ROC_realdata

John Stansfield

December 7, 2017

ROC curves on ranks

Set up

```
# SET UP PARAMETERS
N = 300
FC = 2
```

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

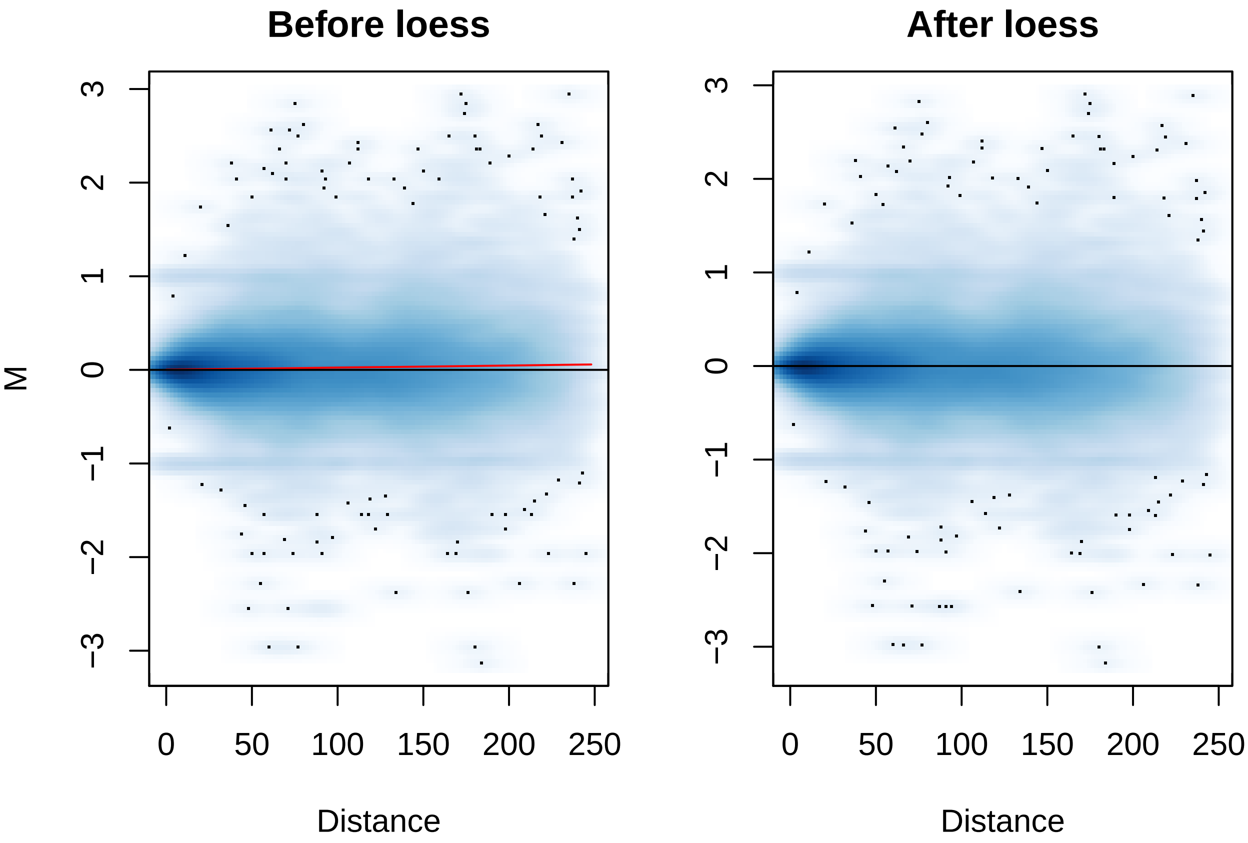
introduce changes

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

# 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))
hic.table[changes,]$IF1 = newIF1
```

```
## 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.
```

```
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 = TRUE)
```

```
## Span for loess: 0.899928222076755
## GCV for loess: 6.32010772053363e-06

## AIC for loess: -0.811906225161962
```

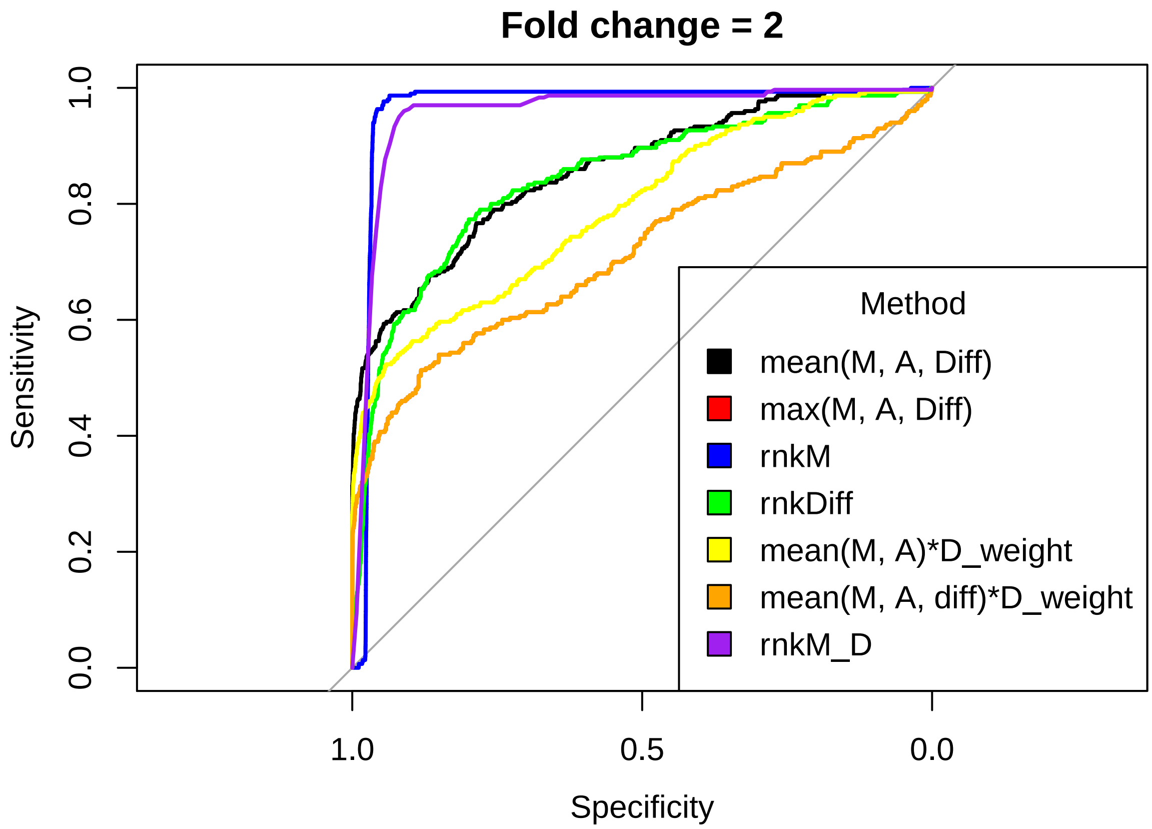


```
hic.table <- hic_diff(hic.table, Plot = FALSE)

backup.table <- hic.table
```

ROC curves

```
truth <- hic.table$truth

roc_mean_A_M_diff <- roc(response = truth, predictor = hic.table$rnkMean)

# for max(M, A, diff)
roc_max_A_M_diff <- roc(response = truth, predictor = hic.table$rnkMax)

# for rnkM only
roc_rnkM <- roc(response = truth, predictor = hic.table$rnkM)

# for rnkDiff only
roc_rnkDiff <- roc(response = truth, predictor = hic.table$rnkDiff)

# for mean(M, A) with D weighting
mean_rank <- hic.table %>% dplyr::select(rnkM, rnkA) %>% as.matrix() %>% apply(., 1, mean)
```

```r
hic.table[, rnkMean := mean_rank]
# create weight for distance
dist_weight <- 1+((hic.table$D + 1)/max(hic.table$D + 1))
hic.table[, rnkMean := dist_weight * rnkMean]
roc_D_weight <- roc(response = truth, predictor = hic.table$rnkMean)

# for rnkM_D only
roc_rnkM_D <- roc(response = truth, predictor = hic.table$rnkM_D)

# for mean(M, A, Diff) with D weighting
mean_rank <- hic.table %>% dplyr::select(rnkM, rnkA, rnkDiff) %>% as.matrix() %>% apply(., 1, mean)
hic.table[, rnkMean := mean_rank]
# create weight for distance
dist_weight <- 1+((hic.table$D + 1)/max(hic.table$D + 1))
hic.table[, rnkMean := dist_weight * rnkMean]
hic.table <- hic.table[order(rnkMean),]
roc_D_weight2 <- roc(response = truth, predictor = hic.table$rnkMean)

# plots
plot.colors <- c('black', 'red', 'blue', 'green', 'yellow', 'orange', 'purple')
plot(roc_mean_A_M_diff, main = paste0('Fold change = ', FC))
plot(roc_max_A_M_diff, col = plot.colors[2], add = TRUE)
plot(roc_rnkM, col = plot.colors[3], add = TRUE)
plot(roc_rnkDiff, col = plot.colors[4], add = TRUE)
plot(roc_D_weight, col = plot.colors[5], add = TRUE) # mean (M, A) w/ Dist weight
plot(roc_D_weight2, col = plot.colors[6], add = TRUE) # mean(M, A, Diff) w/ dist weight
plot(roc_rnkM_D, col = plot.colors[7], add = TRUE)
legend('bottomright', inset = 0, legend = c('mean(M, A, Diff)', 'max(M, A, Diff)', 'rnkM', 'rnkDiff', 'r
                                            'rnkM_D'), title = 'Method', fill = plot.colors, horiz = F)
```



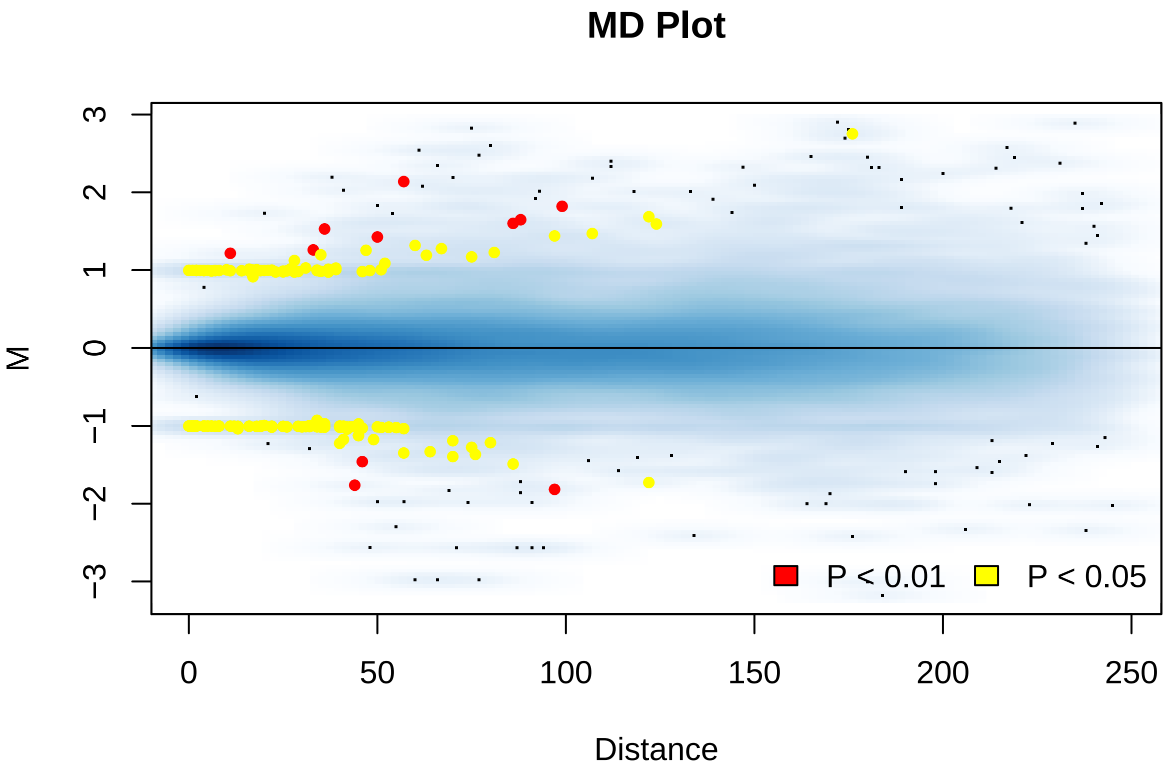
MD plots for raw difference cutoff z-score

```
# weighting by distance
.calc_zscores <- function(hic.table, rawdiff_cutoff, Plot = TRUE) {
  hic.table[, diff := adj.IF2 - adj.IF1]
  # calculate z scores
  Zm1 <- (hic.table$adj.M - mean(hic.table$adj.M)) / sd(hic.table$adj.M)
  # cut off for raw difference on Z score of M
  # Zm_idx <- ifelse(abs(hic.table$diff) < rawdiff_cutoff, TRUE, FALSE)
  # Zm[Zm_idx] <- 0 # set z scores where raw diff is below cut off to 0
  Zm1[hic.table$diff < rawdiff_cutoff & hic.table$diff > -rawdiff_cutoff] <- 0

  hic.table[, Zm := Zm1]
  # calculate distance weighting
  dist_weight <- 1 - ((hic.table$D + 1)/max(hic.table$D + 1))
  hic.table[, D_wt := dist_weight]
  hic.table[, Zwt := Zm * D_wt]
  hic.table[, p.val := 2*pnorm(abs(Zwt), lower.tail = FALSE)]
  hic.table[, p.adj := p.adjust(p.val, method = 'fdr')]
  # MD.plot2(hic.table$adj.M, hic.table$D, hic.table$p.adj)
  if (Plot) MD.plot2(hic.table$adj.M, hic.table$D, hic.table$p.val)
  return(hic.table)
}
```

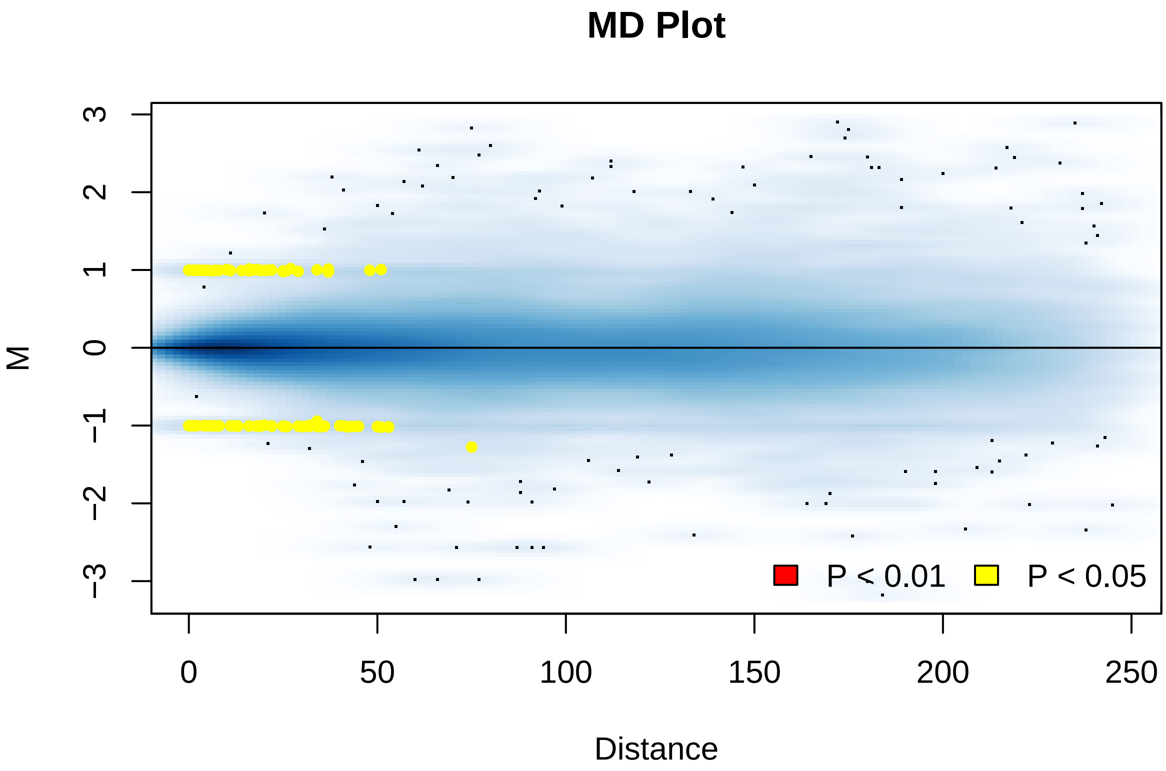
```
# no weighting by distance
.calc_zscores2 <- function(hic.table, rawdiff_cutoff, Plot = TRUE) {
  hic.table[, diff := adj.IF2 - adj.IF1]
  # calculate z scores
  Zm1 <- (hic.table$adj.M - mean(hic.table$adj.M)) / sd(hic.table$adj.M)
  # cut off for raw difference on Z score of M
  # Zm_idx <- ifelse(abs(hic.table$diff) < rawdiff_cutoff, TRUE, FALSE)
  # Zm[Zm_idx] <- 0 # set z scores where raw diff is below cut off to 0
  Zm1[hic.table$diff < rawdiff_cutoff & hic.table$diff > -rawdiff_cutoff] <- 0

  hic.table[, Zm := Zm1]
  # calculate distance weighting
  dist_weight <- 1 - ((hic.table$D + 1)/max(hic.table$D + 1))
  hic.table[, D_wt := dist_weight]
  hic.table[, Zwt := Zm * D_wt]
  hic.table[, p.val := 2*pnorm(abs(Zm), lower.tail = FALSE)]
  hic.table[, p.adj := p.adjust(p.val, method = 'fdr')]
  # MD.plot2(hic.table$adj.M, hic.table$D, hic.table$p.adj)
  if (Plot) MD.plot2(hic.table$adj.M, hic.table$D, hic.table$p.val)
  return(hic.table)
}
```

distance weighting

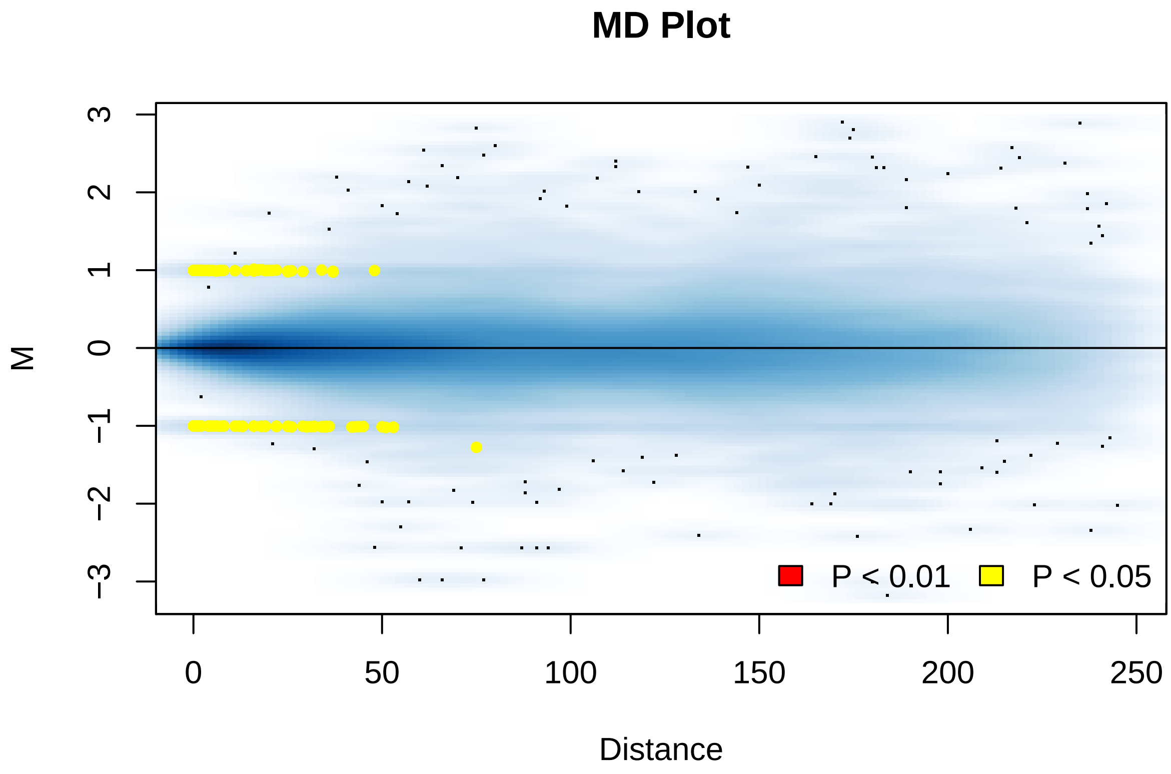
```
hic.table <- backup.table
hic.table <- .calc_zscores(hic.table, 10)
```



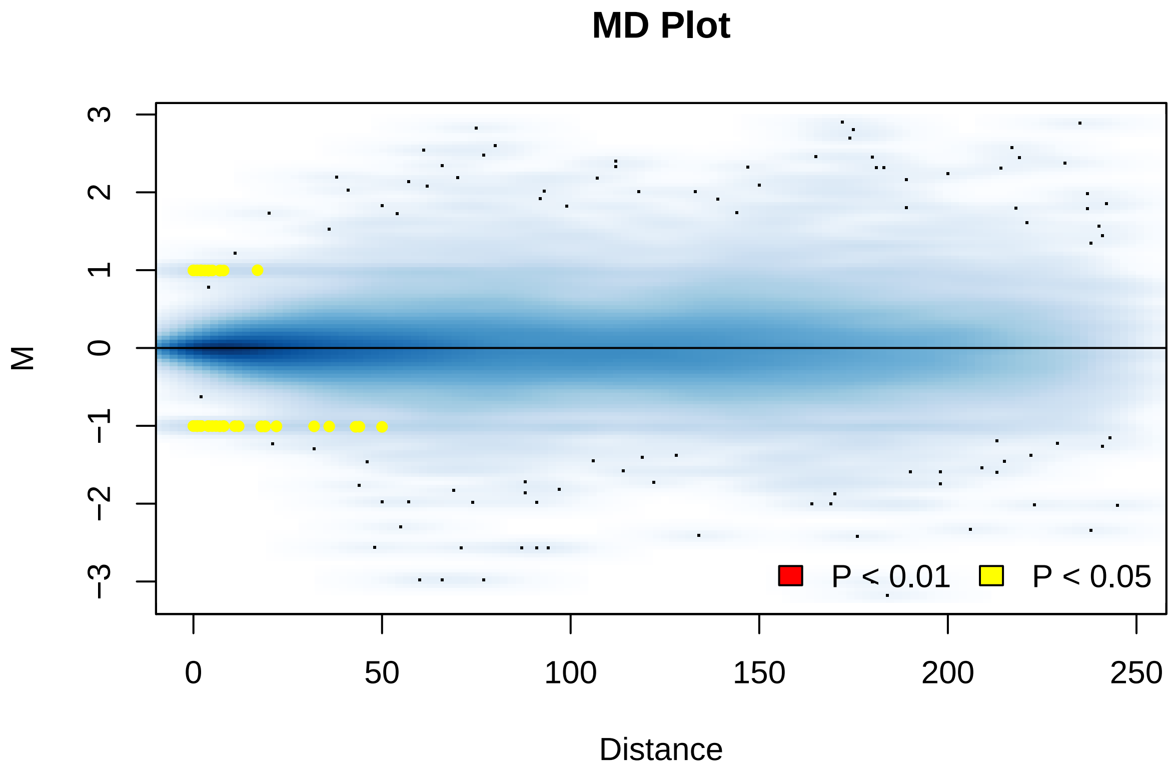
```
hic.table <- .calc_zscores(hic.table, 20)
```



```r
hic.table <- .calc_zscores(hic.table, 30)
```

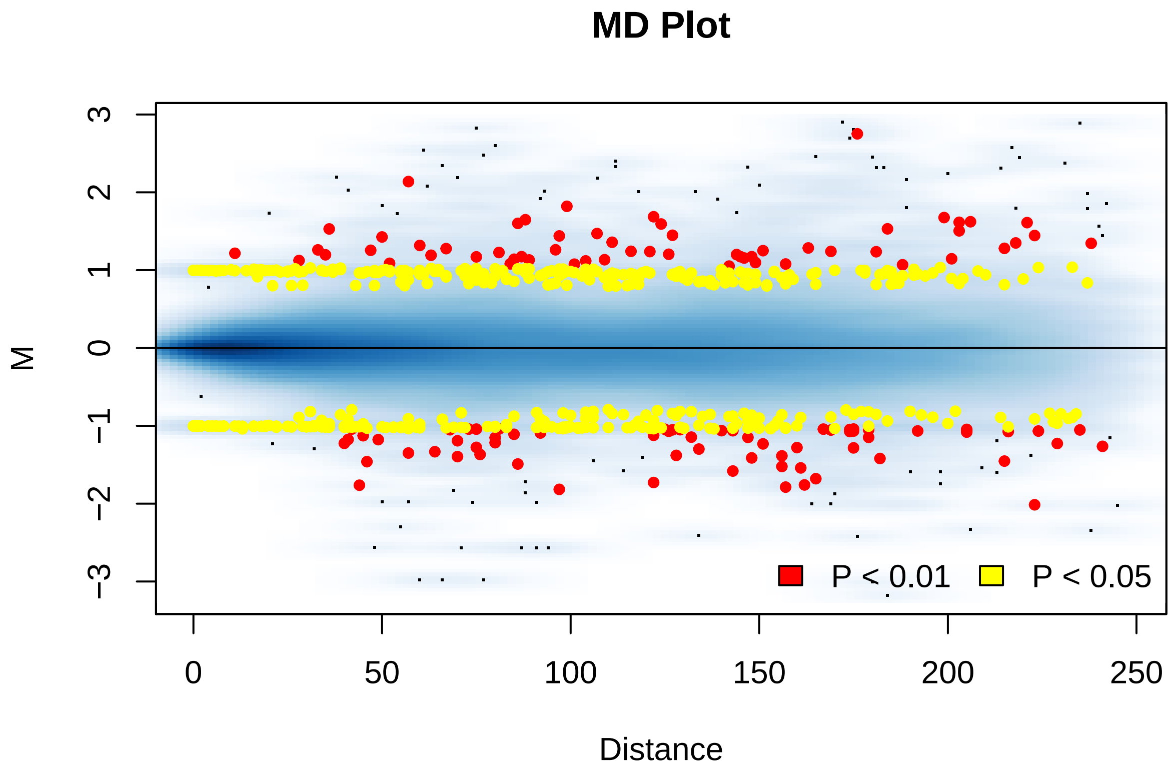


```
hic.table <- .calc_zscores(hic.table, 100)
```

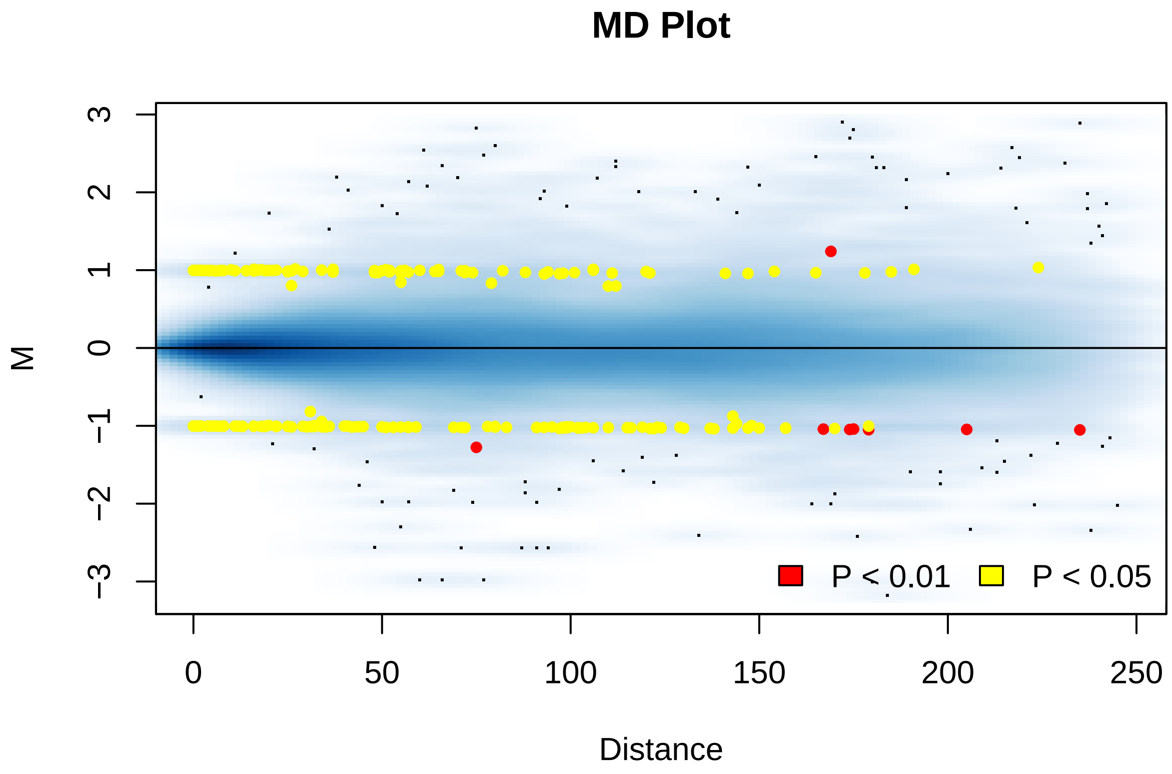


no distance weighting

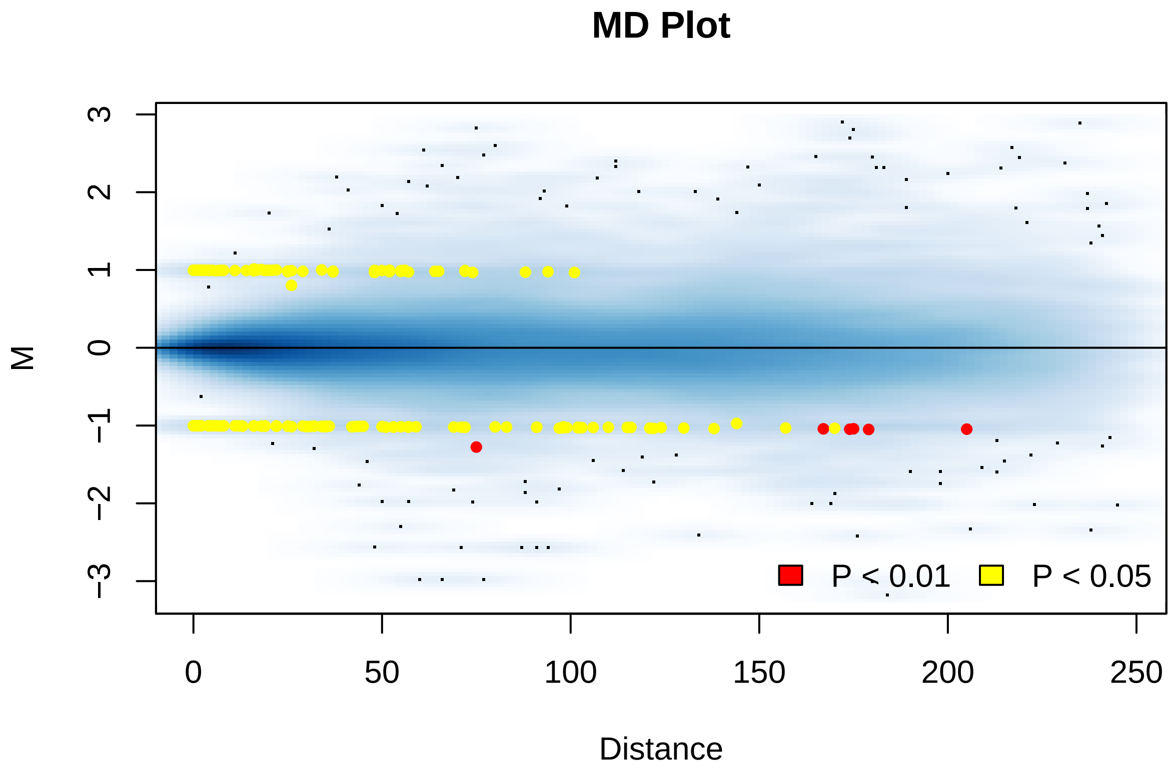
```
hic.table <- backup.table
hic.table <- .calc_zscores2(hic.table, 10)
```



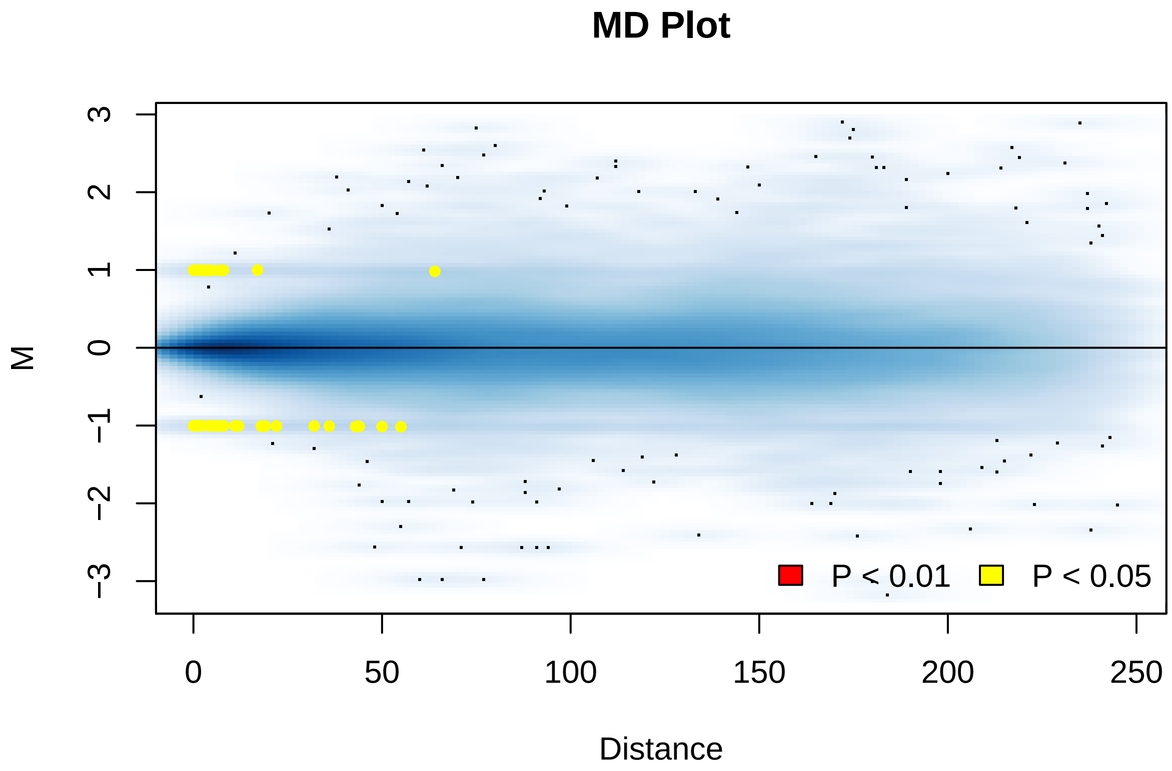
```
hic.table <- .calc_zscores2(hic.table, 20)
```



```
hic.table <- .calc_zscores2(hic.table, 30)
```



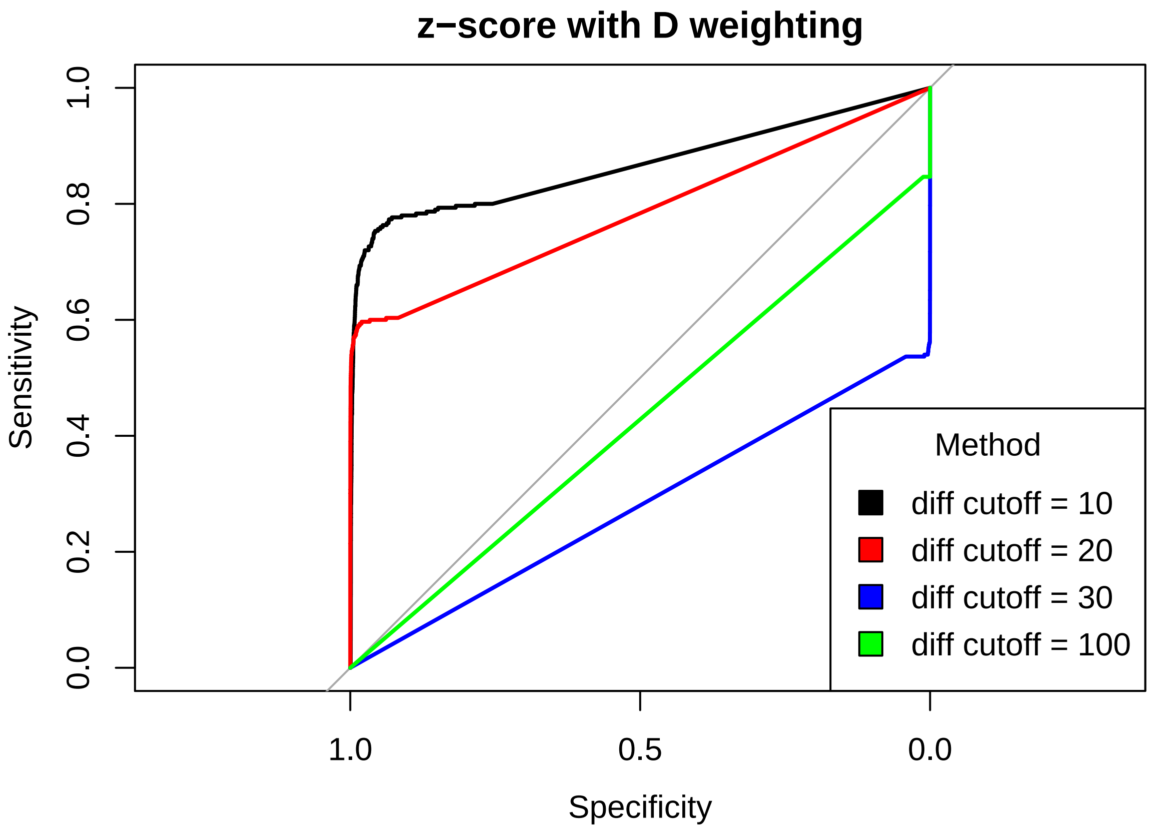
```
hic.table <- .calc_zscores2(hic.table, 100)
```



ROC for Z scores

distance weighting

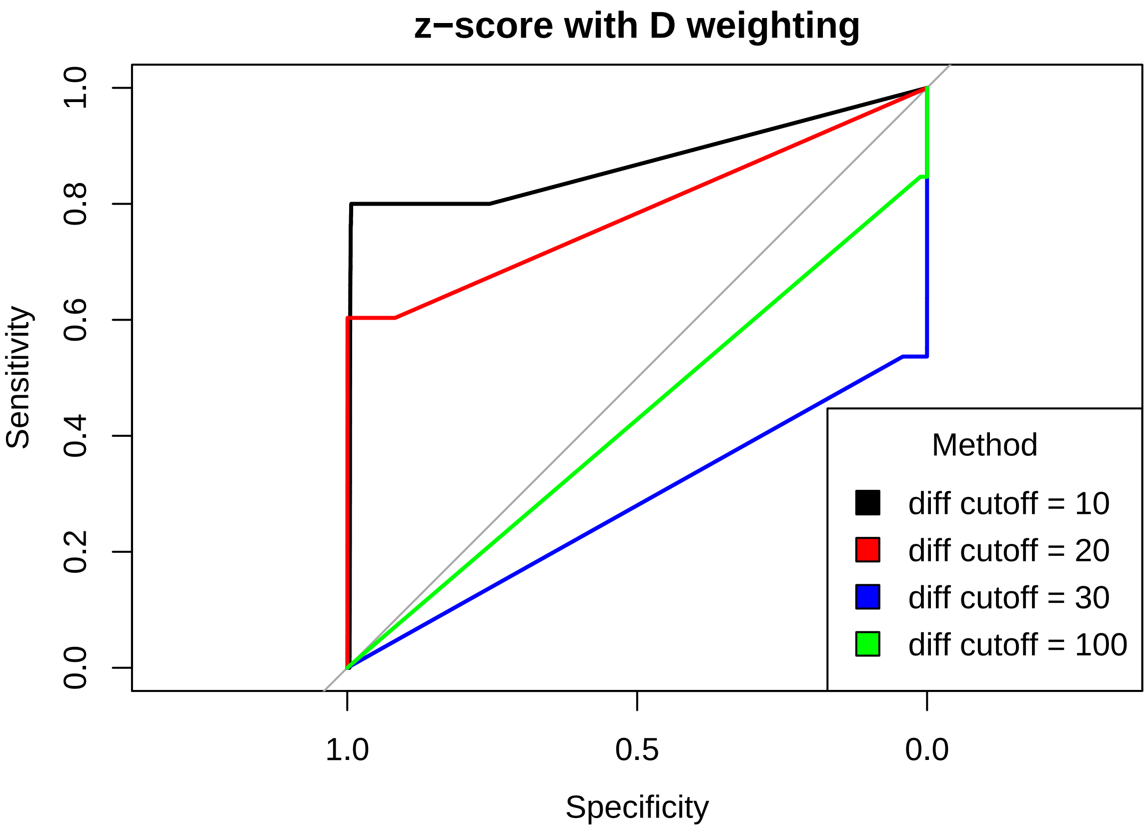
```
plot(roc_w10, main = 'z-score with D weighting')
plot(roc_w20, col = plot.colors[2], add = TRUE)
plot(roc_w30, col = plot.colors[3], add = TRUE)
plot(roc_w100, col = plot.colors[4], add = TRUE)
legend('bottomright', inset = 0, legend = c('diff cutoff = 10', 'diff cutoff = 20', 'diff cutoff = 30',
       title = 'Method', fill = plot.colors[1:4], horiz = F)
```



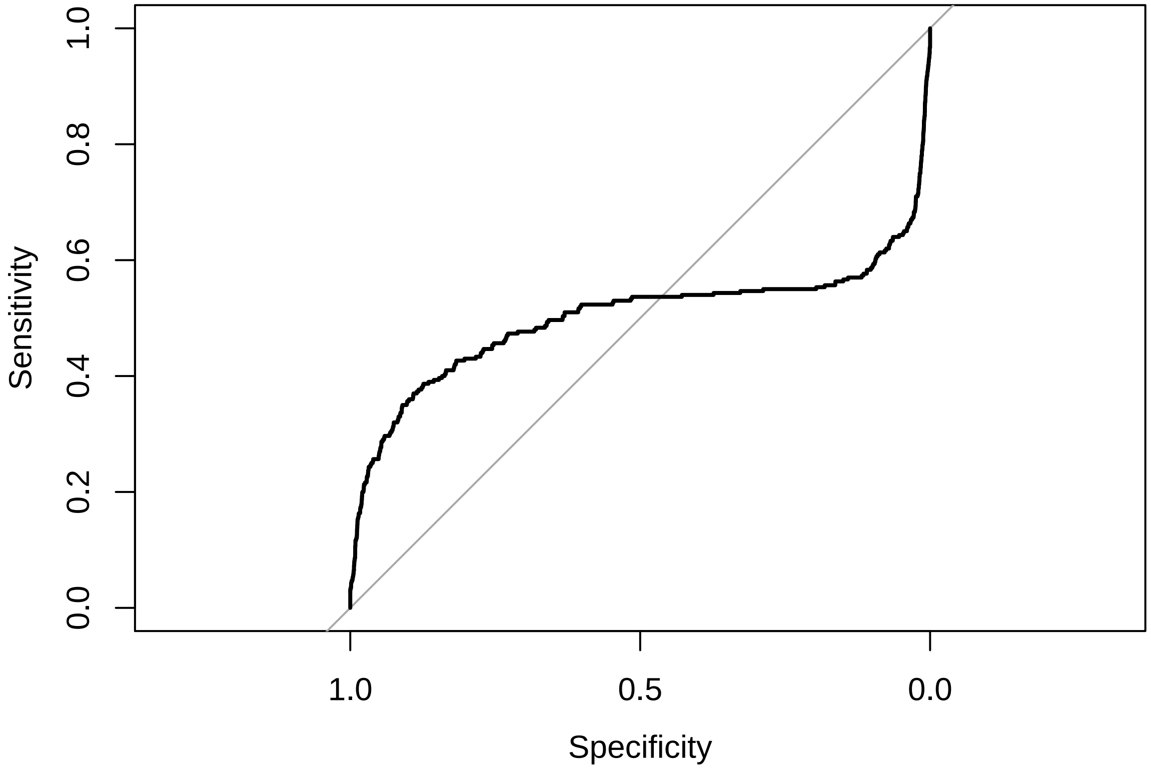
no distance weighting

```
hic.table <- backup.table
hic.table <- .calc_zscores2(hic.table, 10, FALSE)
roc_w10 <- roc(response = hic.table$truth, predictor = hic.table$p.val)
hic.table <- .calc_zscores2(hic.table, 20, FALSE)
roc_w20 <- roc(response = hic.table$truth, predictor = hic.table$p.val)
hic.table <- .calc_zscores2(hic.table, 30, FALSE)
roc_w30 <- roc(response = hic.table$truth, predictor = hic.table$p.val)
hic.table <- .calc_zscores2(hic.table, 100, FALSE)
roc_w100 <- roc(response = hic.table$truth, predictor = hic.table$p.val)
```

```
plot(roc_w10, main = 'z-score with D weighting')
plot(roc_w20, col = plot.colors[2], add = TRUE)
plot(roc_w30, col = plot.colors[3], add = TRUE)
plot(roc_w100, col = plot.colors[4], add = TRUE)
legend('bottomright', inset = 0, legend = c('diff cutoff = 10', 'diff cutoff = 20', 'diff cutoff = 30',
       title = 'Method', fill = plot.colors[1:4], horiz = F)
```

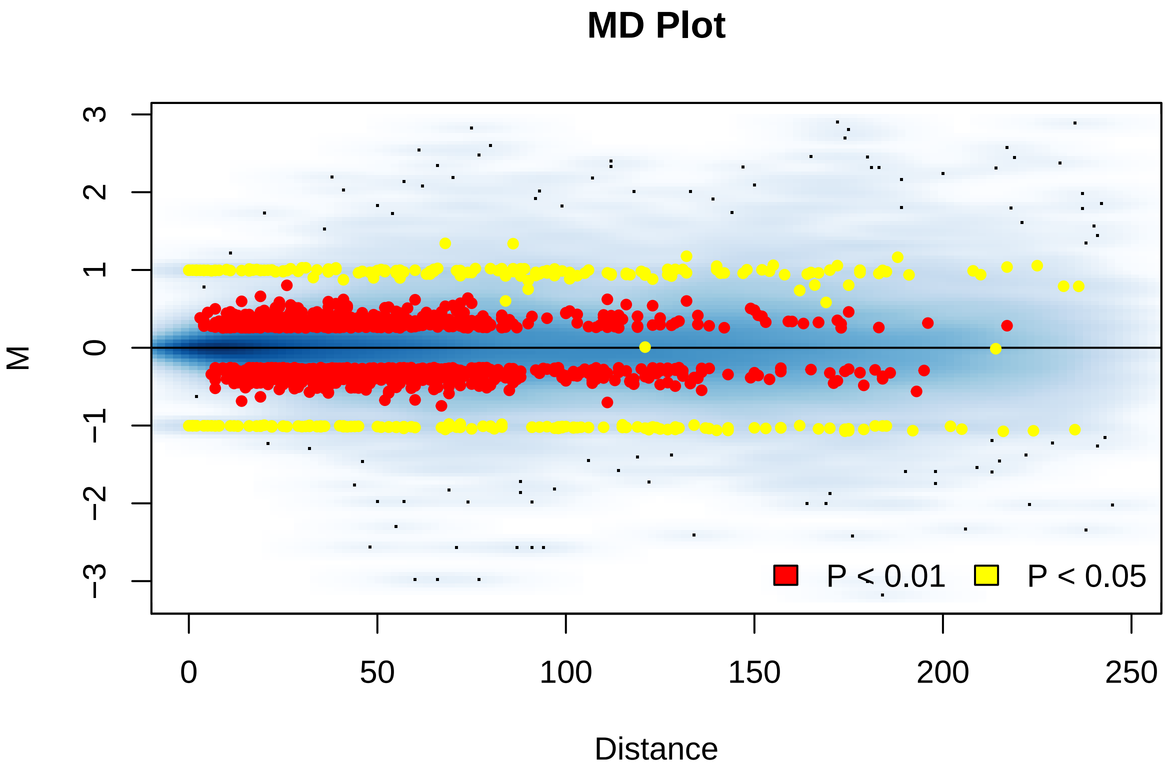


```
hic.table <- backup.table
hic.table[, diff := adj.IF2 - adj.IF1]

roc_diff <- roc(response = hic.table$truth, predictor = hic.table$diff)
plot(roc_diff)
```



MD plots

```
hic.table <- backup.table
alpha <- 0.05
idx <- 1:(nrow(hic.table) * alpha)


hic.table <- hic.table[order(rnkMax),]
topRanks <- rep(1, nrow(hic.table)) # make indicator for top ranks
topRanks[idx] <- 0 # set top ranking rows to 0 indicator for plotting on MD plot
topRanks[hic.table$truth == 1] <- 0.04
MD.plot2(hic.table$adj.M, hic.table$D, p.val = topRanks)
```


MD Plot
M
3
2
1
0
−1
−2
−3
0
50
100
150
200
250
Distance
🟥 P < 0.01
🟨 P < 0.05