```
## Code for Distribution Free Weighted Fold Change (DFW), as described
## in Chen, McGee, Liu, and Scheuermann
##
## To use the following code, copy and paste it into a plain text file.
## Open an R session, source the file into R and run once.
## Then type:
##
## b = ReadAffy()
## This obtains an AffyBatch object from your CEL files
##
## a = expresso(b, bgcorrect.method="none",
##      normalize.method="quantiles", pmcorrect.method="pmonly",
##      summary.method="dfw")
##
## To read output into a CSV file on original scale (makes retrieval
## easier)
##   aa=exprs(a)
##   exprs(a)=2^aa
##   aaa=exprs(a)
##   write.table(data.frame(aaa,check.names=FALSE),
##   "filename.csv",sep=",",col.names=NA,quote=FALSE)

################################################
require(affy)

generateExprSet.methods=c(generateExprSet.methods,"dfw")
express.summary.stat.methods=c(express.summary.stat.methods,"dfw")
generateExprVal.method.dfw=function(probes){

## the weighting function
weights=function(x){
med=median(x)
y=x-med
c=max(abs(y))
if (c==0) w=x else w=(1-(y/c)^2)^2
w=w/sum(w)
return(w)
}
probes=t(t(probes))
npm=nrow(probes)    ## num of PMs for the probeset
narray=ncol(probes) ## num of arrays

fc=log2(probes)    ## relative fold change (log2 scale)

ran=rep(0,npm)
sd=rep(0,npm)
for (i in 1:npm) {if (narray==1) {ran[i]=fc[i,]
                                  sd[i]=1}

      else {ran[i]=max(fc[i,])-min(fc[i,])
            sd[i]=sd(fc[i,])}
     }
## based on ranges and sd's to calculate the weights
wei=weights(ran)
wei.sd=weights(sd)

sd.wei=sum(sd*wei.sd) ## overall (weighted) sd
```

```
## weighted fc for each array (use both range and sd of fc):
fc.wei=fc.wei02=rep(0,narray)

for (i in 1:narray) {
fc.wei[i]=sum(fc[,i]*wei)

}

min=min(fc.wei)
max=max(fc.wei)
range=max-min

if (range==0) fc.wei=0 else {
                fc.wei=(fc.wei-min)/range }

expr=min+fc.wei*range^3*sd.wei/100

express=expr   ## expression values:
return(list(exprs=as.numeric(express),se.exprs=as.numeric(fc.wei)))
}
```