check KR

John Stansfield

December 19, 2017

```
library(HiCcompare)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(chromoR)
```

```
## Loading required package: haarfisz
```

```
## Loading required package: wavethresh
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## WaveThresh: R wavelet software, release 4.6.8, installed
```

```
## Copyright Guy Nason and others 1993-2016
```

```
## Note: nlevels has been renamed to nlevelsWT
```

```
## Loading required package: gdata
```

```
## gdata: Unable to locate valid perl interpreter
## gdata:
## gdata: read.xls() will be unable to read Excel XLS and XLSX files
## gdata: unless the 'perl=' argument is used to specify the location
## gdata: of a valid perl intrpreter.
## gdata:
## gdata: (To avoid display of this message in the future, please
## gdata: ensure perl is installed and available on the executable
## gdata: search path.)
```

```
## gdata: Unable to load perl libaries needed by read.xls()
## gdata: to support 'XLX' (Excel 97-2004) files.
```

```
##
```

```
## gdata: Unable to load perl libaries needed by read.xls()
## gdata: to support 'XLSX' (Excel 2007+) files.
```

```
##
## gdata: Run the function 'installXLSXsupport()'
## gdata: to automatically download and install the perl
## gdata: libaries needed to support Excel XLS and XLSX formats.
```

```
##
## Attaching package: 'gdata'
```

```
## The following objects are masked from 'package:dplyr':
##
##     combine, first, last
```

```
## The following object is masked from 'package:stats':
##
##     nobs
```

```
## The following object is masked from 'package:utils':
##
##     object.size
```

```
## The following object is masked from 'package:base':
##
##     startsWith
```

```r
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(MLmetrics)
```

```
##
## Attaching package: 'MLmetrics'
```

```
## The following object is masked from 'package:base':
##
##     Recall
```

```r
library(HiTC)
```

```
## Loading required package: IRanges
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
## The following object is masked from 'package:pROC':
##
##     var

## The following object is masked from 'package:gdata':
##
##     combine

## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, cbind, colMeans,
##     colnames, colSums, do.call, duplicated, eval, evalq, Filter,
##     Find, get, grep, grepl, intersect, is.unsorted, lapply,
##     lengths, Map, mapply, match, mget, order, paste, pmax,
##     pmax.int, pmin, pmin.int, Position, rank, rbind, Reduce,
##     rowMeans, rownames, rowSums, sapply, setdiff, sort, table,
##     tapply, union, unique, unsplit, which, which.max, which.min

## Loading required package: S4Vectors

## Loading required package: stats4

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:gdata':
##
##     first, first<-

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:base':
##
##     expand.grid

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:gdata':
##
##     trim

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

##
## Attaching package: 'HiTC'
## The following object is masked from 'package:dplyr':
##
##     id
```

```r
library(Matrix)
```

```
## Warning: package 'Matrix' was built under R version 3.4.3

##
## Attaching package: 'Matrix'

## The following object is masked from 'package:S4Vectors':
##
##     expand
```

```r
library(GenomicRanges)
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:BiocGenerics':
##
##     combine

## The following object is masked from 'package:gdata':
##
##     combine

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following object is masked from 'package:GenomicRanges':
##
##     shift

## The following object is masked from 'package:IRanges':
##
##     shift

## The following objects are masked from 'package:S4Vectors':
##
##     first, second

## The following objects are masked from 'package:gdata':
##
##     first, last

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```r
# load data
githubURL <- "https://github.com/dozmorovlab/HiCdiff/raw/supplemental/Supplemental_data/S1_File_data.RDa
load(url(githubURL))
```

```
## `hic.table` format

chr1.tab      <- create.hic.table(S1.dpnii.chr1,   S1.mbol.chr1,        chr = 'chr1')
chr11.tab     <- create.hic.table(S1.dpnii.chr11,  S1.mbol.chr11,       chr = 'chr11')
chr18.tab     <- create.hic.table(S1.dpnii.chr18,  S1.mbol.chr18,       chr = 'chr18')
chr19.tab     <- create.hic.table(S1.dpnii.chr19,  S1.mbol.chr19,       chr = 'chr19')
replicate.tab <- create.hic.table(S1.primary.chr1, S1.replicate.chr1,   chr = 'chr1')
rep.chr11.tab <- create.hic.table(S1.primary.chr11, S1.replicate.chr11, chr = 'chr1')
rep.chr18.tab <- create.hic.table(S1.primary.chr18, S1.replicate.chr18, chr = 'chr1')
rep.chr19.tab <- create.hic.table(S1.primary.chr19, S1.replicate.chr19, chr = 'chr1')

unscaled.tab       <- create.hic.table(S1.dpnii.chr1,  S1.mbol.chr1,  chr='chr1',  scale=T)
chr11.unscaled.tab <- create.hic.table(S1.dpnii.chr11, S1.mbol.chr11, chr='chr11', scale=T)

# BEDPE-like hic.table object
#head(chr1.tab)
```

default KR

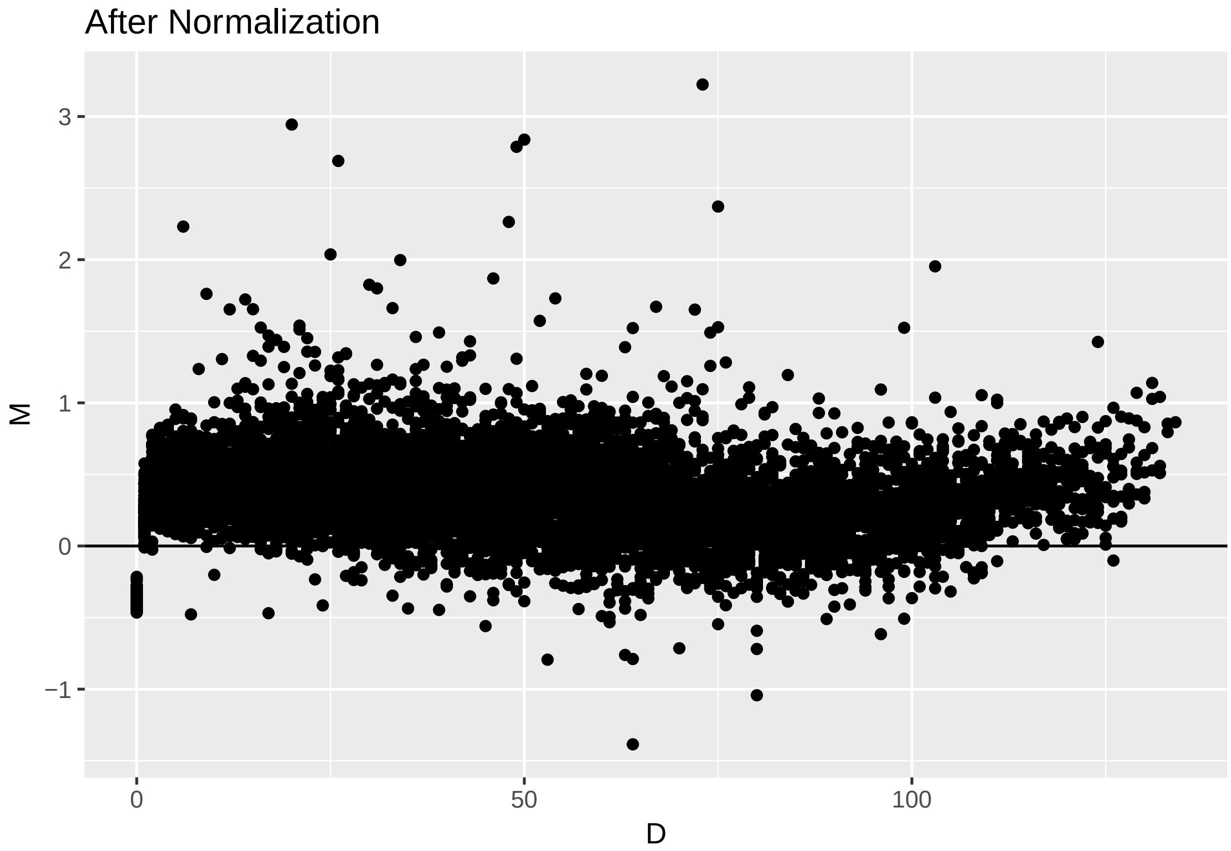
```
mat1 = sparse2full(chr11.tab[, c('start1', 'start2', 'IF1'), with=F])
```

```
## Matrix dimensions: 135x135
```

```
mat2 = sparse2full(chr11.tab[, c('start1', 'start2', 'IF2'), with=F])
```

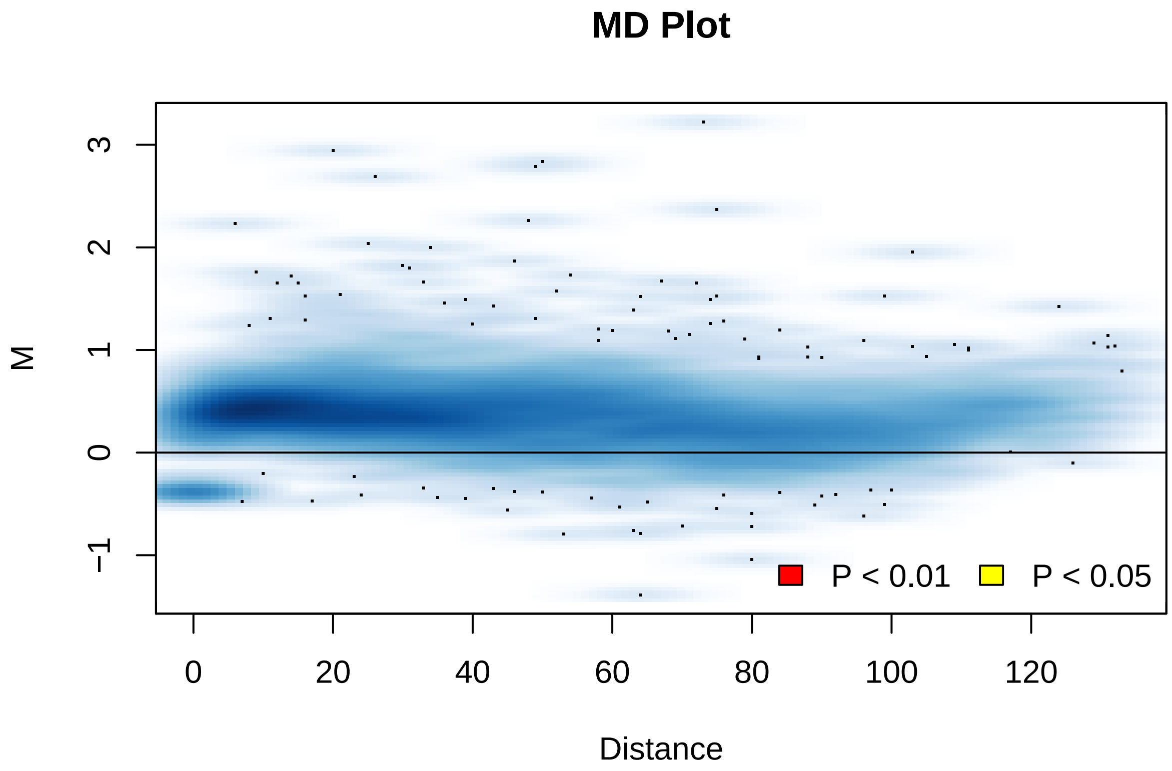
```
## Matrix dimensions: 135x135
```

```
zeros1 = which(colSums(mat1) == 0)
zeros2 = which(colSums(mat2) == 0)
if (length(zeros1) > 0) {
  cr.mat1 = mat1[-zeros1, -zeros1]
} else {
  cr.mat1 = mat1
}
if (length(zeros2) > 0) {
  cr.mat2 = mat2[-zeros2, -zeros2]
} else {
  cr.mat2 = mat2
}
sim1.kr = KRnorm(cr.mat1)
sim2.kr = KRnorm(cr.mat2)
colnames(sim1.kr) = colnames(cr.mat1)
colnames(sim2.kr) = colnames(cr.mat2)
sim1.kr = full2sparse(sim1.kr)
sim2.kr = full2sparse(sim2.kr)
kr.table = create.hic.table(sim1.kr, sim2.kr, scale = FALSE, chr = 'chr11')

kr.table[, ':=' (adj.IF1 = IF1, adj.IF2 = IF2, adj.M = M)]

# p1 = MD.plot2(tab$M, tab$D, smooth = FALSE) + ggtitle('Before Normalization')
MD.plot2(kr.table$M, kr.table$D, smooth = FALSE) + ggtitle('After Normalization')
```



```
# grid.arrange(p1, p2, ncol = 2)


diffs = hic_compare(kr.table, Plot = T,  adjust_dist = FALSE, Plot.smooth = FALSE)
```

```
## Warning in `[<-.data.table`(`*tmp*`, is.na(Z), , value = structure(list(:
## Supplied 17 columns to be assigned a list (length 18) of values (1 unused)
```



```
counts = sum(diffs$p.adj < 0.05)
print(paste0(counts, ' differences found between the datasets'))
```

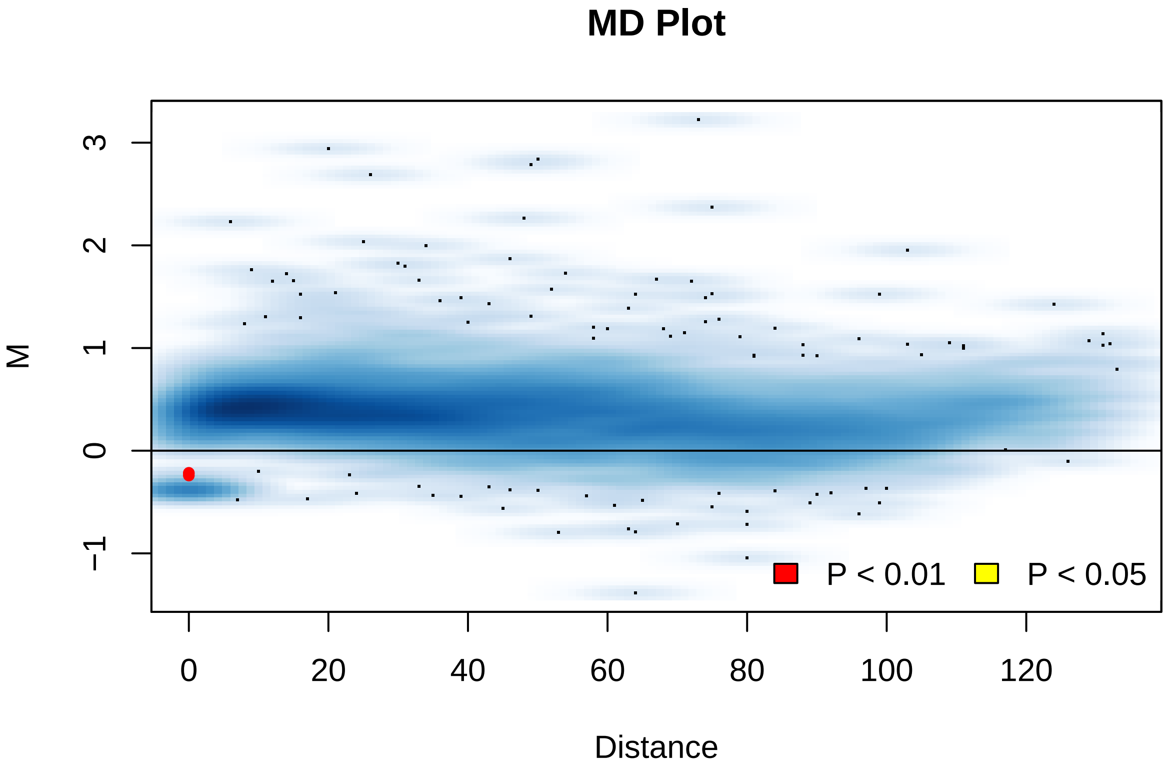
```
## [1] "0 differences found between the datasets"
```

KR multiply matrix by 10k

```
zeros1 = which(colSums(mat1) == 0)
zeros2 = which(colSums(mat2) == 0)
if (length(zeros1) > 0) {
  cr.mat1 = mat1[-zeros1, -zeros1]
} else {
  cr.mat1 = mat1
}
if (length(zeros2) > 0) {
  cr.mat2 = mat2[-zeros2, -zeros2]
} else {
  cr.mat2 = mat2
}
sim1.kr = KRnorm(cr.mat1)
sim2.kr = KRnorm(cr.mat2)
colnames(sim1.kr) = colnames(cr.mat1)
colnames(sim2.kr) = colnames(cr.mat2)
sim1.kr = full2sparse(sim1.kr)
```

```
sim2.kr = full2sparse(sim2.kr)
kr.table = create.hic.table(sim1.kr, sim2.kr, scale = FALSE, chr = 'chr11')

kr.table[, ':=' (adj.IF1 = 10000 * IF1, adj.IF2 = 10000 * IF2)]
kr.table[, adj.M := log2(adj.IF2 / adj.IF1)]

# p1 = MD.plot2(tab$M, tab$D, smooth = FALSE) + ggtitle('Before Normalization')
p2 = MD.plot2(kr.table$M, kr.table$D, smooth = FALSE) + ggtitle('After Normalization')
# grid.arrange(p1, p2, ncol = 2)


diffs = hic_compare(kr.table, Plot = T,  adjust_dist = FALSE, Plot.smooth = FALSE)
```

```
## Warning in `[<-.data.table`(`*tmp*`, is.na(Z), , value = structure(list(:
## Supplied 17 columns to be assigned a list (length 18) of values (1 unused)
```



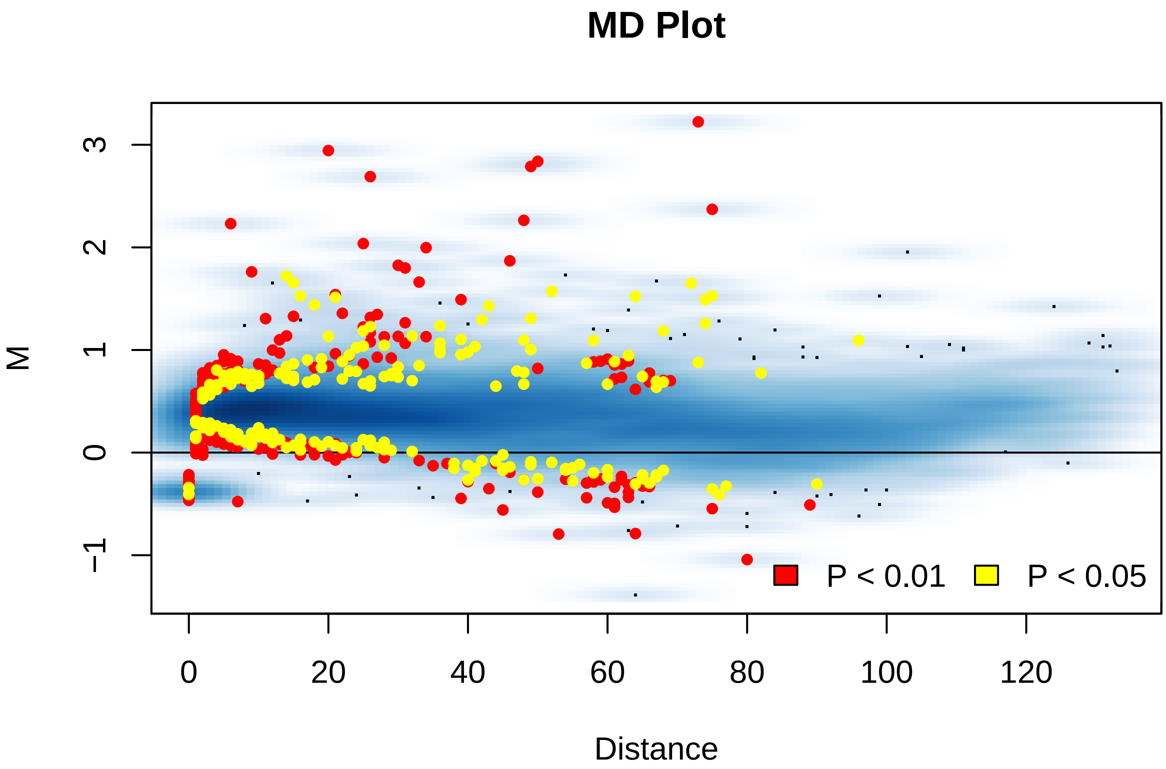
```
counts = sum(diffs$p.adj < 0.05)
print(paste0(counts, ' differences found between the datasets'))
```

```
## [1] "3 differences found between the datasets"
```

KR multiply matrix by 100k

```
zeros1 = which(colSums(mat1) == 0)
zeros2 = which(colSums(mat2) == 0)
if (length(zeros1) > 0) {
  cr.mat1 = mat1[-zeros1, -zeros1]
} else {
  cr.mat1 = mat1
}
if (length(zeros2) > 0) {
  cr.mat2 = mat2[-zeros2, -zeros2]
} else {
  cr.mat2 = mat2
}
sim1.kr = KRnorm(cr.mat1)
sim2.kr = KRnorm(cr.mat2)
colnames(sim1.kr) = colnames(cr.mat1)
colnames(sim2.kr) = colnames(cr.mat2)
sim1.kr = full2sparse(sim1.kr)
sim2.kr = full2sparse(sim2.kr)
kr.table = create.hic.table(sim1.kr, sim2.kr, scale = FALSE, chr = 'chr11')

kr.table[, ':=' (adj.IF1 = 100000 * IF1, adj.IF2 = 100000 * IF2)]
kr.table[, adj.M := log2(adj.IF2 / adj.IF1)]

# p1 = MD.plot2(tab$M, tab$D, smooth = FALSE) + ggtitle('Before Normalization')
p2 = MD.plot2(kr.table$M, kr.table$D, smooth = FALSE) + ggtitle('After Normalization')
# grid.arrange(p1, p2, ncol = 2)


diffs = hic_compare(kr.table, Plot = T,  adjust_dist = FALSE, Plot.smooth = FALSE)
```

```
## Warning in `[<-.data.table`(`*tmp*`, is.na(Z), , value = structure(list(:
## Supplied 17 columns to be assigned a list (length 18) of values (1 unused)
```



```
counts = sum(diffs$p.adj < 0.05)
print(paste0(counts, ' differences found between the datasets'))
```

```
## [1] "512 differences found between the datasets"
```