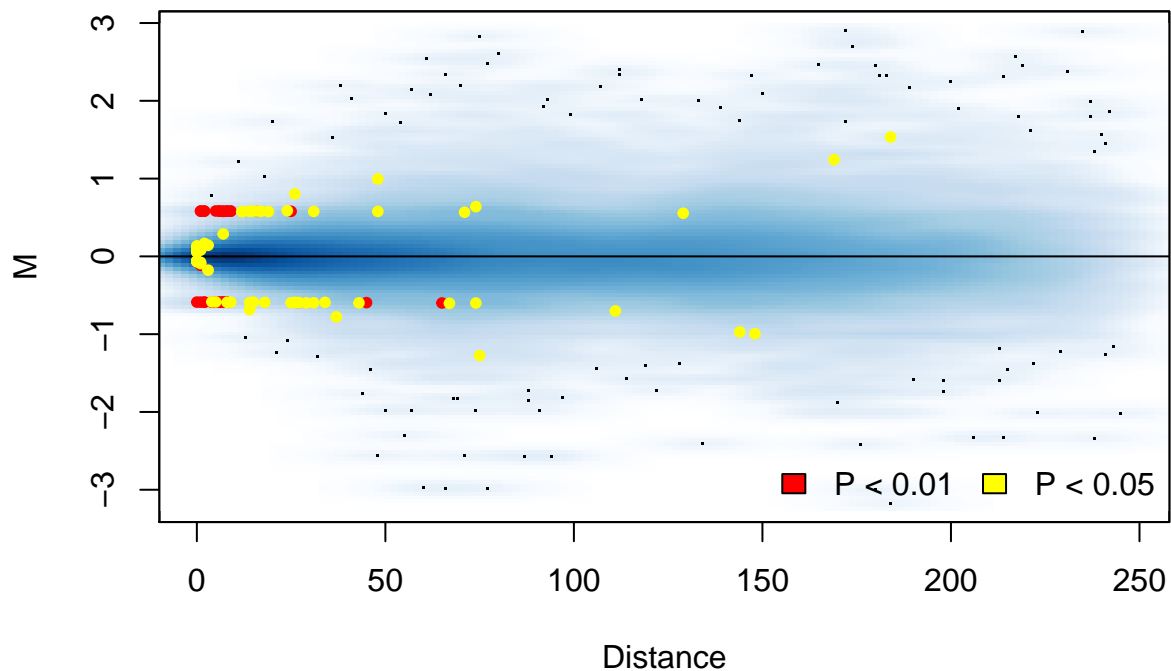
## Warning in `[<-.data.table`(`*tmp*`, changes, , value = structure(list(chr1

```

```
## = 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: 0.899945506167573
## GCV for loess: 6.05389485492863e-06
## AIC for loess: -0.854940638615042
```

## MD Plot

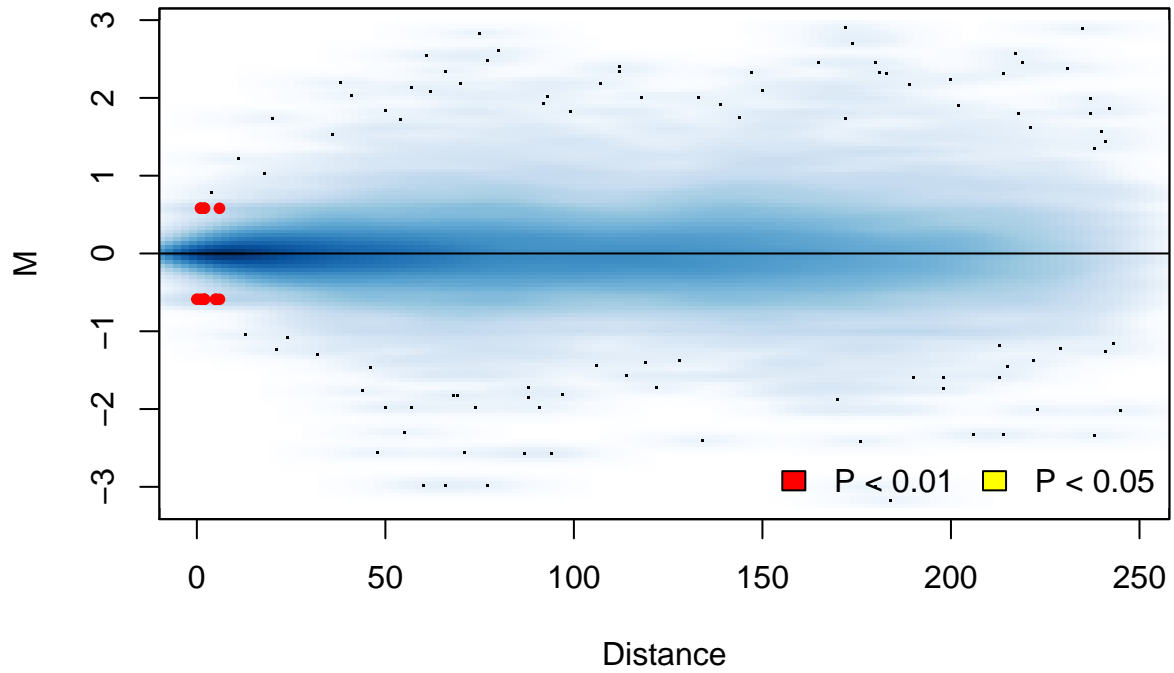


```
## True positive rate:
calc_stats1(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.21	0.9983161	0.5943396	0.990789	0.0016839	0.79	0.4056604	0.9891624

```
# FDR on full vector
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

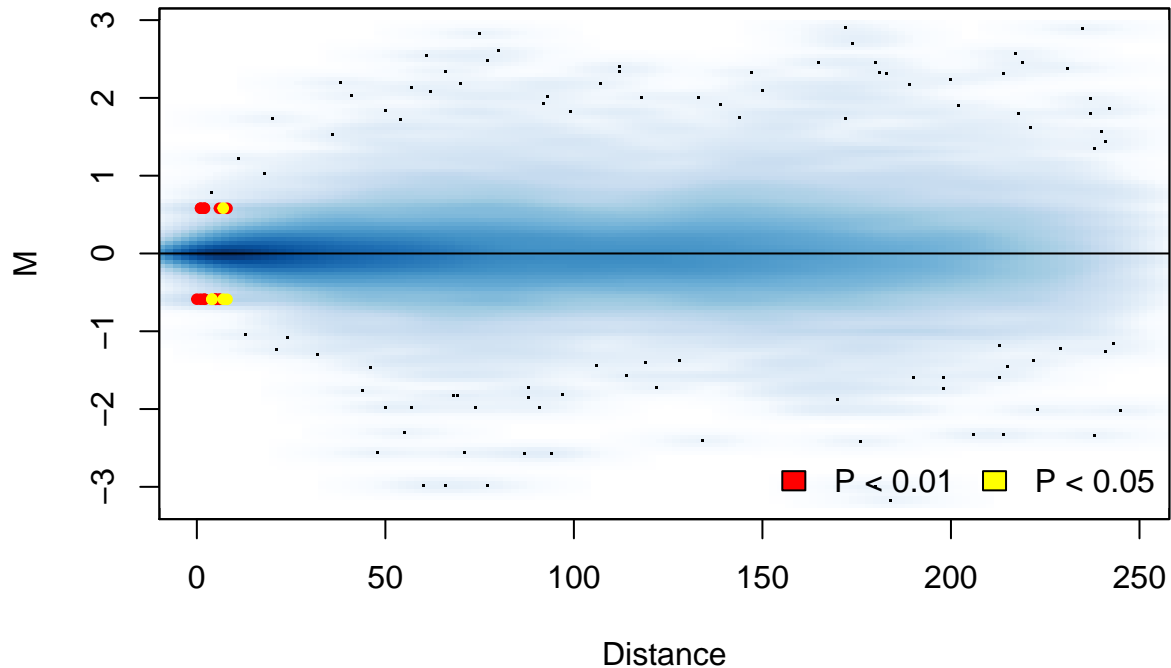


```
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.04	1	1	0.9888476	0	0.96	0	0.9888528

```
# FDR by distance
hic.table <- fdr_adjust(hic.table)
```

## MD Plot



```
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.0666667	1	1	0.989154	0	0.9333333	0	0.9891624

**N = 300 FC = 2**

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

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

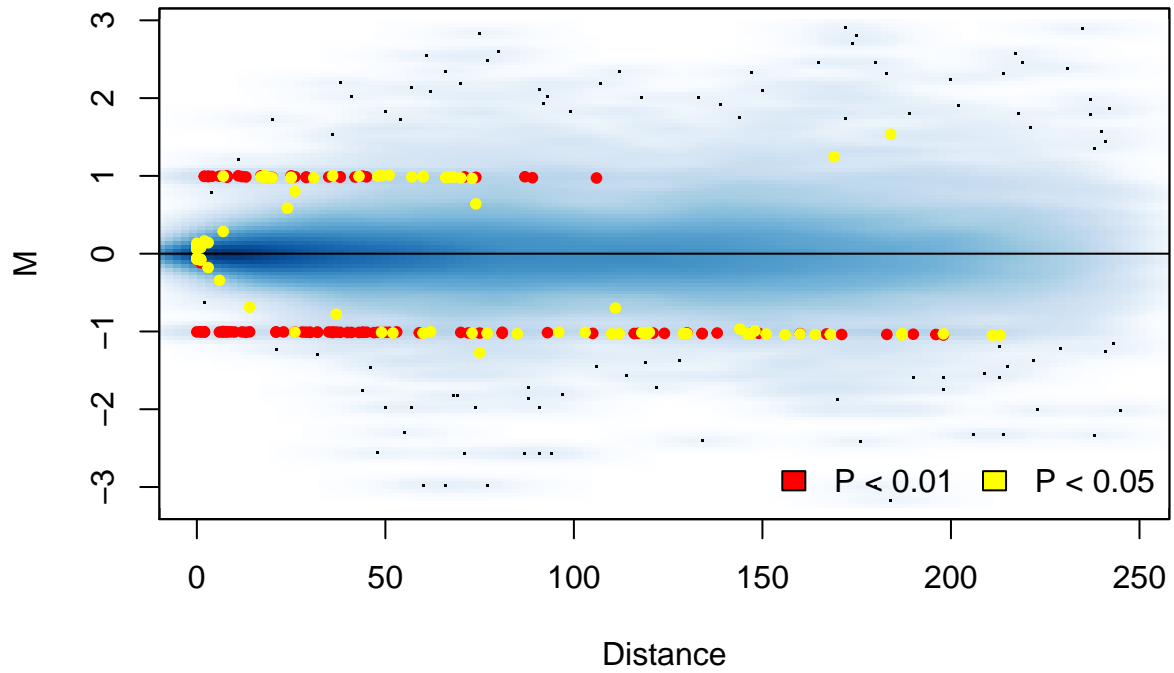
```
## Warning in `[<-.data.table`(`*tmp*`, changes, , value = structure(list(chr1
## = c("chr1", : Coerced 'double' RHS to 'integer' to match the column's
## type; may have truncated precision. Either change the target column to
## 'double' first (by creating a new 'double' vector length 25836 (nrows 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: 0.899936573127168
```

```
## GCV for loess: 6.32672292842298e-06
```

```
## AIC for loess: -0.810860082048539
```

## MD Plot

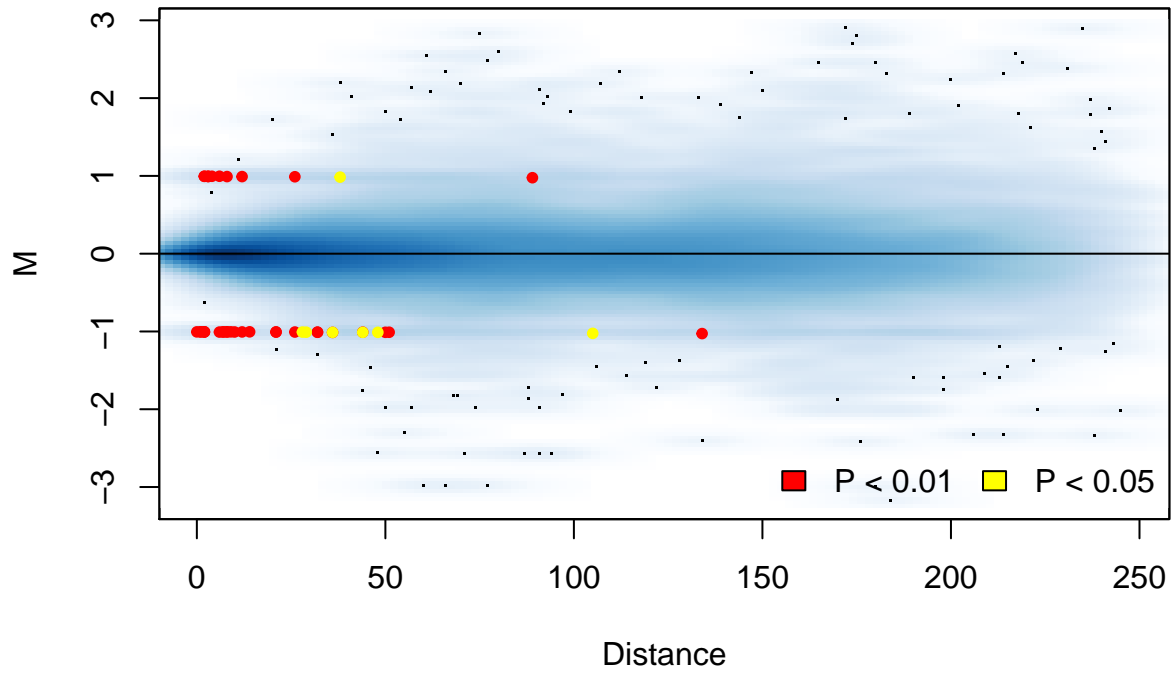


```
## True positive rate:
calc_stats1(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.51	0.9982769	0.7766497	0.9942665	0.0017231	0.49	0.2233503	0.9926072

```
# FDR on full vector
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

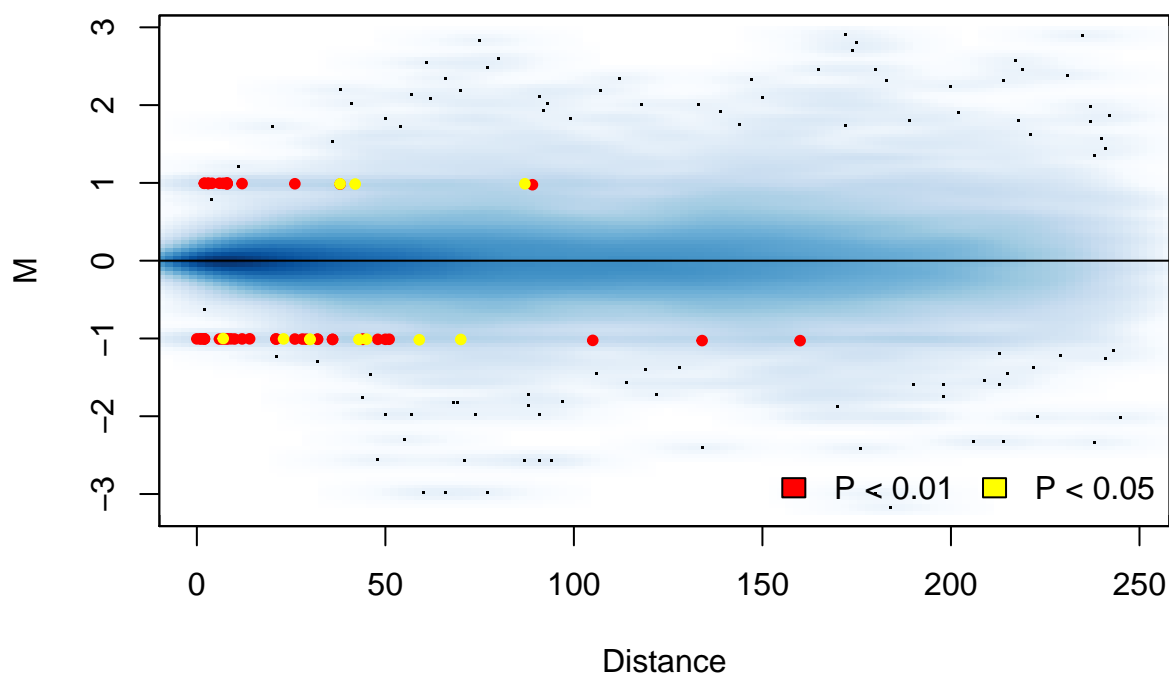


```
## True positive rate:
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.1433333	1	1	0.9900361	0	0.8566667	0	0.9900526

```
# FDR by distance
hic.table <- fdr_adjust(hic.table)
```

## MD Plot



```
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.1966667	1	1	0.9906506	0	0.8033333	0	0.9906719

**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.
```

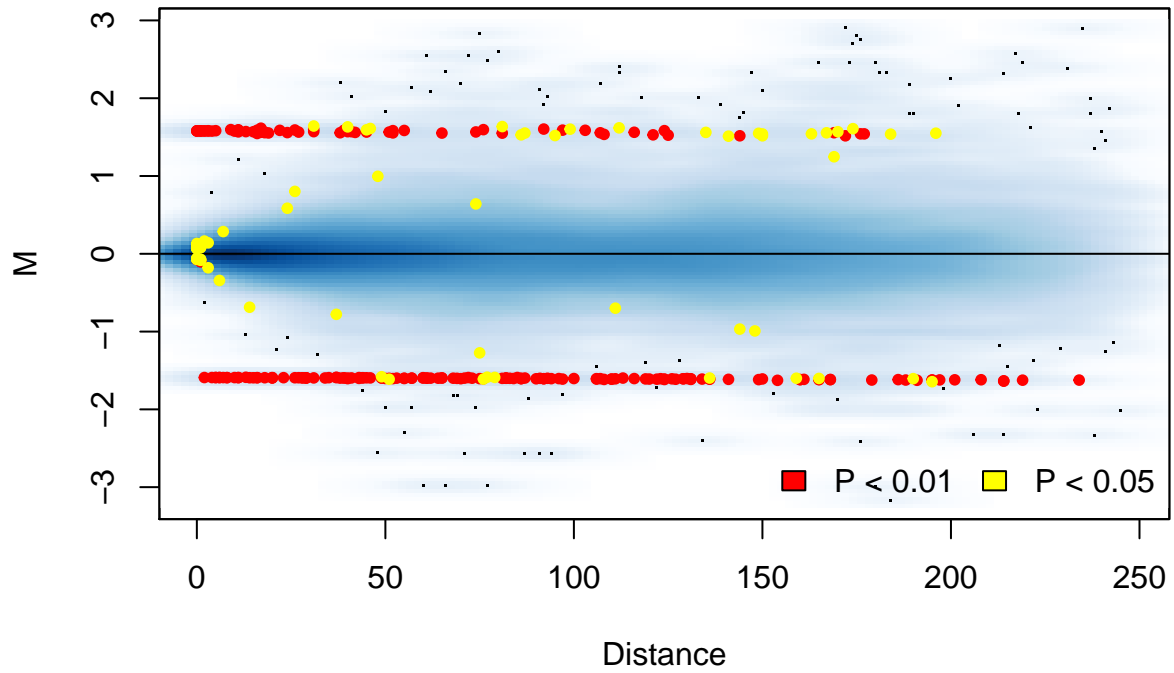
```
## Span for loess: 0.898969769851922
```

```
## GCV for loess: 7.01961161203821e-06
```

```
## AIC for loess: -0.706934385337706
```



## MD Plot

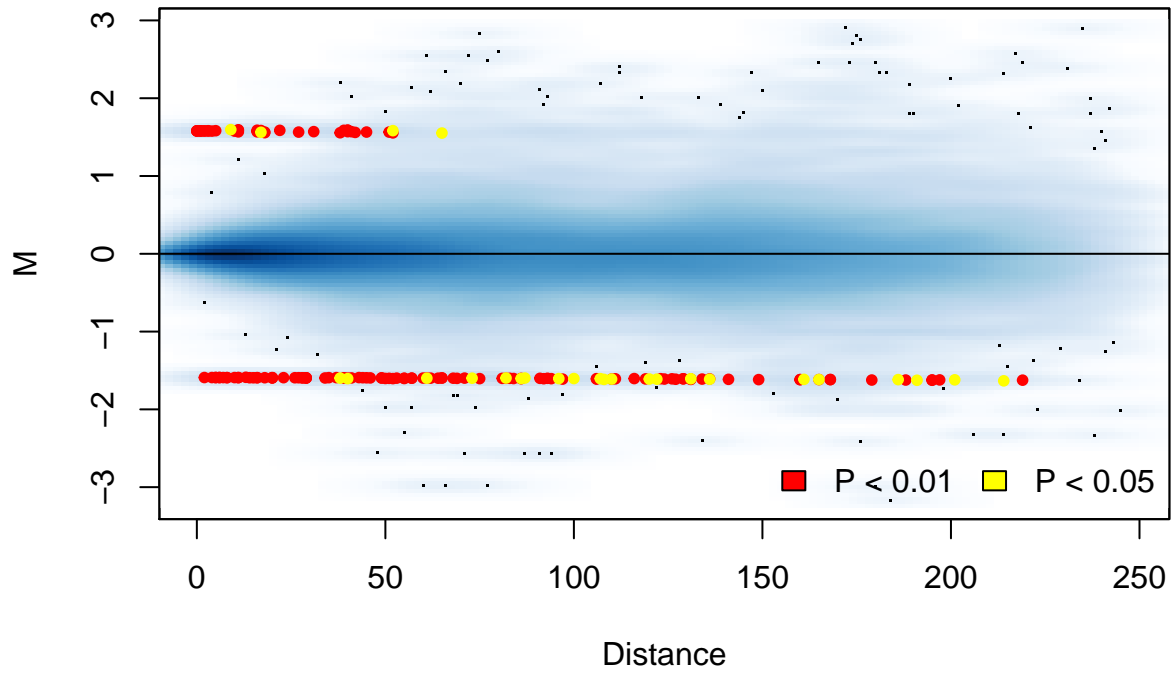


```
## True positive rate:
calc_stats1(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.79	0.9983553	0.8494624	0.9975349	0.0016447	0.21	0.1505376	0.9959359

```
# FDR on full vector
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

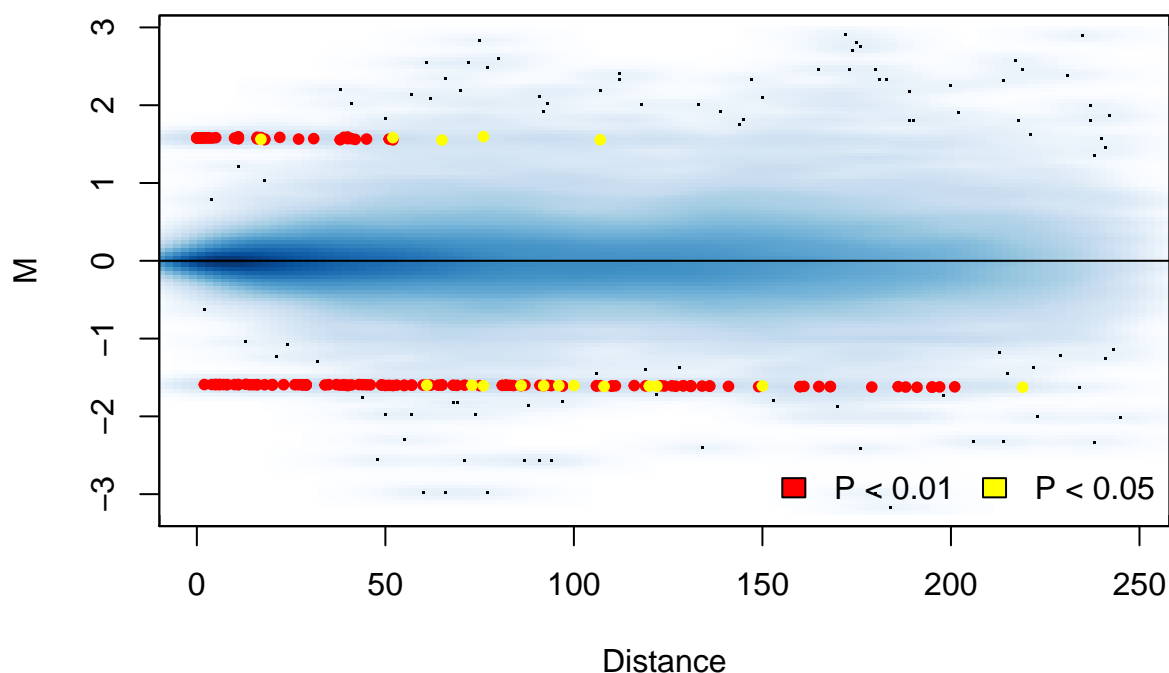


```
## True positive rate:
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.5333333	1	1	0.9945474	0	0.4666667	0	0.9945812

```
# FDR by distance
hic.table <- fdr_adjust(hic.table)
```

## MD Plot



```
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.5433333	1	1	0.9946637	0	0.4566667	0	0.9946973

$N = 300$   $FC = 4$

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

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

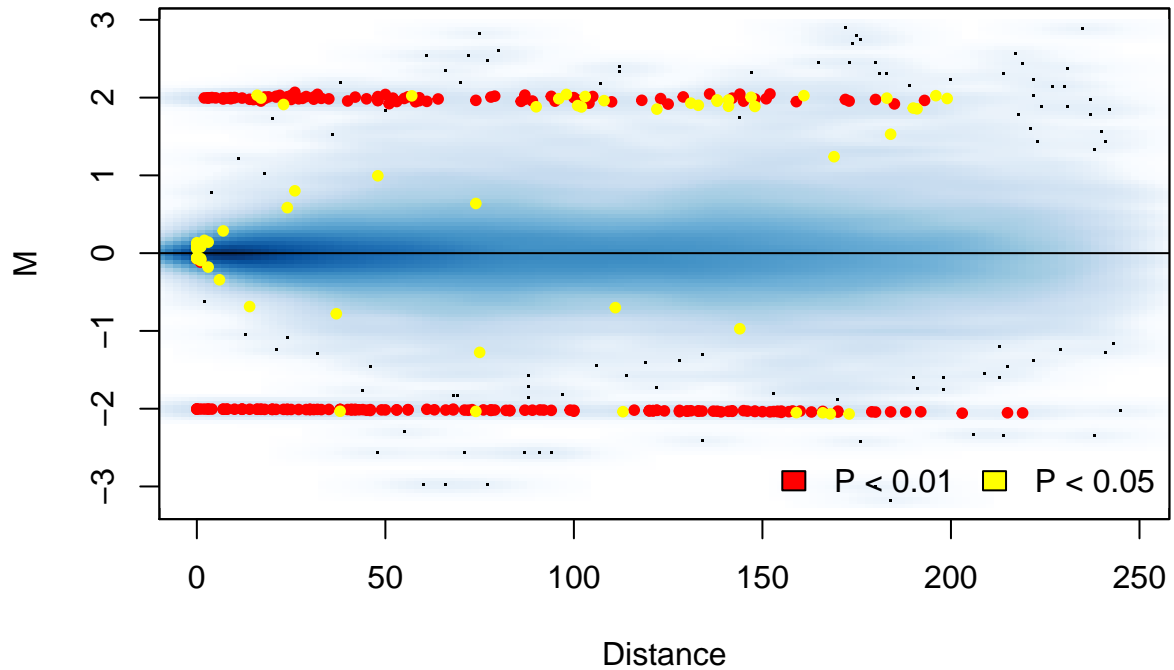
```
## Warning in `[<-.data.table`(`*tmp*`, changes, , value = structure(list(chr1
## = c("chr1", : Coerced 'double' RHS to 'integer' to match the column's
## type; may have truncated precision. Either change the target column to
## 'double' first (by creating a new 'double' vector length 25836 (nrows 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: 0.899708073618214
```

```
## GCV for loess: 7.67006491516724e-06
```

```
## AIC for loess: -0.618317352427572
```

## MD Plot

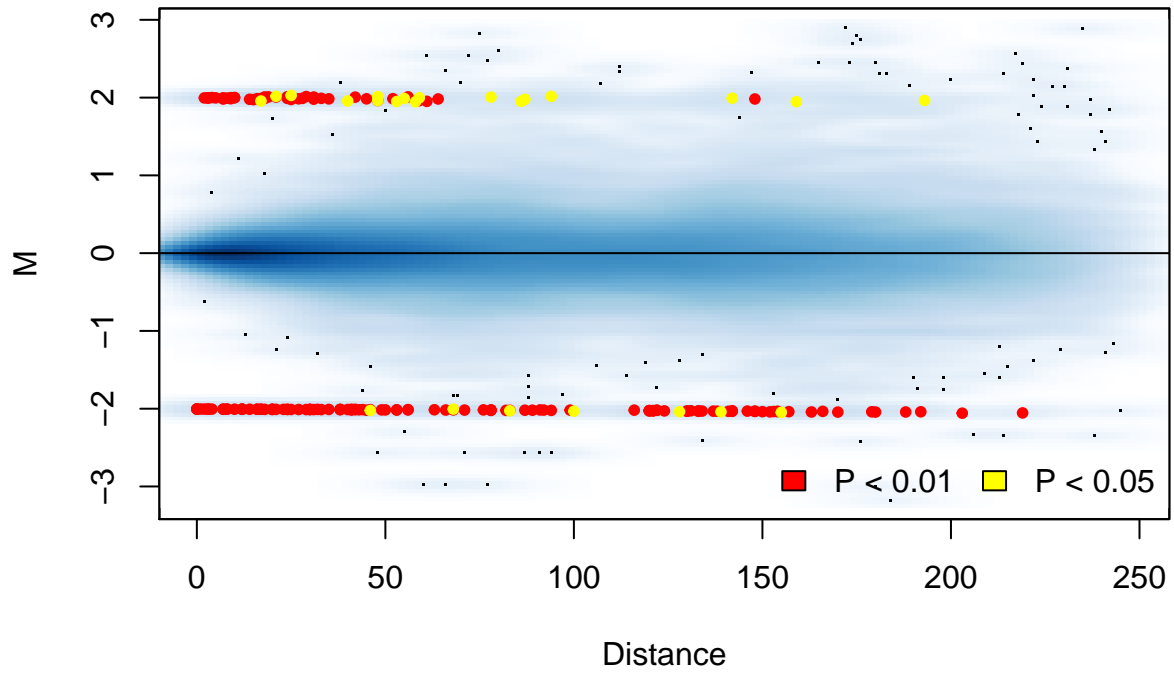


```
## True positive rate:
calc_stats1(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.8533333	0.9983553	0.8590604	0.9982771	0.0016447	0.1466667	0.1409396	0.9966713

```
# FDR on full vector
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

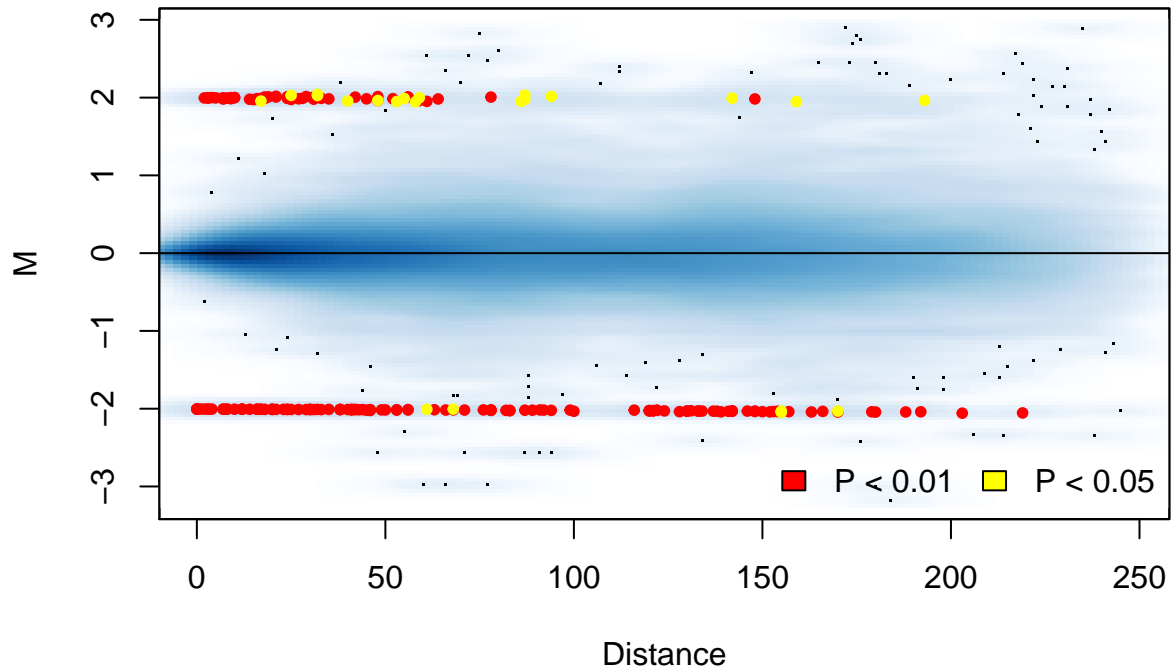


```
## True positive rate:
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.5933333	1	1	0.9952451	0	0.4066667	0	0.9952779

```
# FDR by distance
hic.table <- fdr_adjust(hic.table)
```

## MD Plot



```
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.6133333	1	1	0.9954779	0	0.3866667	0	0.9955101

$N = 50$   $FC = 3$

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

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

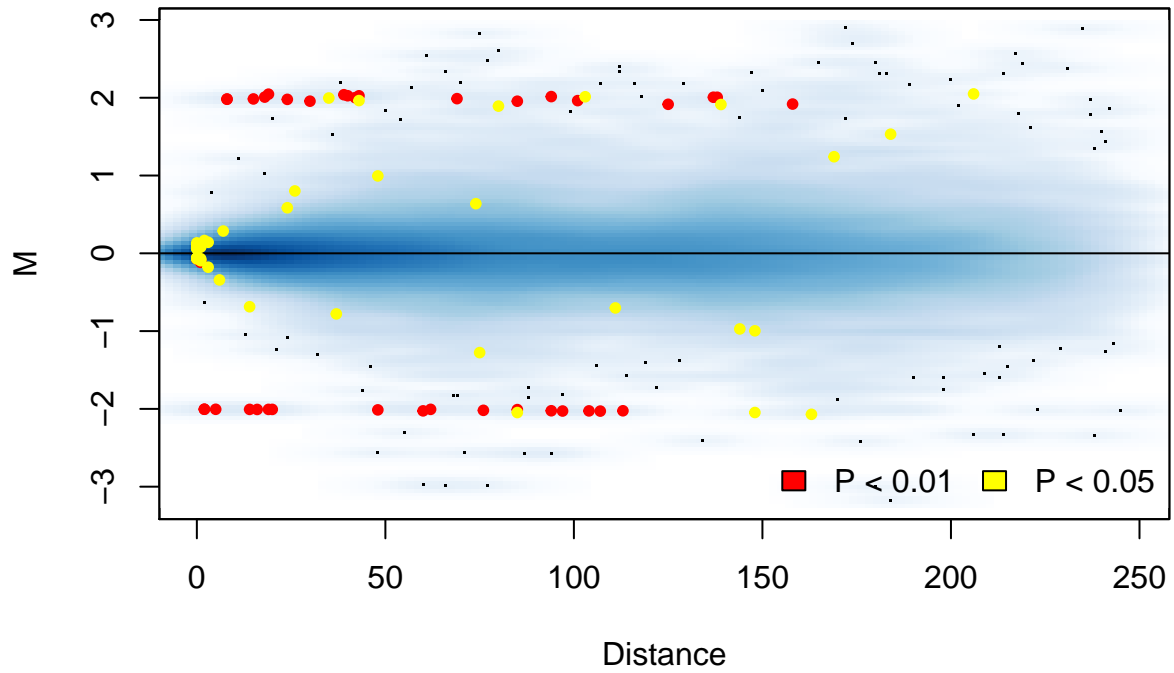
```
## Warning in `[<-.data.table`(`*tmp*`, changes, , value = structure(list(chr1
## = c("chr1", : Coerced 'double' RHS to 'integer' to match the column's
## type; may have truncated precision. Either change the target column to
## 'double' first (by creating a new 'double' vector length 25836 (nrows 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: 6.25156149692119e-06
```

```
## AIC for loess: -0.822811205877684
```

## MD Plot

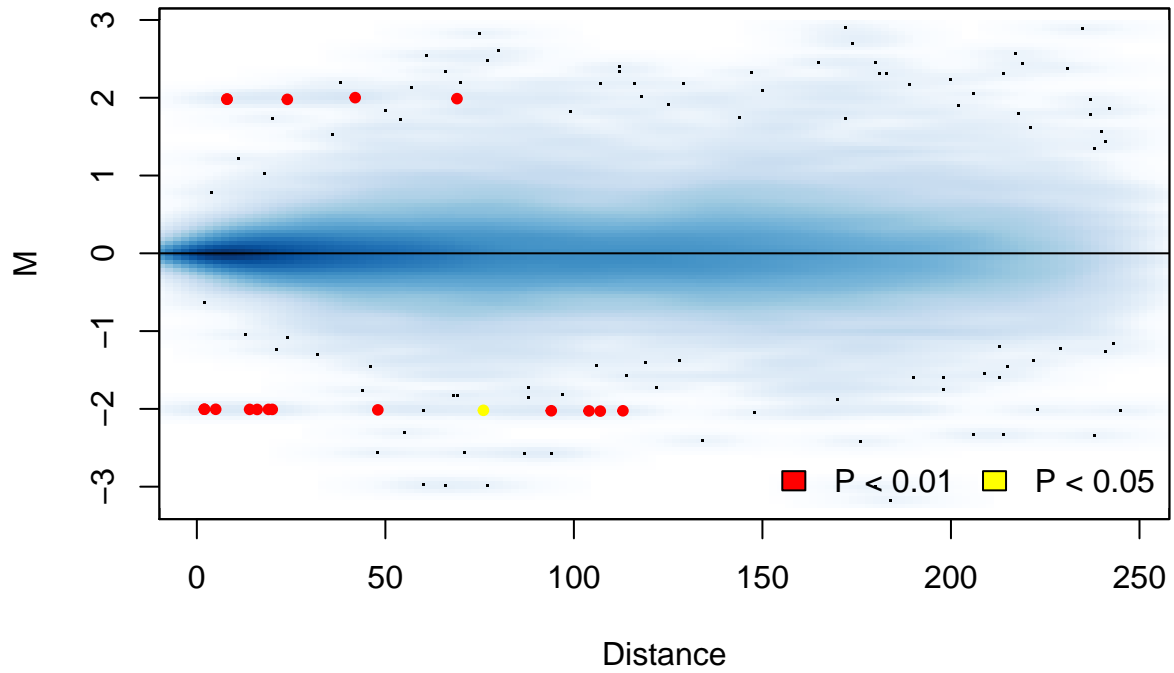


```
calc_stats1(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.9	0.9983324	0.5113636	0.9998058	0.0016676	0.1	0.4886364	0.9981421

```
# FDR on full vector
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



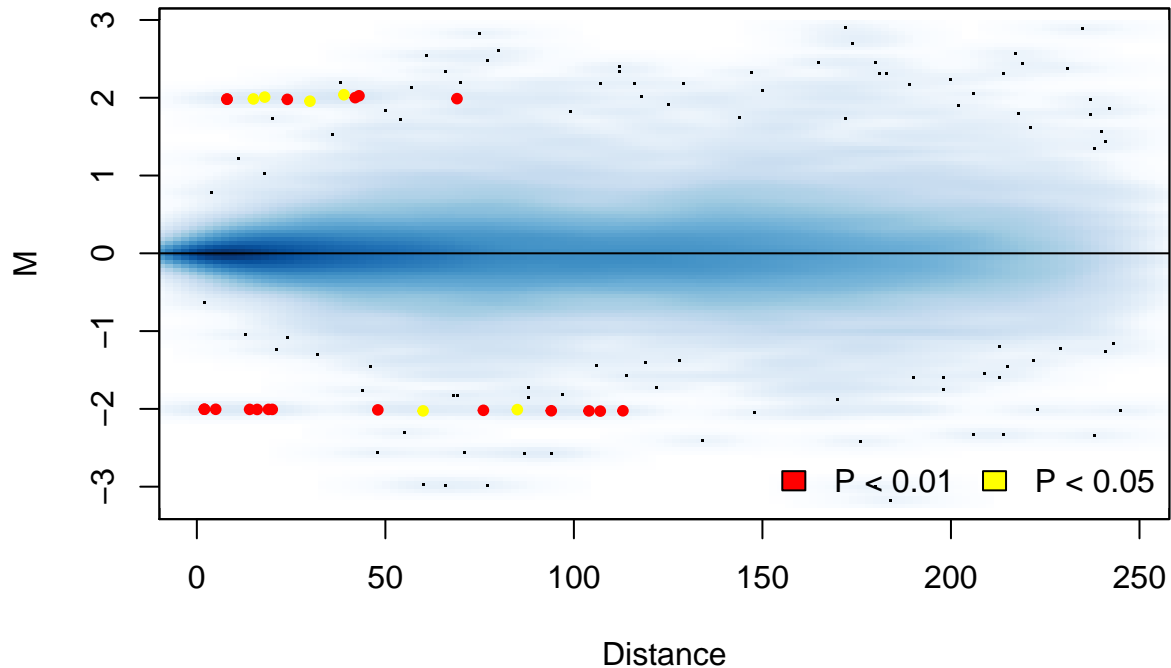
```
## True positive rate:
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.36	1	1	0.9987606	0	0.64	0	0.9987614

```
# FDR by distance
hic.table <- fdr_adjust(hic.table)
```



## MD Plot



```
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.5	1	1	0.9990314	0	0.5	0	0.9990324

$N = 50$   $FC = 4$

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

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

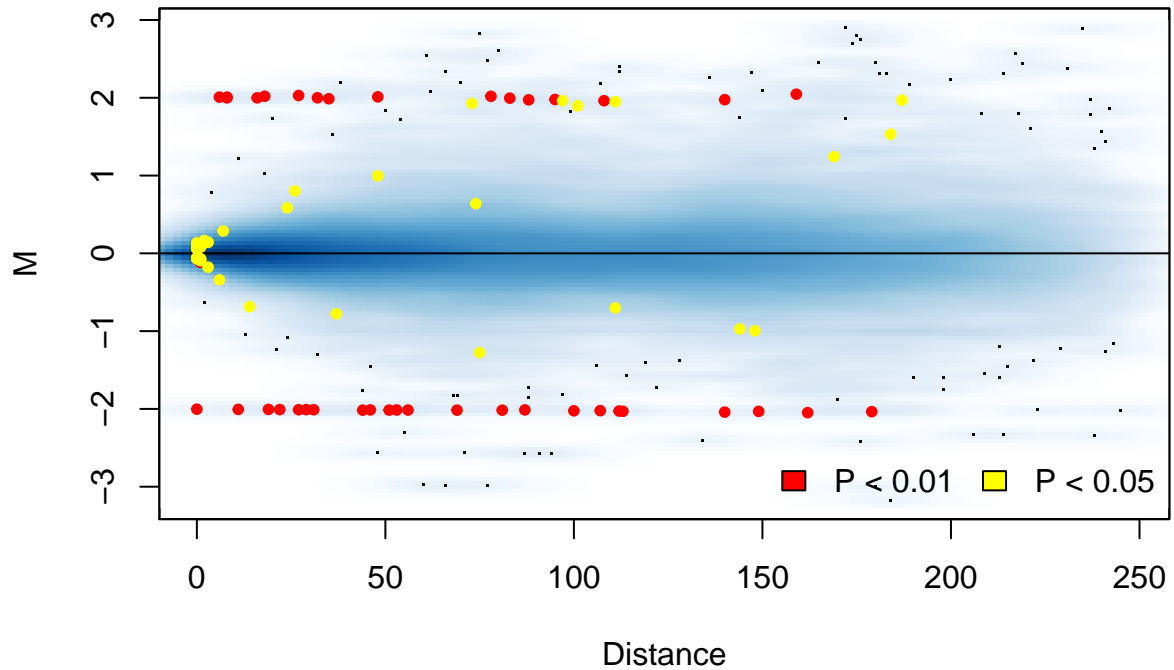
```
## Warning in `[<-.data.table`(`*tmp*`, changes, , value = structure(list(chr1
## = c("chr1", : Coerced 'double' RHS to 'integer' to match the column's
## type; may have truncated precision. Either change the target column to
## 'double' first (by creating a new 'double' vector length 25836 (nrows 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: 0.899945490871329
```

```
## GCV for loess: 6.258587563122e-06
```

```
## AIC for loess: -0.82168795043305
```

## MD Plot

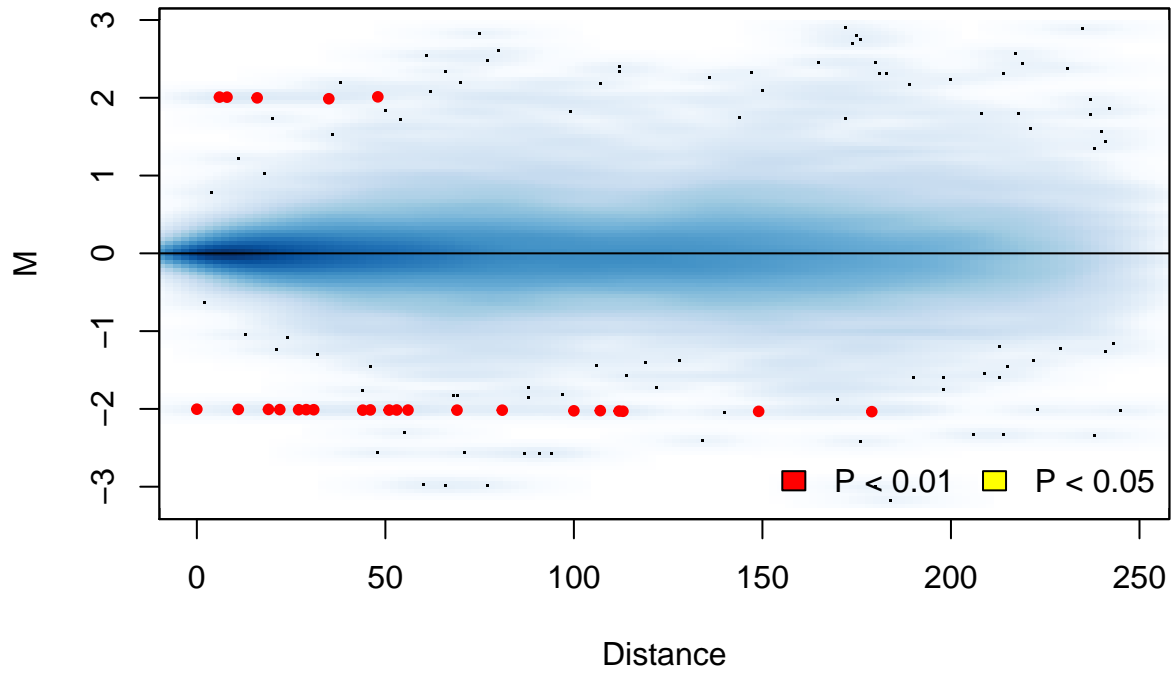


```
## True positive rate:
calc_stats1(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.88	0.9982936	0.5	0.999767	0.0017064	0.12	0.5	0.9980647

```
# FDR on full vector
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

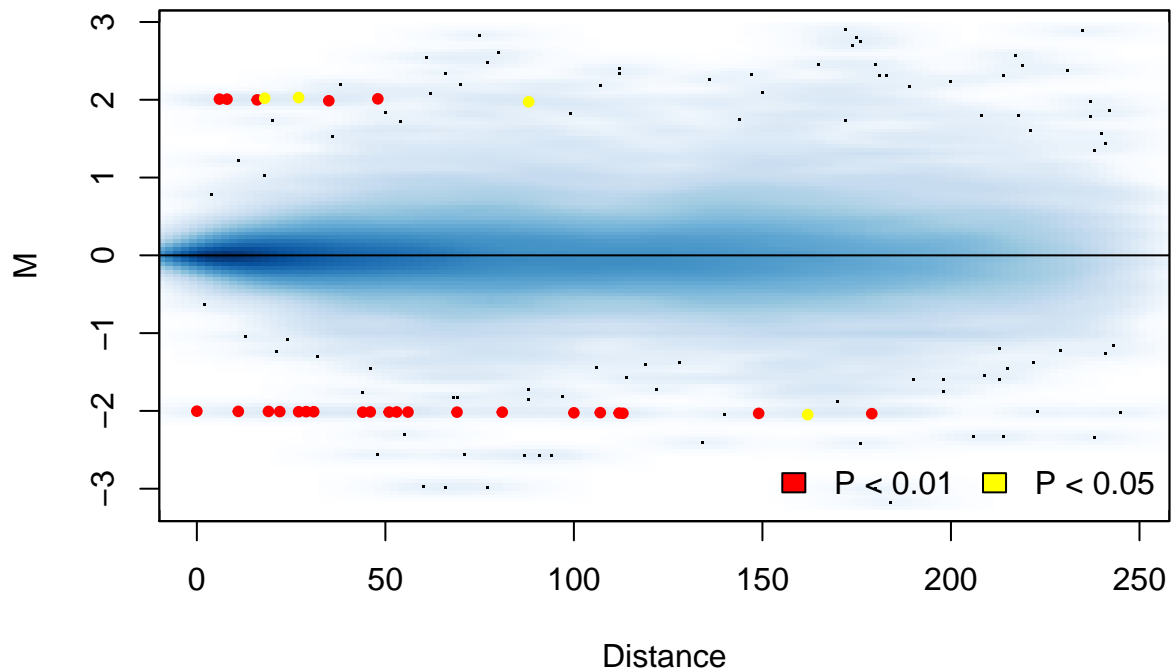


```
## True positive rate:
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.5	1	1	0.9990314	0	0.5	0	0.9990324

```
# FDR by distance
hic.table <- fdr_adjust(hic.table)
```

## MD Plot



```
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
0.58	1	1	0.9991863	0	0.42	0	0.9991872

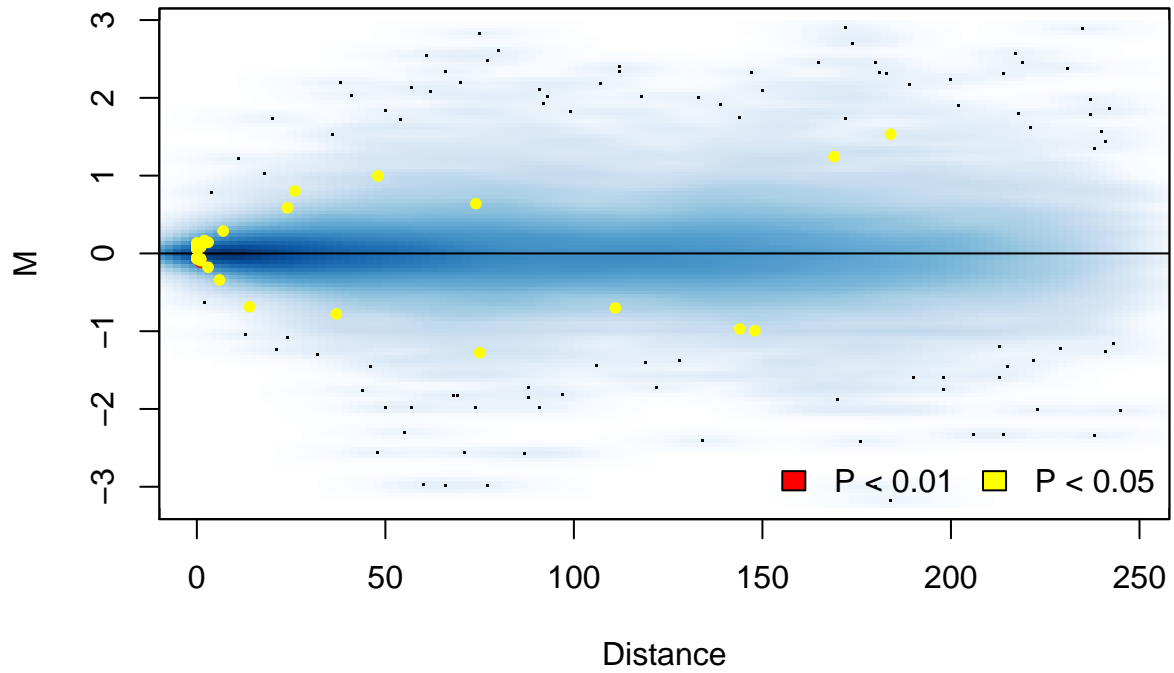
## No changes

```
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

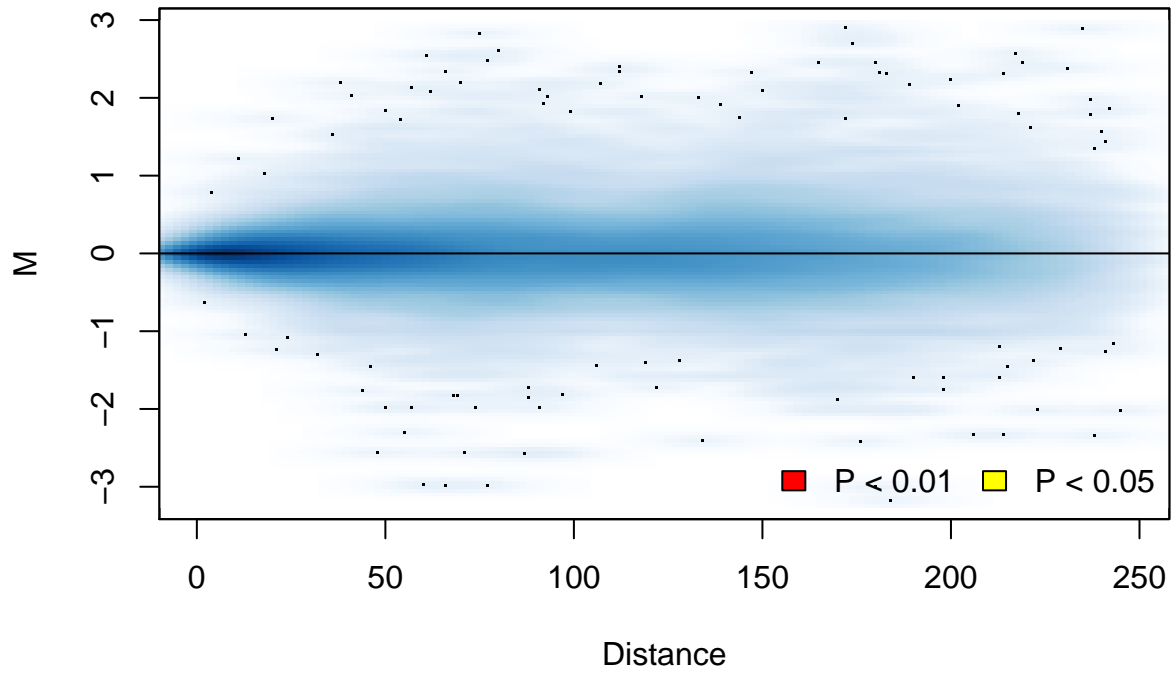


```
## False positives
calc_stats1(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
# FDR on full vector
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

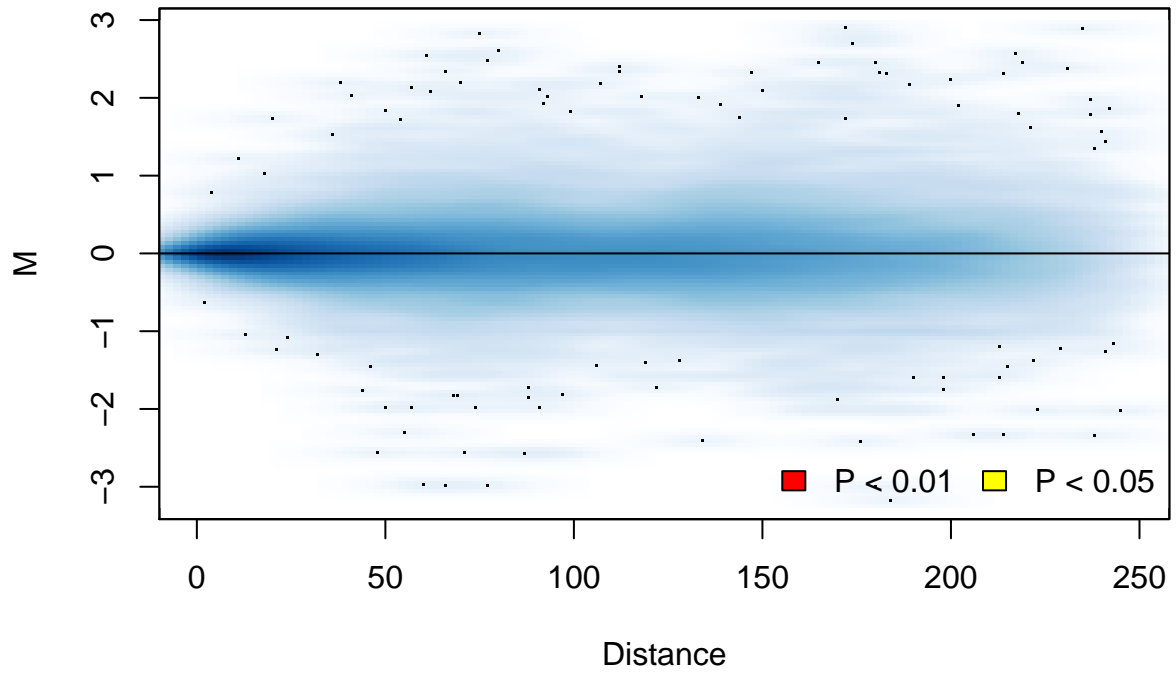


```
## False positives
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
# FDR by distance
hic.table <- fdr_adjust(hic.table)
```

## MD Plot



```
calc_stats(hic.table)
```

TPR	TNR	PPV	NPV	FPR	FNR	FDR	ACC
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN