differences by distance

John Stansfield

January 3, 2018

Set up

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

read in GM12878 replicates data

```
chr1.primary <- read.table("D:/3D_DNA/GM12878_replicates/GM12878_primary_1000000/primary.chr1.1000000.t
chr1.replicate <- read.table("D:/3D_DNA/GM12878_replicates/GM12878_replicate_1000000/replicate.chr1.100

chr1.table <- create.hic.table(chr1.primary, chr1.replicate, chr = 'chr1', scale = TRUE)
backup.table <- create.hic.table(chr1.primary, chr1.replicate, chr = 'chr1', scale = TRUE)
```

See what is found different before any changes

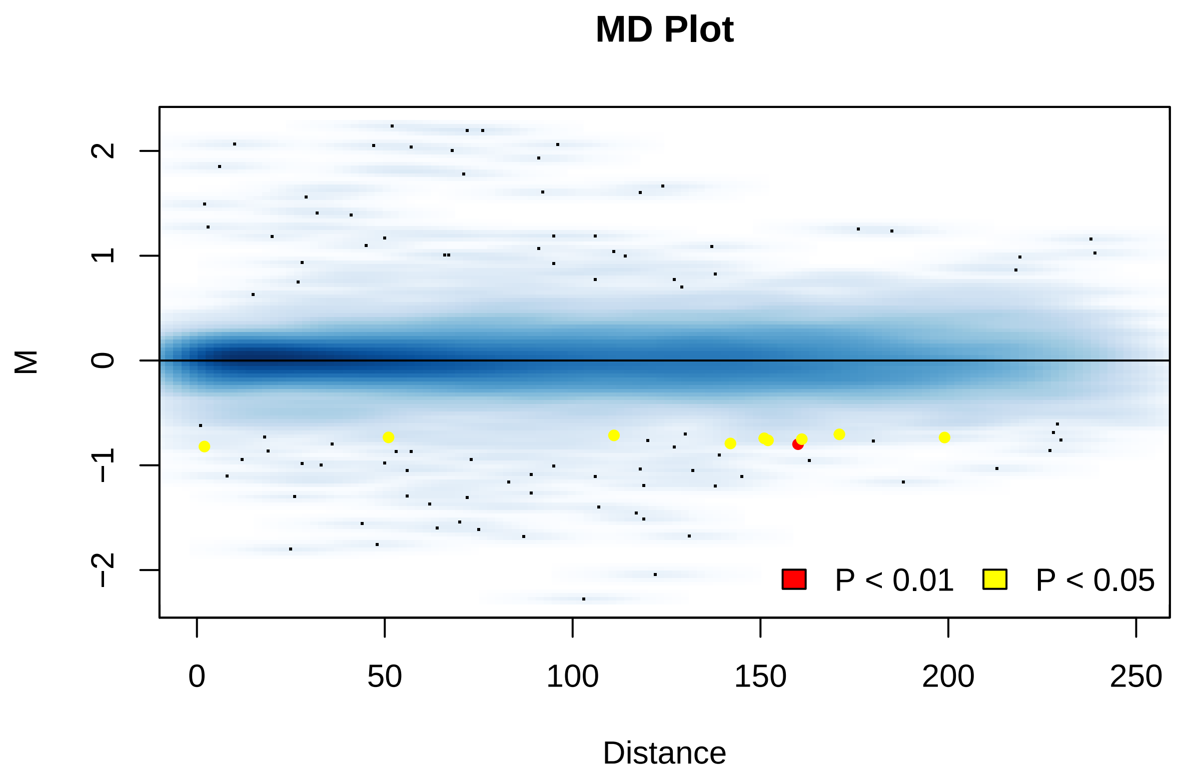
```
chr1.table <- hic_loess(chr1.table)
```

```
## Span for loess: 0.0905762145919513
```

```
## GCV for loess: 1.56323544402087e-06
```

```
## AIC for loess: -2.1795356885631
```

```
chr1.table <- hic_compare(chr1.table, adjust_dist = TRUE, p.method = 'holm', A.quantile = 0.1, Plot = T
```



```
sum(chr1.table$p.adj < 0.05)
```

```
## [1] 10
```

function to spike differences and generate ROC

```
make_ROC <- function(raw1, raw2, FC = 2, numChanges= 10, dist = 1, A.quantile = 0.1, adjust_dist = TRUE
  hic.table <- create.hic.table(raw1, raw2, chr='chr1', scale = TRUE)
  # spike in differences
  # get which interactions at distance
  sample_space <- which(hic.table$D == dist)
  changes <- sample(sample_space, numChanges)
  # set IFs to mean IF then multiply one by FC

  meanIF <- ((hic.table[changes,]$IF1 + hic.table[changes,]$IF2) / 2) %>% round() %>% as.integer()
  hic.table[changes, IF1 := meanIF ]
  hic.table[changes, IF2 := meanIF]
  newIF <- hic.table[changes,]$IF1 * FC %>% as.integer()
  hic.table[changes, IF1 :=  newIF]
  hic.table = hic.table[, M := log2(IF2/IF1)]
  truth <- rep(0, nrow(hic.table))
  truth[changes] <- 1
  hic.table[, truth := truth]

  # run HiCcompare
  hic.table <- hic_loess(hic.table, Plot = TRUE)
  hic.table <- hic_compare(hic.table, adjust_dist = adjust_dist, A.quantile = A.quantile, p.method = p.m


  # get number true positive
  # print('number TRUE POSITIVE')
  # print(sum(hic.table[changes,]$p.adj < 0.05))

  # make ROC
  result <- list(roc(response = hic.table$truth, predictor = hic.table$p.adj), hic.table)
  return(result)
}
```

ROC

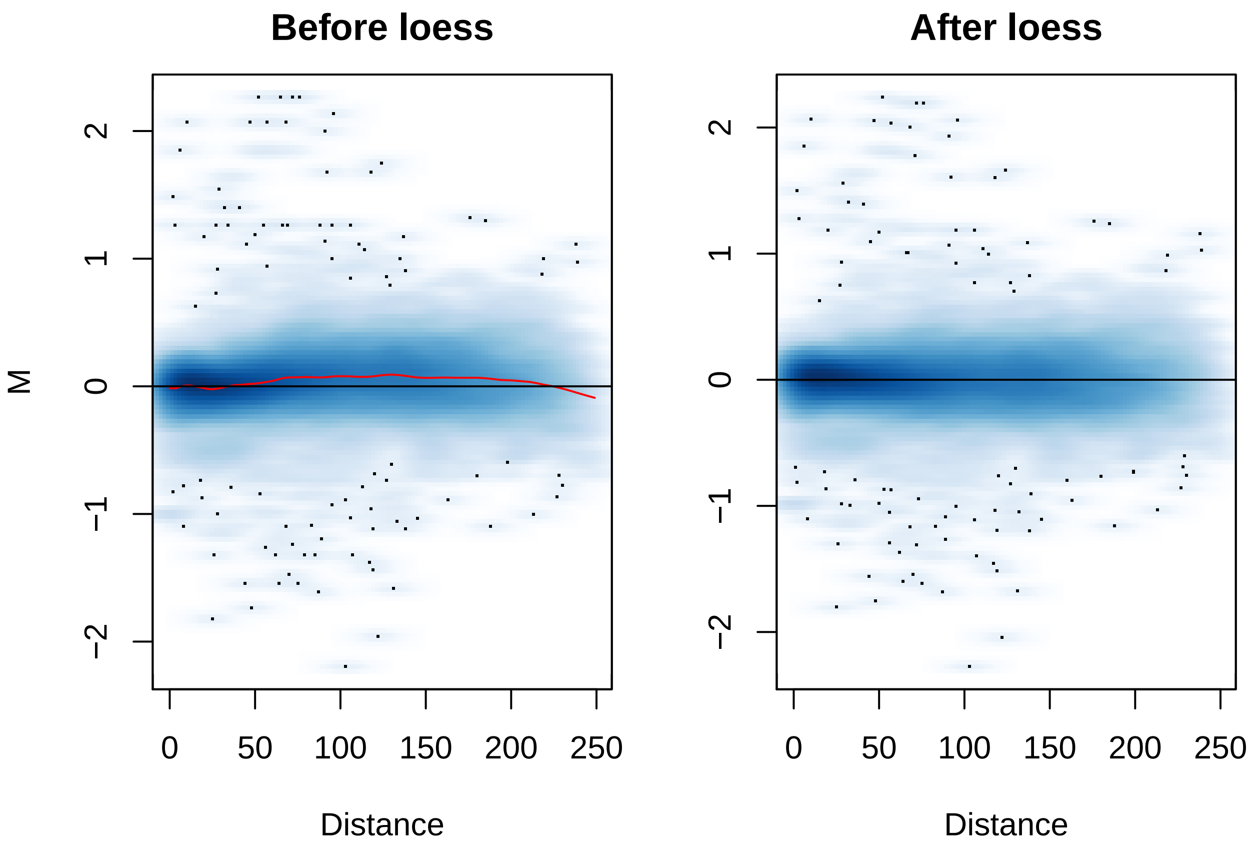
```
# chr1.table <- backup.table
fc2d1 <- make_ROC(chr1.primary, chr1.replicate, FC = 2, numChanges = 10, dist = 1, A.quantile = 0.1, ad
```

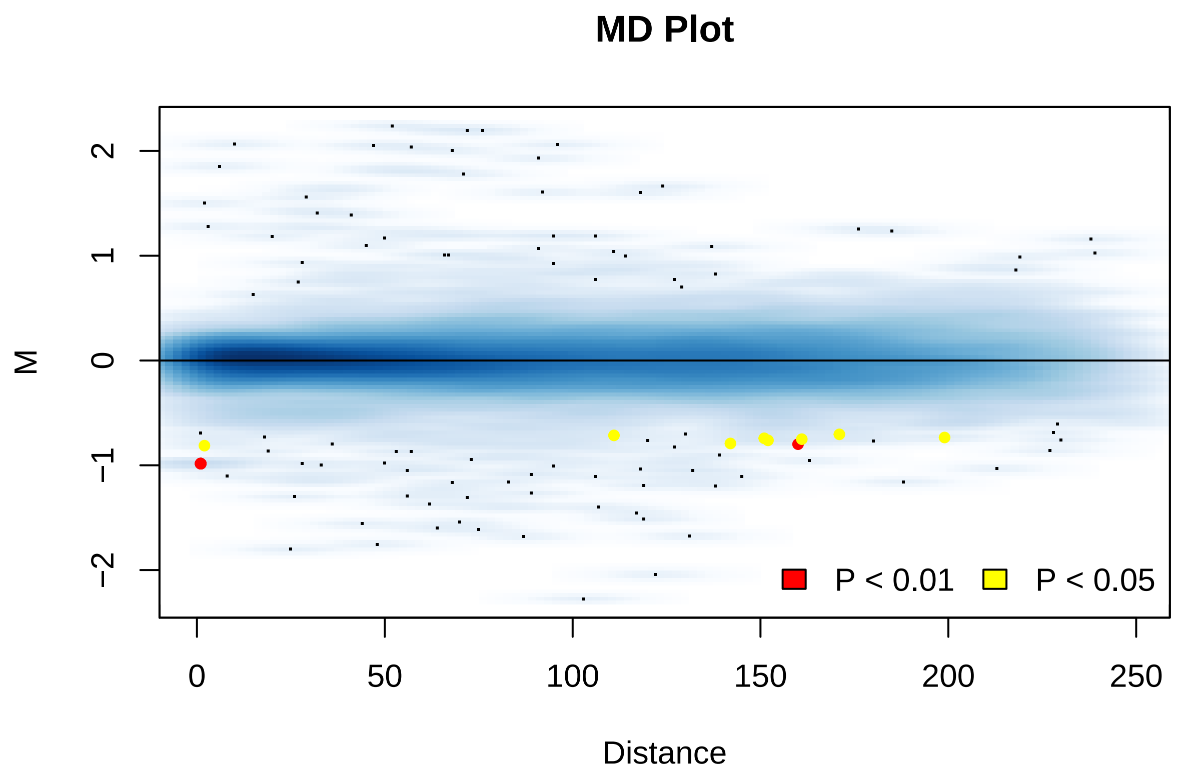
```
## Warning in `[.data.table`(hic.table, changes, `:=`(IF2, meanIF)): Coerced
## 'integer' RHS to 'double' to match the column's type. Either change the
## target column to 'integer' first (by creating a new 'integer' vector length
## 26562 (nrows of entire table) and assign that; i.e. 'replace' column), or
## coerce RHS to 'double' (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: 0.0905649195041254
```

```
## GCV for loess: 1.57678941939928e-06
```

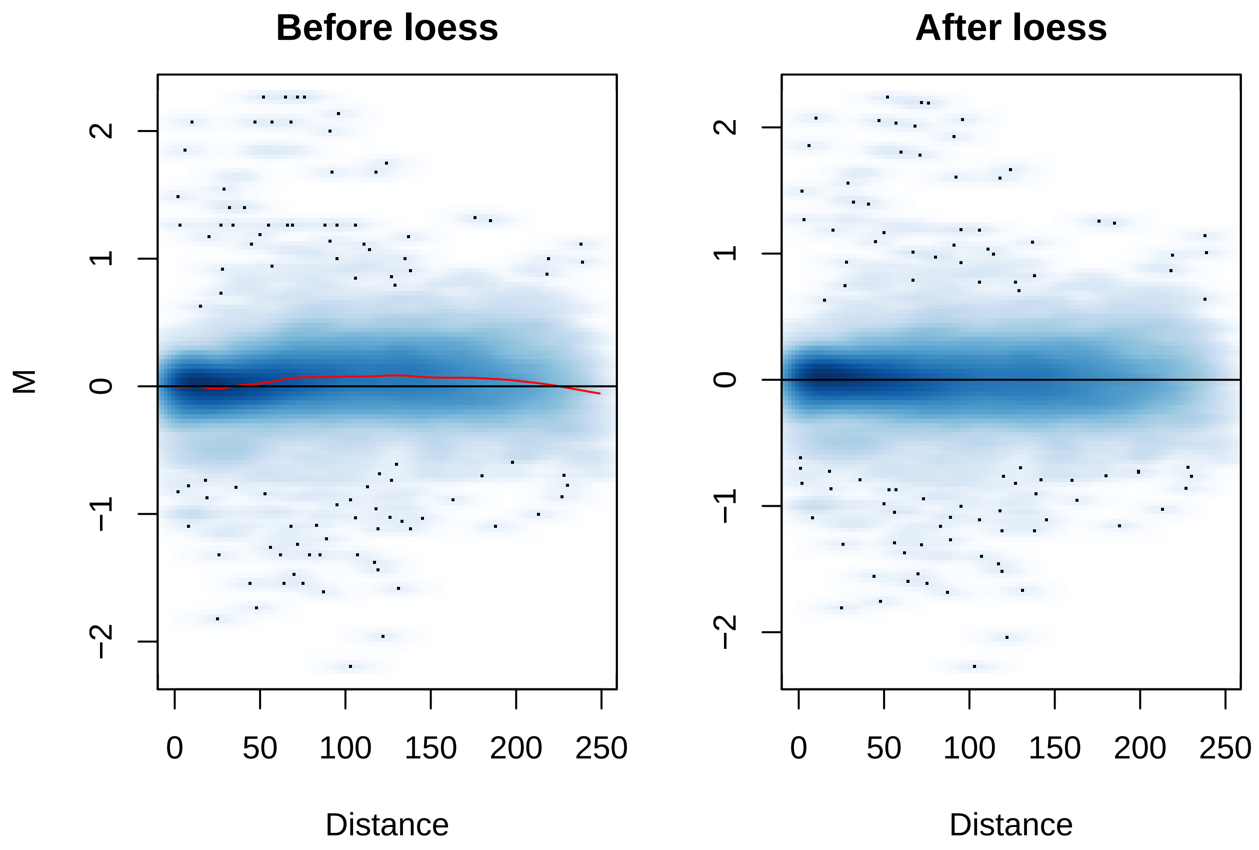
```
## AIC for loess: -2.17090236945863
```

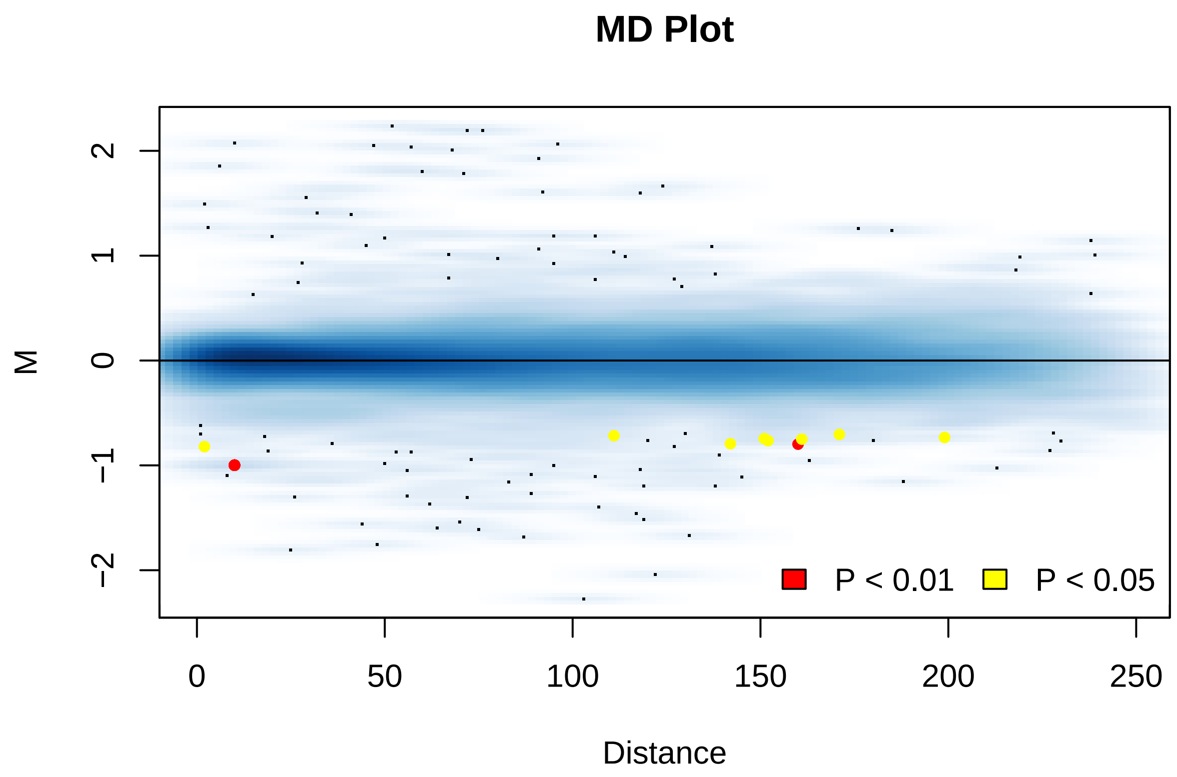

Before loess
After loess
M
Distance
Distance
2
1
0
−1
−2
0
50
100
150
200
250



```
fc2d10 <- make_ROC(chr1.primary, chr1.replicate, FC = 2, numChanges = 10, dist = 10, A.quantile = 0.1, a
```

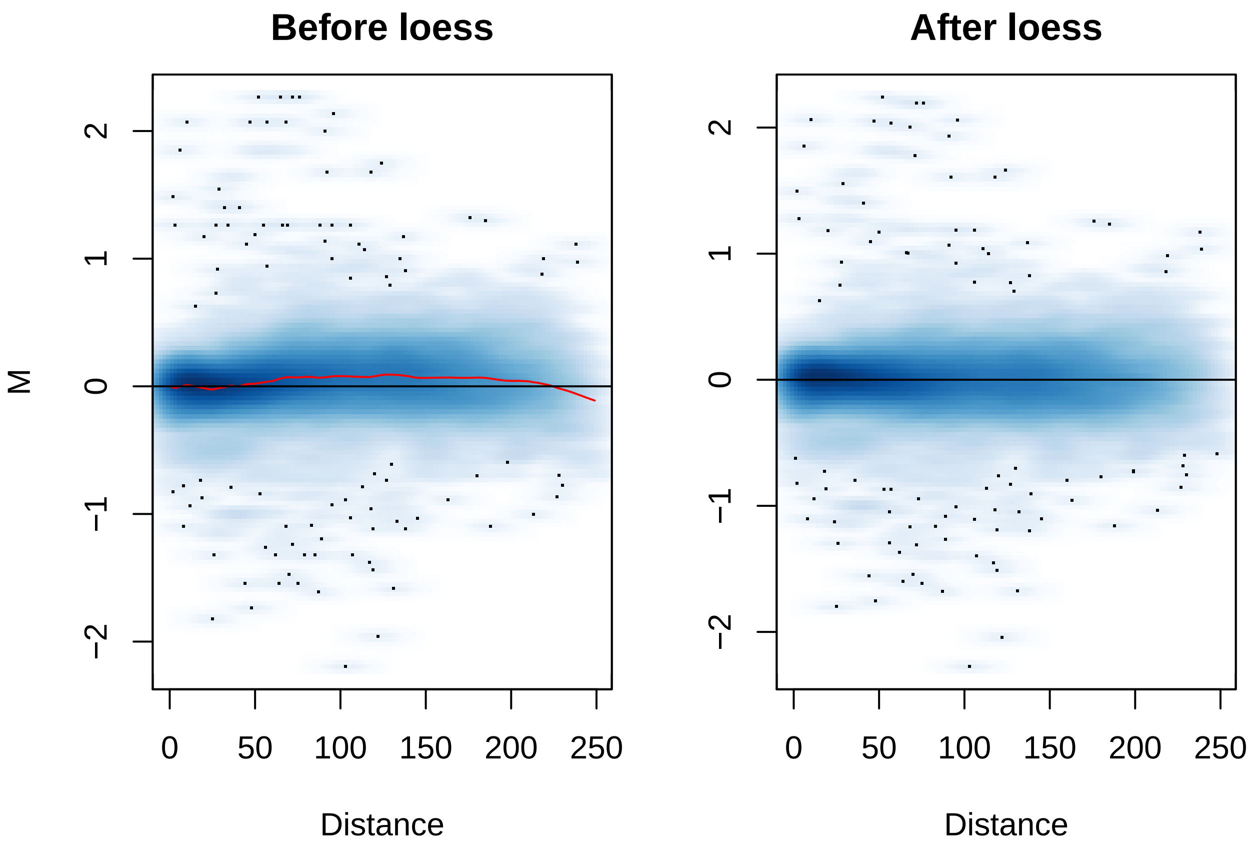
```
## Warning in `[.data.table`(hic.table, changes, `:=`(IF2, meanIF)): Coerced
## 'integer' RHS to 'double' to match the column's type. Either change the
## target column to 'integer' first (by creating a new 'integer' vector length
## 26562 (nrows of entire table) and assign that; i.e. 'replace' column), or
## coerce RHS to 'double' (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: 0.158792829569718

## GCV for loess: 1.5769911680256e-06

## AIC for loess: -2.17156129692418
```


Before loess
After loess
M
Distance
Distance
2
1
0
−1
−2
0
50
100
150
200
250

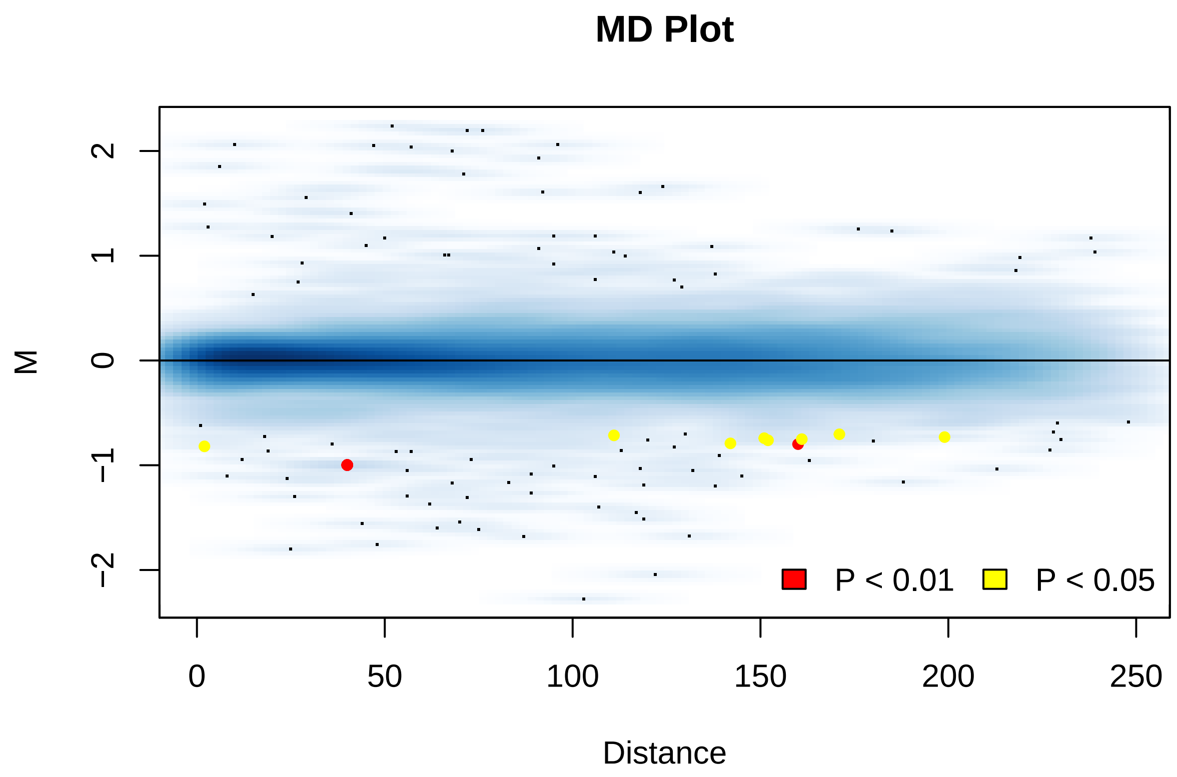


```
fc2d40 <- make_ROC(chr1.primary, chr1.replicate, FC = 2, numChanges = 10, dist = 40, A.quantile = 0.1, a
```

```
## Warning in `[.data.table`(hic.table, changes, `:=`(IF2, meanIF)): Coerced
## 'integer' RHS to 'double' to match the column's type. Either change the
## target column to 'integer' first (by creating a new 'integer' vector length
## 26562 (nrows of entire table) and assign that; i.e. 'replace' column), or
## coerce RHS to 'double' (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: 0.0704240656691385

## GCV for loess: 1.5773290172118e-06

## AIC for loess: -2.17003594666054
```

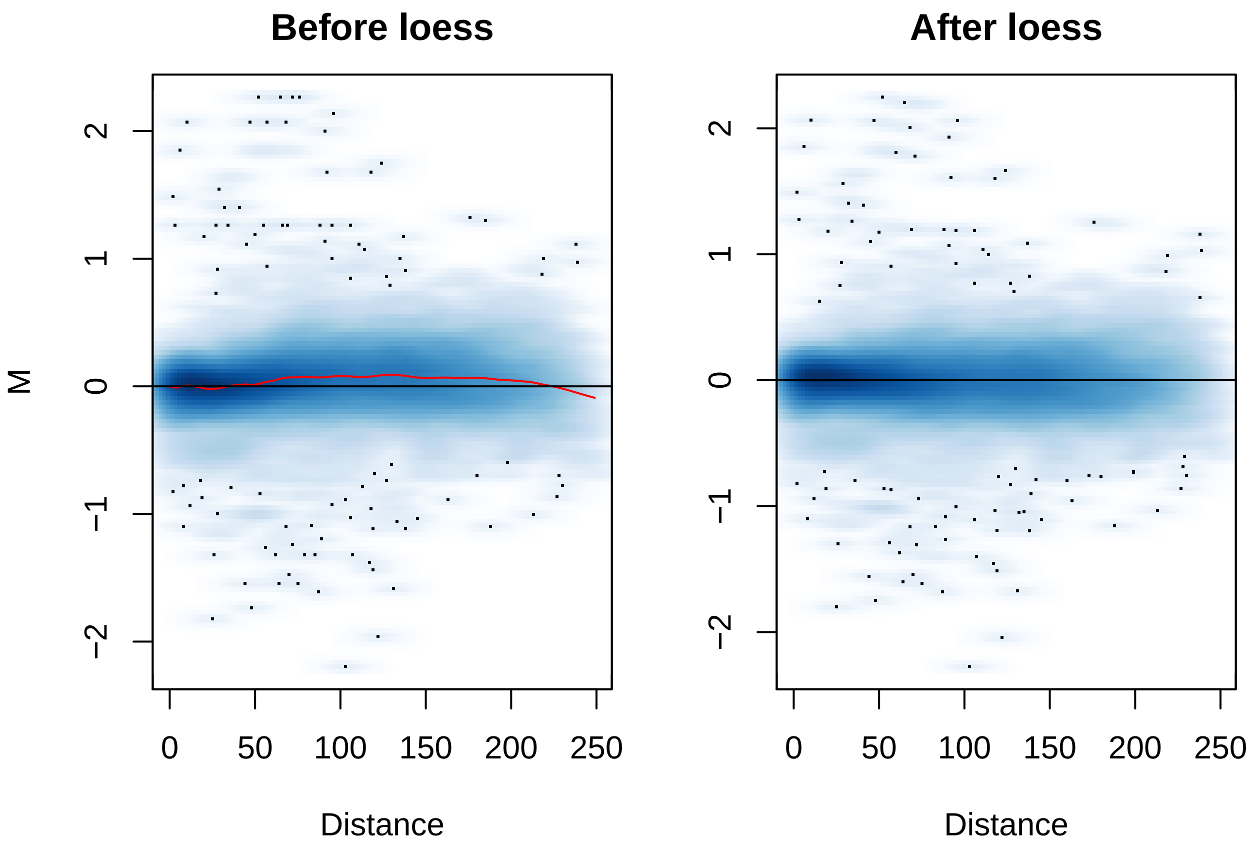


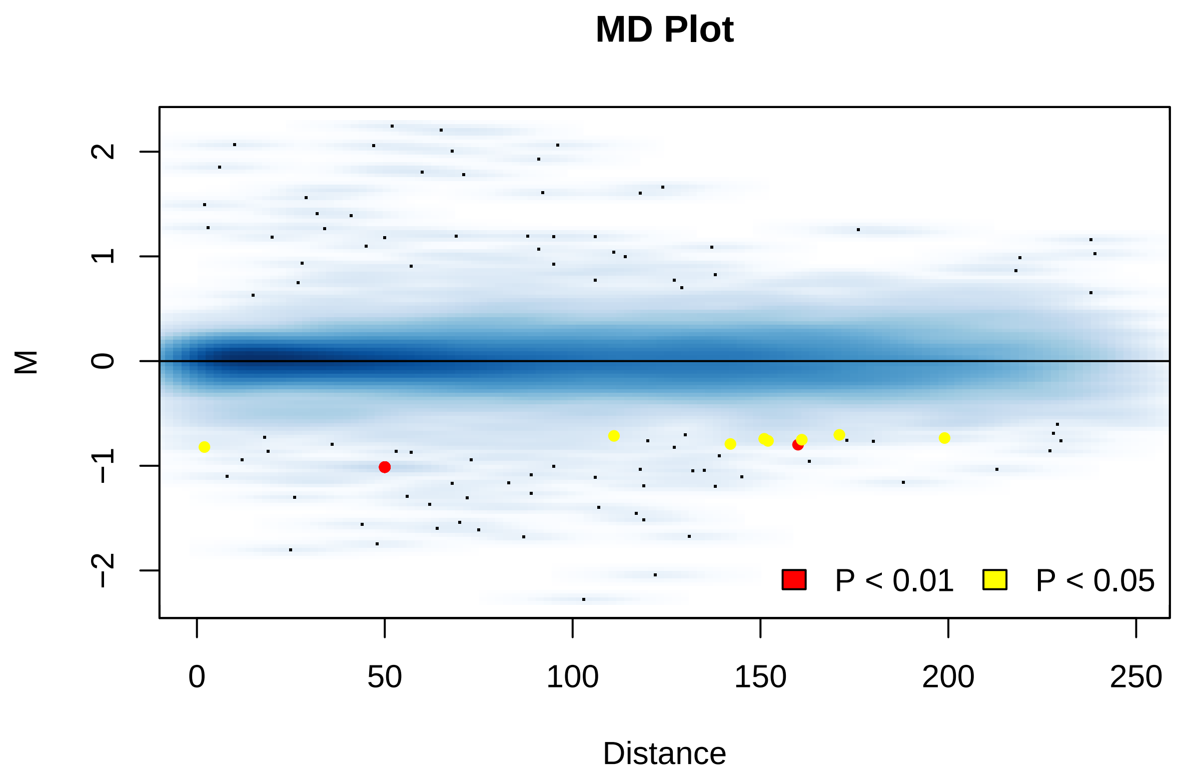
Before loess
After loess
M
Distance
Distance



```
fc2d50 <- make_ROC(chr1.primary, chr1.replicate, FC = 2, numChanges = 10, dist = 50, A.quantile = 0.1, a
```

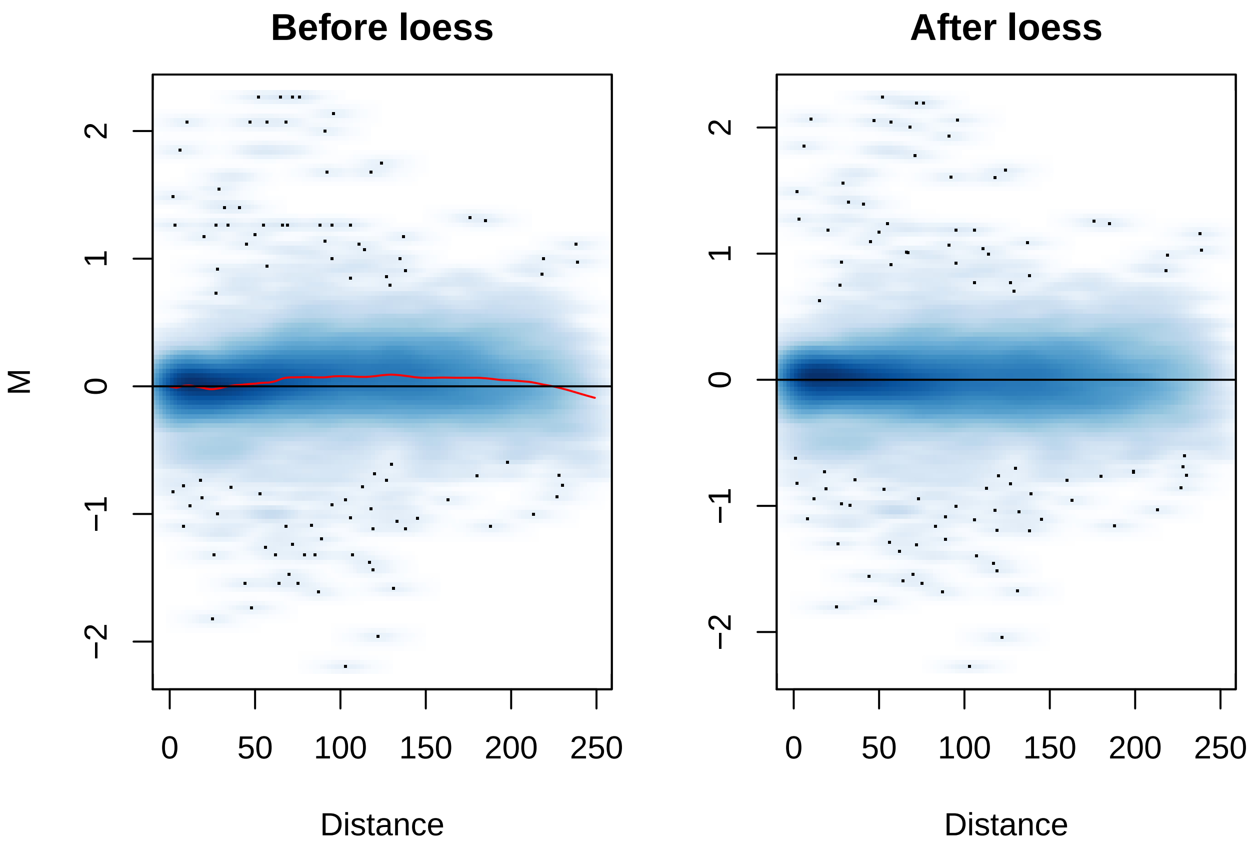
```
## Warning in `[.data.table`(hic.table, changes, `:=`(IF2, meanIF)): Coerced
## 'integer' RHS to 'double' to match the column's type. Either change the
## target column to 'integer' first (by creating a new 'integer' vector length
## 26562 (nrows of entire table) and assign that; i.e. 'replace' column), or
## coerce RHS to 'double' (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: 0.09056038130597

## GCV for loess: 1.57735742598325e-06

## AIC for loess: -2.17054211266886
```

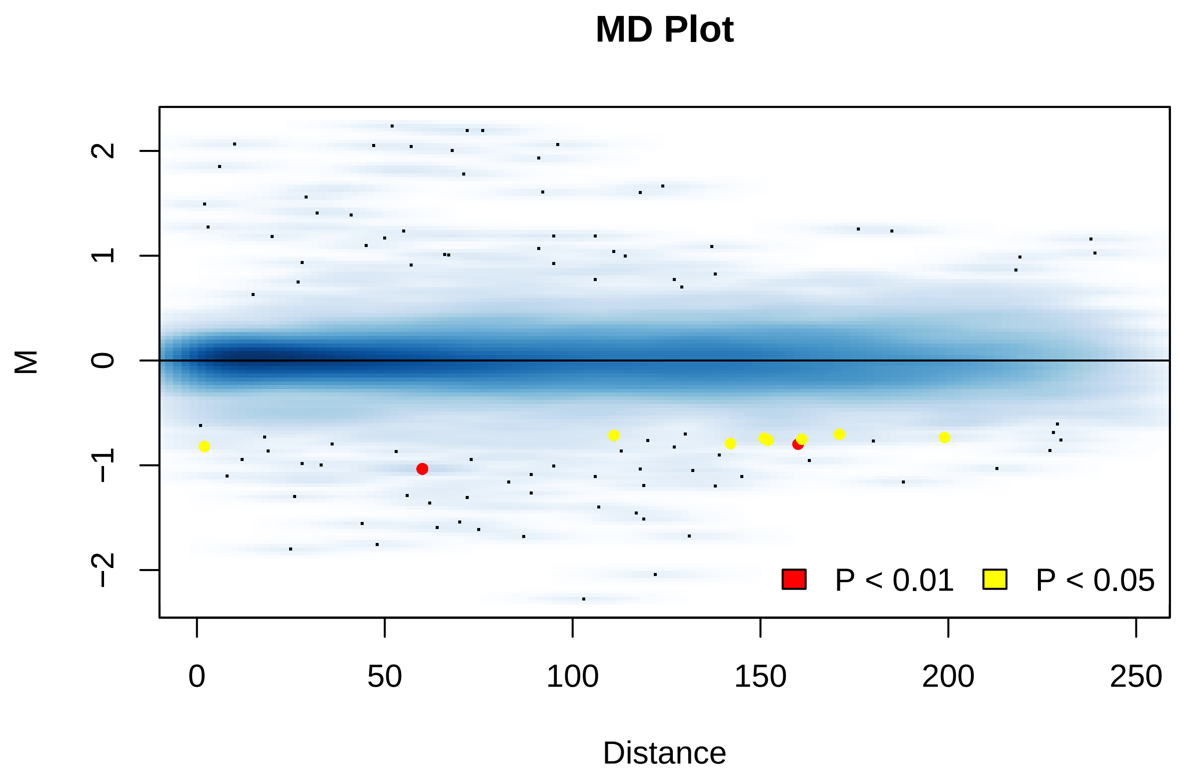

Before loess
After loess
M
2
1
0
−1
−2
0
50
100
150
200
250
Distance
Distance



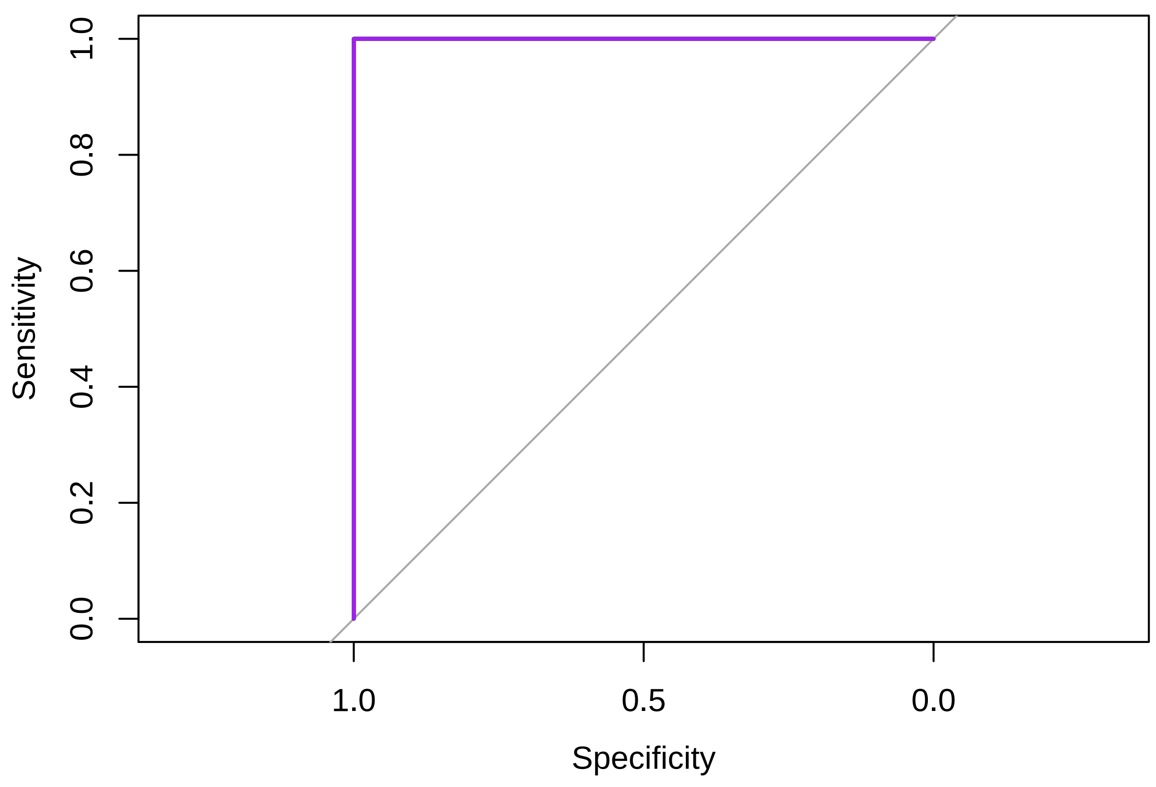
```
fc2d60 <- make_ROC(chr1.primary, chr1.replicate, FC = 2, numChanges = 10, dist = 60, A.quantile = 0.1, a
```

```
## Warning in `[.data.table`(hic.table, changes, `:=`(IF2, meanIF)): Coerced
## 'integer' RHS to 'double' to match the column's type. Either change the
## target column to 'integer' first (by creating a new 'integer' vector length
## 26562 (nrows of entire table) and assign that; i.e. 'replace' column), or
## coerce RHS to 'double' (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: 0.0905796084081443

## GCV for loess: 1.5781755678589e-06

## AIC for loess: -2.17002395730826
```


Before loess
After loess
M
Distance
Distance



```r
plot_colors = c('black', 'blue', 'red', 'green', 'purple', 'orange')
plot(fc2d1[[1]])
plot(fc2d10[[1]], col = plot_colors[2], add = TRUE)
plot(fc2d40[[1]], col = plot_colors[3], add = TRUE)
plot(fc2d50[[1]], col = plot_colors[4], add = TRUE)
plot(fc2d60[[1]], col = plot_colors[5], add = TRUE)
```



```
sum(fc2d1[[2]][truth == 1,]$p.adj < 0.05)
```

```
## [1] 10
```

```
sum(fc2d10[[2]][truth == 1,]$p.adj < 0.05)
```

```
## [1] 10
```

```
sum(fc2d40[[2]][truth == 1,]$p.adj < 0.05)
```

```
## [1] 10
```

```
sum(fc2d50[[2]][truth == 1,]$p.adj < 0.05)
```

```
## [1] 10
```

```
sum(fc2d60[[2]][truth == 1,]$p.adj < 0.05)
```

```
## [1] 10
```