The exact algorithm for locating and visualizing SIR sequence motifs in microduplication and microdeletion syndromes is listed below:

```
library(BSgenome.Hsapiens.ucsc.hg38)
genome <- BSgenome.Hsapiens.ucsc.hg38
seqnames <- seqnames(genome)
hjtable <- read.table("C:\\hj.txt", header = TRUE)
for (i in 1:nrow(hjtable))
{
for (seqname in seqnames)
{
x <- genome[[seqname]] 113
#print(paste("Run HJ no. ", seqname, i, hjtable$HJ[i]))
Lpattern <- toString(hjtable$HJ[i])
M1 <- matchPattern(Lpattern, x, max.mismatch=0, min.mismatch=0,
with.indels=FALSE, fixed=TRUE, algorithm="auto")
#print(paste("M1 = ", M1))
myfile <- paste("C:\\hj", hjtable$HJ[i],
".bed", sep="")
sink(myfile, append=TRUE)
if (length(M1) != 0)
```



```
{
chrvalue <- seqname
for (k in 1:length(M1))
cat(chrvalue)
cat("\t")
cat(start(M1[k]))
cat("\t")
cat(end(M1[k]))
cat("\t")
cat(toString(hjtable$HJ[i]))
cat("\n")
}
}
sink()
}
library("regioneR")
test <- import(myfile)
test
Microdel.hg38_path <- system.file("tests", package = "rtracklayer")
Microdel.hg38_bed <- file.path(Microdel.hg38_path, "Microdel.hg38.bed")
Microdel.hg38 <- import(Microdel.hg38_bed)
Microdel.hg38
pt <- overlapPermTest(A=Microdel.hg38, B=test, ntimes=1000)
pt
lz <- localZScore(pt=pt, A=Microdel.hg38, B=test)
plot(lz)
plot(pt)
output <- overlapRegions(test, Microdel.hg38)
output <- paste("C:\\op", hjtable$HJ[i],
".bed", sep="")
}
```

The exact algorithm for locating and visualizing PRDM9 sequence motifs in microduplication and microdeletion syndromes is listed below:

```
library(BSgenome.Hsapiens.UCSC.hg19)
genome <- BSgenome.Hsapiens.UCSC.hg19
seqnames <- seqnames(genome)
prdm9table <- read.table("C:\\temp\\prdm9.txt", header =
TRUE)
for (i in 1:1024)
{
 for (seqname in seqnames)
 {
 x <- genome[[seqname]]
 #print(paste("Run PRDM9 no. ", seqname, i, prdm9table$PRDM9[i]))
 Lpattern <- toString(prdm9table$PRDM9[i])
 M1 <- matchPattern(Lpattern, x, max.mismatch=0, min.mismatch=0,
with.indels=FALSE, fixed=TRUE, algorithm="auto")
 #print(paste("M1 = ", M1))
 myfile <- paste("C:\\Work\\GeneScreening\\prdm9\\",
prdm9table$PRDM9[i], ".bed", sep="")
 sink(myfile, append=TRUE)
 if (length(M1) != 0)
 {
 chrvalue <- seqname
 for (k in 1:length(M1))
 {
cat(chrvalue)
 cat("\t")
 cat(start(M1[k]))
 cat("\t")
 cat(end(M1[k]))
 cat("\t")
 cat(toString(prdm9table$PRDM9[i]))
 cat("\n")
 }
 }
 sink()
 }
 myfile2 <- paste("C:\\Work\\GeneScreening\\prdm9\\lz",
prdm9table$PRDM9[i], ".jpg", sep="")
 myfile3 <- paste("C:\\Work\\GeneScreening\\prdm9\\pt",
prdm9table$PRDM9[i], ".jpg", sep="")
 myfile4 <- paste("C:\\Work\\GeneScreening\\prdm9\\output",
prdm9table$PRDM9[i], ".txt", sep="")
library("regioneR")
 test <- import(myfile)
 test
 Microdeletion_path <- system.file("tests", package = "rtracklayer")
 Microdeletion_bed <- file.path(Microdeletion_path, "Microdeletion.bed")
 Microdeletion <- import(Microdeletion_bed)
 Microdeletion
 pt <- overlapPermTest(A=Microdeletion, B=test, ntimes=1000)
 pt
 lz <- localZScore(pt=pt, A=Microdeletion, B=test)
jpeg(myfile2)
 plot(lz)
```

```
 dev.off()
 jpeg(myfile3)
 plot(pt)
 dev.off()
 output <- overlapRegions(test, Microdeletion)
 output
 write.table(output, myfile4, append = FALSE, quote = TRUE, sep = " ",
 eol = "\n", na = "NA", dec = ".", row.names = TRUE,
 col.names = TRUE, qmethod = c("escape", "double"),
 fileEncoding = "")
}
```