Meta-analysis for Microarray Experiments

Robert Gentleman, Markus Ruschhaupt, Wolfgang Huber, and Lara Lusa

October 30, 2025

1 Introduction

The use of meta-analysis tools and strategies for combining data from microarray experiments seems to be a good and practical idea. Choi et al. (2003) is among the first authors to address these issues. Of great importance in working with these data is the realization that different experiments typically have been designed to address different questions. In general, it will only make sense to combine data sets if the questions are the same, or, if some aspects of the experiments are sufficiently similar that one can hope to make better inference from the whole than from the experiments separately. Just because two experiments were run on the same microarray platform is not sufficient justification for combining them.

In *GeneMeta* we have implemented many of the tools described by (Choi et al., 2003). They focused on the combination of datasets based on two sample comparisons. Hence, their procedures are largely based on the t-test. It is not clear whether improvements would eventuate if some of the more popular adjustments to these tests were used instead.

Consider the situation where data from k trials is available and we want to estimate the mean difference in expression, for each gene, between two commonly measured phenotypes (here we use the term phenotype loosely). The setting considered by Choi et al was that of a tumor versus normal comparison.

The general model for this setting, is as follows. Let μ denote the parameter of interest (the true difference in mean, say). Let y_i denote the measure effect for study i, with $i = 1, \ldots, k$. Then the hierarchical model is:

$$\begin{aligned} y_i &= \theta_i + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma_i^2) \\ \theta_i &= \mu + \delta_i, \qquad \delta_i \sim N(0, \tau^2) \end{aligned}$$

where τ^2 represents the between study variability and σ_i^2 denotes the within study variability. The analysis is different depending on whether a fixed effect model (FEM) is deemed appropriate, or a random effects model (REM) is deemed appropriate. Under a FEM, the basic presumption is that $\tau = 0$. If this does not hold then a REM will need to be fit. The estimates of the overall effect, μ, are different depending on which model is used.

Choi et al. (2003) suggest using an estimator due to DerSimonian and Laird for the REM model. This estimator is computed using the function `tau2.DL`, and its variance via `var.tau2`

Simple Usage

In this vignette we want to show how these methods can be used to combine data sets. Typically matching of identifiers is an important component. We don't want to address the problem here and so just do the following: we split a data set and then combine these two splits. We show that the combination of the splits is as nearly good as the original set. So in this paper we also do not address the problem, that is mentioned above, i.e. to combine only things that are measuring the same thing. In this example we know that the same thing has be measured.

Getting the data

We first load a data sets that were reported by West et al. (2001) and were collected on patients with breast cancer. `Nevins` includes data from 46 hybridizations on hu6800 Affymetrix chips.

```
> library(GeneMeta)
> library(RColorBrewer)
> #load("~/Bioconductor/Projects/GraphCombine/MetaBreast/data/Nevins.RData")
> data(Nevins)
```

We want to look at the estrogen receptor status and find genes that have a high 't-statistic' for the difference between estrogen receptor positive and negative patients. Actually we don't use the t statistic itself but

$$d = t \cdot \sqrt{\frac{n_1 + n_2}{n_1 \cdot n_2}}$$

.
Here t is the 'usual' t-statistic and n_1 and n_2 are the number of elements in the two groups. We create two data sets from the original set by splitting. We make sure that the same fraction of ER positive cases is in each group.

```
> set.seed(1609)
> thestatus  <- pData(Nevins)[,"ER.status"]
> group1     <- which(thestatus=="pos")
> group2     <- which(thestatus=="neg")
> rrr        <- c(sample(group1, floor(length(group1)/2)),
+                 sample(group2,ceiling(length(group2)/2)))
> Split1     <- Nevins[,rrr]
> Split2     <- Nevins[,-rrr]
```

For each data set (Split1 and Split2) we extract the estrogen receptor (ER) status and code it as a 0-1 vector.

```
> #obtain classes
> Split1.ER<-pData(Split1)[,"ER.status"]
> levels(Split1.ER) <- c(0,1)
> Split1.ER<- as.numeric(as.character(Split1.ER))
> Split2.ER<-pData(Split2)[,"ER.status"]
> levels(Split2.ER) <- c(0,1)
> Split2.ER<- as.numeric(as.character(Split2.ER))
```

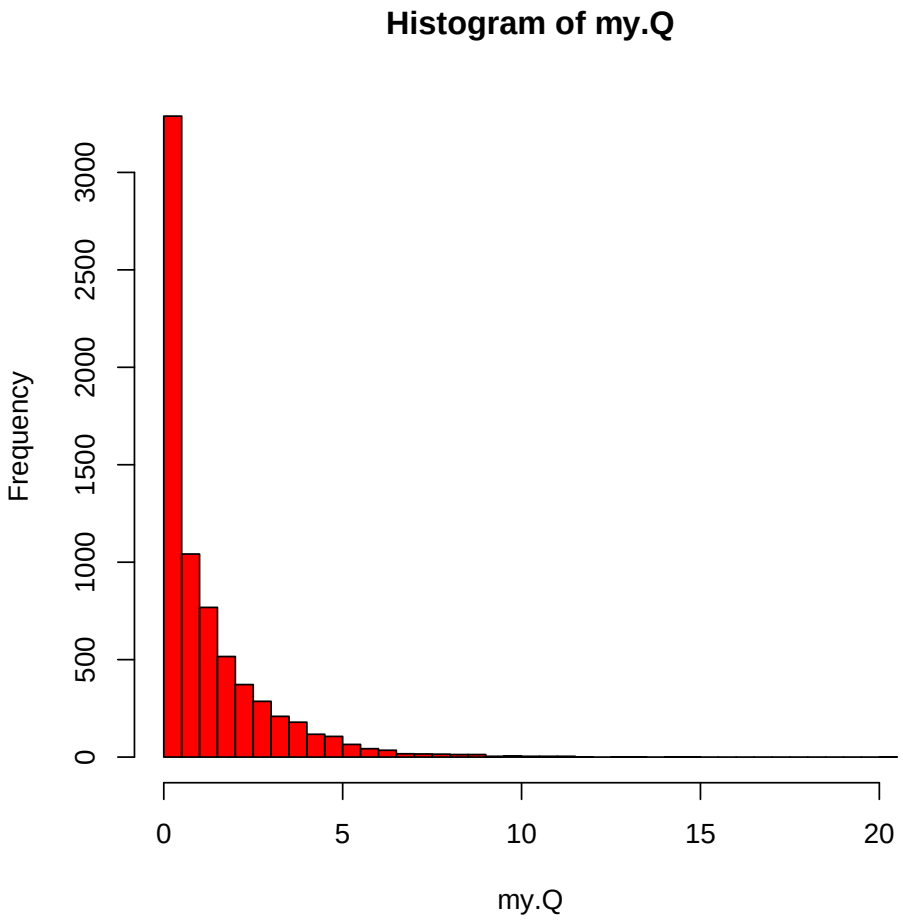
Combining the data

Next we compute the unbiased estimates of the effect (`d.adj.Split1` and `d.adj.Split2`) and its variance (`var.d.adj.Split1` and `var.d.adj.Split2`). Our goal is to compute Cochran's Q statistic to determine whether we should be considering a fixed effects or a random effects model for the data.

```
> #calculate d for Split1
> d.Split1     <- getdF(Split1, Split1.ER)
> #adjust d value
> d.adj.Split1    <- dstar(d.Split1, length(Split1.ER))
> var.d.adj.Spli1 <- sigmad(d.adj.Split1, sum(Split1.ER==0), sum(Split1.ER==1))
> #calculate d for Split2
> d.Split2 <- getdF(Split2, Split2.ER)
> #adjust d value
> d.adj.Split2     <- dstar(d.Split2, length(Split2.ER))
> var.d.adj.Split2 <- sigmad(d.adj.Split2, sum(Split2.ER==0), sum(Split2.ER==1))
>
```

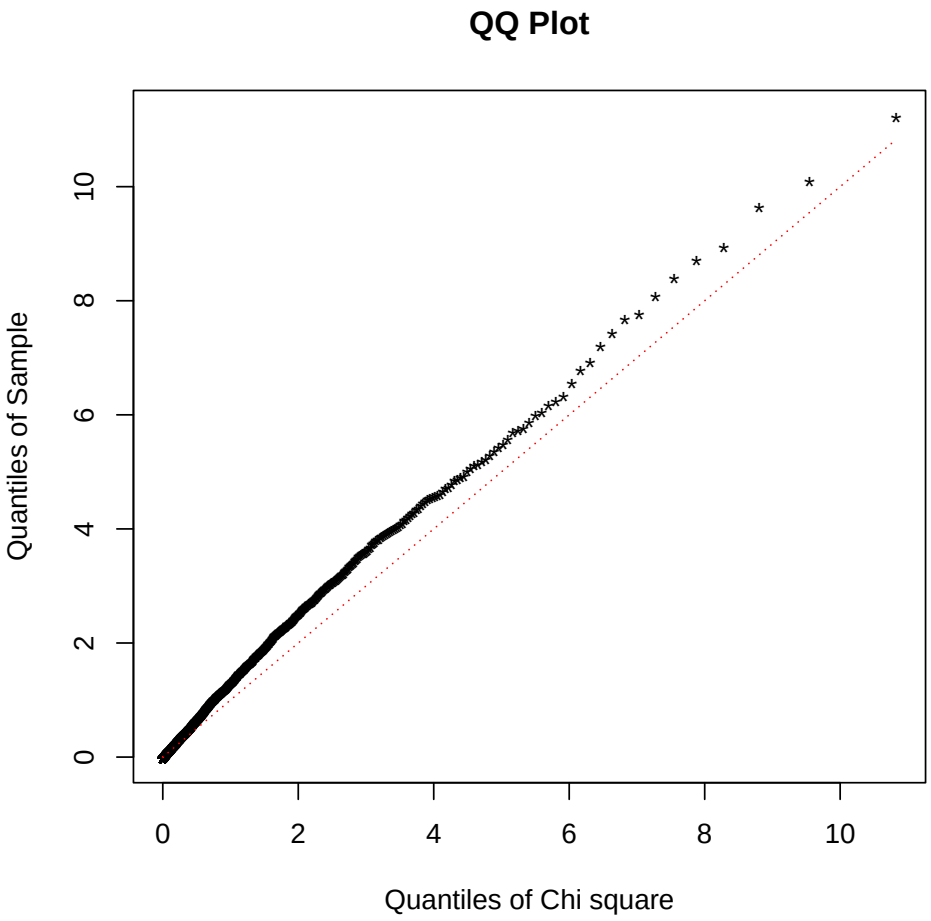
Now, with those in hand we can compute Q and then create and display a qq-plot for comparing the observed values to a χ^2_1 random variable (since we have two experiments).

```
> #calculate Q
> mymns  <- cbind(d.adj.Split1, d.adj.Split2)
> myvars <- cbind(var.d.adj.Spli1,var.d.adj.Split2)
> my.Q   <- f.Q(mymns, myvars)
> mean(my.Q)

[1] 1.229402

> hist(my.Q,breaks=50,col="red")
```



We can see immediately from the histogram and the mean of the Q values that the hypothesis that these values come from a χ^2_1 random variable seems to be valid.

```
> ############ graphics ################
>
> num.studies<-2
> #quantiles of the chisq distribution
> chisqq <- qchisq(seq(0, .9999, .001), df=num.studies-1)
> tmp<-quantile(my.Q, seq(0, .9999, .001))
> qqplot(chisqq, tmp, ylab="Quantiles of Sample",pch="*",
+     xlab="Quantiles of Chi square", main="QQ Plot")
> lines(chisqq, chisqq, lty="dotted",col="red")
```

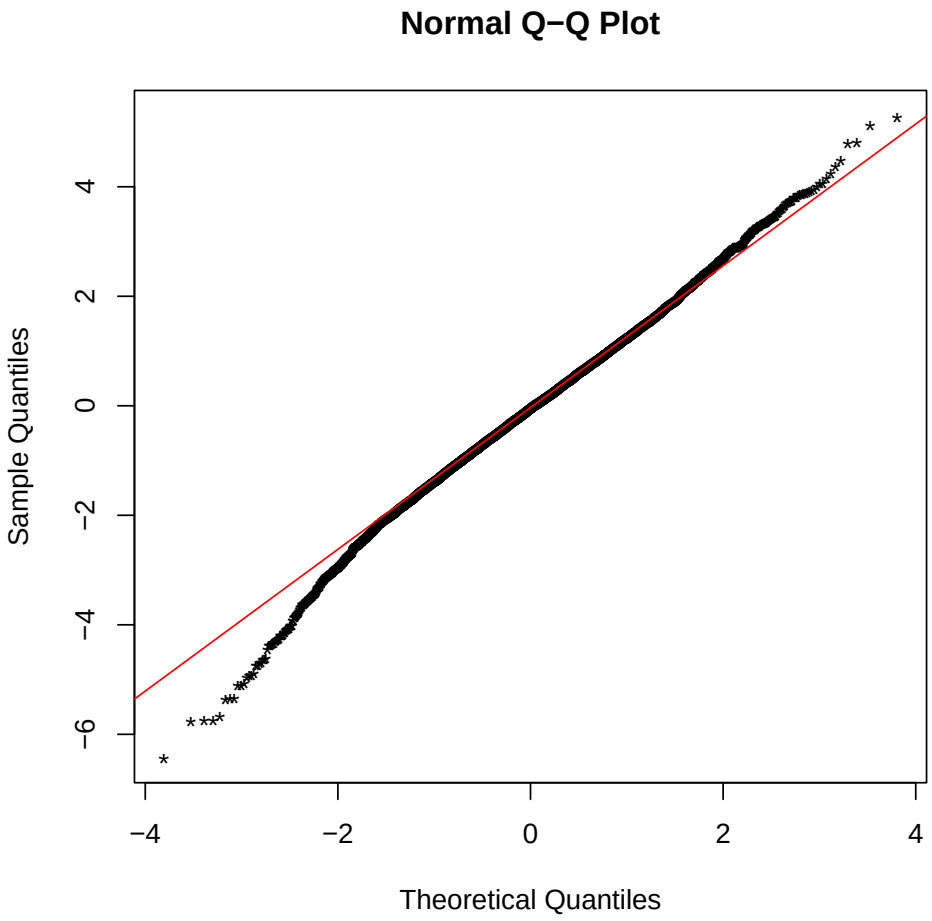


Given that we need to fit a FEM model we next compute the estimated effect sizes. Each effect size is a weighted average of the effects for the individual experiments divided by its standard error. The weights are the reciprocal of the estimated variances.

```
> muFEM = mu.tau2(mymns, myvars)
> sdFEM = var.tau2(myvars)
> ZFEM = muFEM/sqrt(sdFEM)
```

Plotting the quantiles of the effects we can see that the presumption of approximate Normality seems to be appropriate.

```
> qqnorm(ZFEM,pch="*")
> qqline(ZFEM,col="red")
```

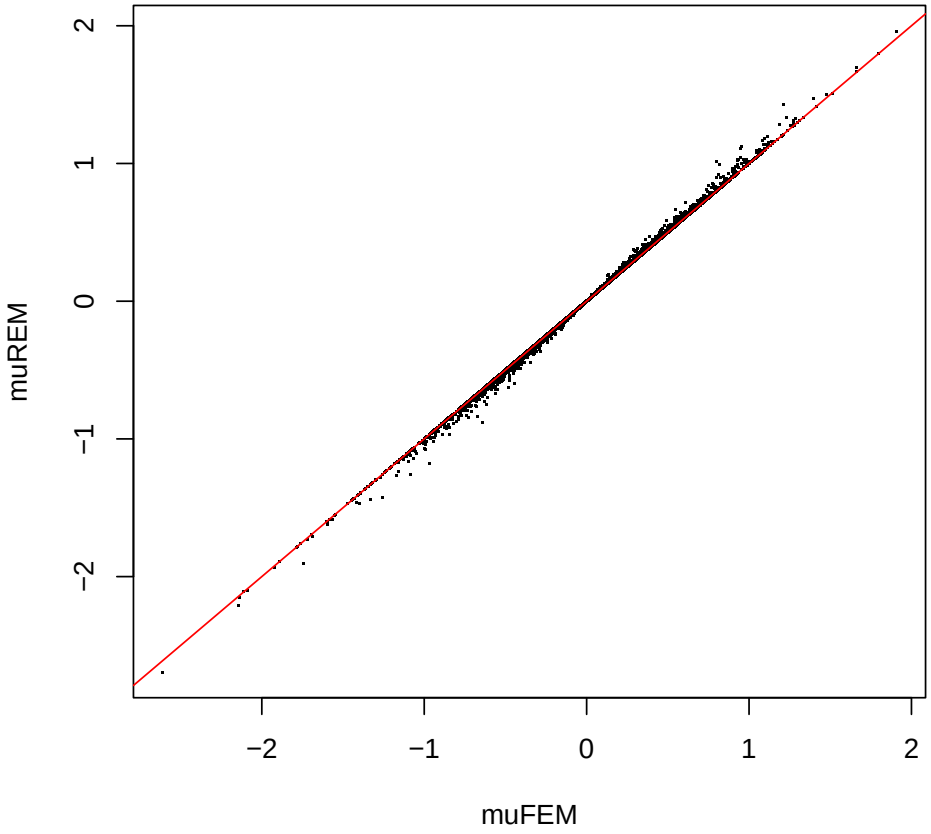


If instead we would have to fit a REM model we would compute the estimated effect sizes using the DerSimonian and Laird estimator. Therefore, we must first estimate the variance τ of the 'between experiments' random variable.

```
> my.tau2.DL<-tau2.DL(my.Q, num.studies, my.weights=1/myvars)
> #obtain new variances s^2+tau^2
> myvarsDL <- myvars + my.tau2.DL
> #compute
> muREM <- mu.tau2(mymns, myvarsDL)
> #cumpute mu(tau)
> varREM <- var.tau2(myvarsDL)
> ZREM <- muREM/sqrt(varREM)
```

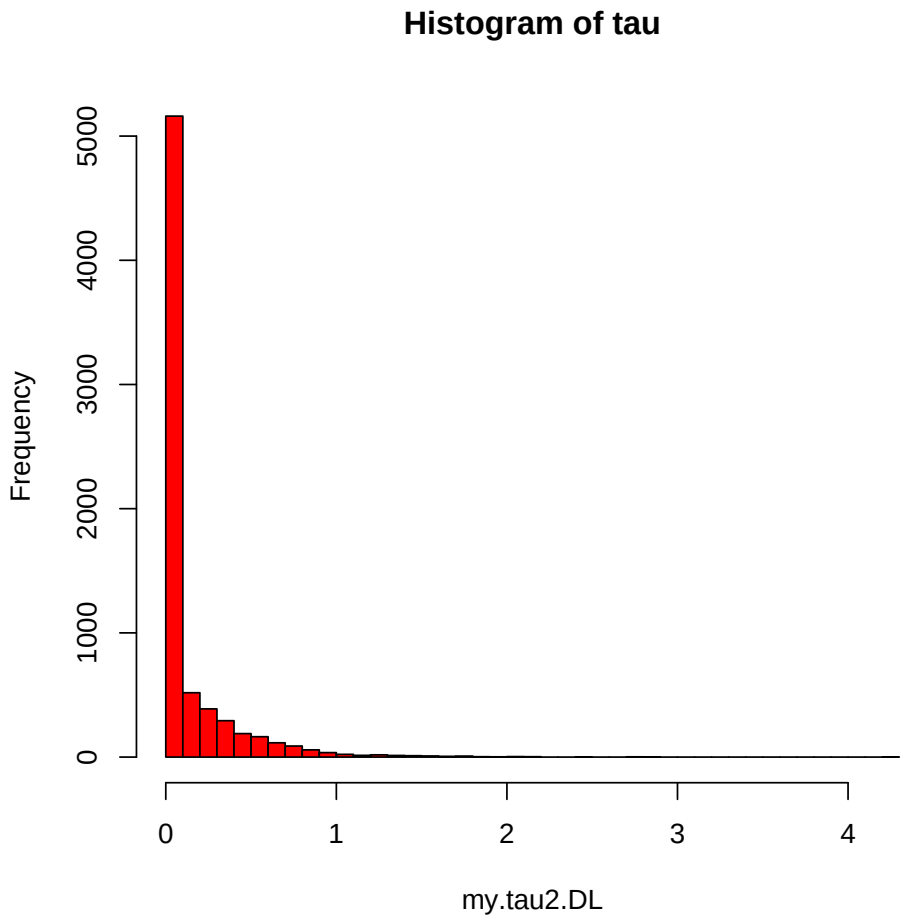
We can easily compare the two different estimates,

```
> plot(muFEM, muREM, pch=".")
> abline(0,1,col="red")
```



We do not see much difference here. This is because in the REM model for most of the genes the variance τ is estimated as zero.

```
> hist(my.tau2.DL,col="red",breaks=50,main="Histogram of tau")
```



The procedure described above is also implemented in the function `zScores` (part of this package) and `meta.summaries` from the package *rmeta*. While `meta.summaries` do the calculation for arbitrary effects and their variances, `zScores` exactly follows the calculation from Choi et al. (2003). The arguments of this function are a list of expression sets and a list of classes. We include our two splits and also the original data set. By default `zScores` would combine all expression sets in the list, but we only want the combine the first two. So we have to set an additional parameter.

```
> esets     <- list(Split1,Split2,Nevins)
> data.ER   <-pData(Nevins)[,"ER.status"]
> levels(data.ER) <- c(0,1)
> data.ER<- as.numeric(as.character(data.ER))
> classes   <- list(Split1.ER,Split2.ER,data.ER)
> theScores <- zScores(esets,classes,useREM=FALSE,CombineExp=1:2)
```

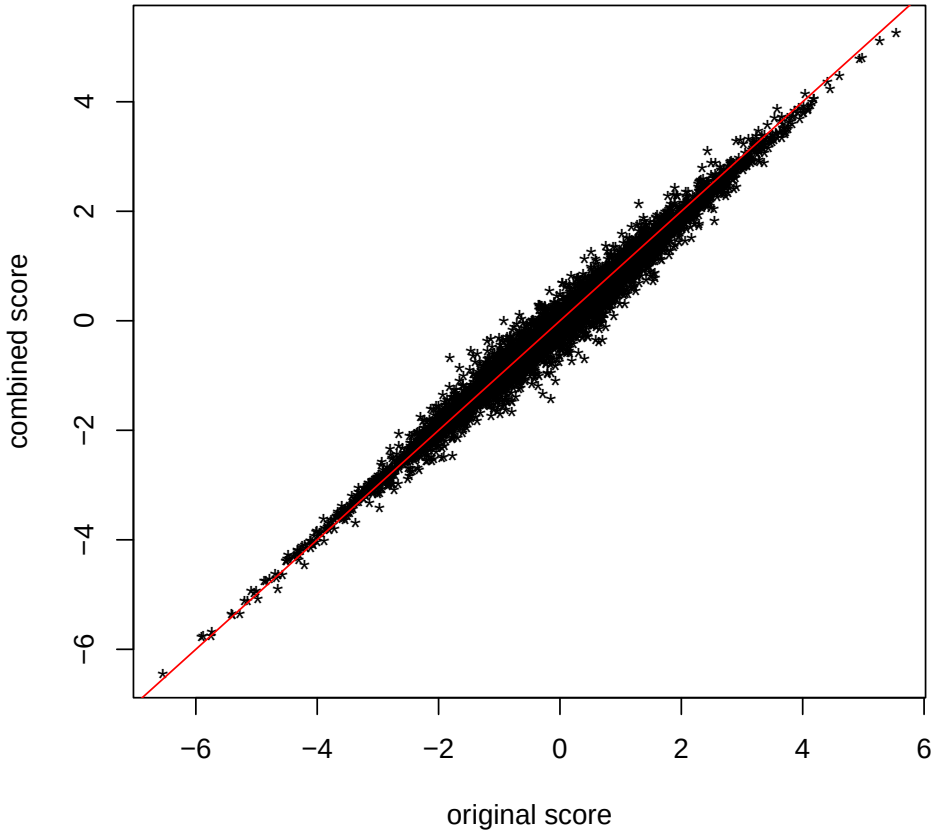
We get a matrix in the following form.

```
> theScores[1:2,]
```

```
            zSco_Ex_1 zSco_Ex_2  zSco_Ex_3       zSco     MUvals     MUsds
A28102_at   0.2933826 -1.518038 -0.5794422 -0.8500399 -0.2542250 0.2990741
AB000114_at 0.5850821 -1.486041 -0.6361189 -0.6217634 -0.1861197 0.2993417
               Qvals df  Qpvalues     Chisq Effect_Ex_1 Effect_Ex_2 Effect_Ex_3
A28102_at   1.667944  1 0.1965341 0.3953029   0.1225795  -0.6501593  -0.1711808
AB000114_at 2.164050  1 0.1412719 0.5340975   0.2451409  -0.6357567  -0.1879951
            EffectVar_Ex_1 EffectVar_Ex_2 EffectVar_Ex_3
A28102_at        0.1745691      0.1834317     0.08727503
AB000114_at      0.1755488      0.1830291     0.08734068
```

Here `Effect_Ex_1` and `Effect_Ex_2` are the unbiased estimates of the effect (`d.adj.Split1` and `d.adj.Split2`). `EffectVar_Ex_1` and `EffectVar_Ex_2` are the estimated variances of the unbiased effects (`var.d.adj.Split1` and `var.d.adj.Split2`). `zSco_Ex_1` and `zSco_Ex_2` are the unbiased estimates of the effects divided by their standard deviation. The same values are also calculated the the complete data set (`Effect_Ex_3`,`EffectVar_Ex_3`, and `ZSco_Ex_3`).
`Qvals` are the Q statistics (`my.Q`) and `df` is the number of combined experiments minus one. `MUvals` and `MUsds` are equal to `muFEM` and `sdFEM` (the overall mean effect size and its standard deviation). `zSco` are the z scores (`ZFEM`). `Qpvalues` is for each gene the probability that a chisq distribution with `df` degree of freedom has a higher value than its Q statistic. And `Chisq` is the probability that a chisq distribution with 1 degree of freedom has a higher value than $\texttt{zSco}^2$.
We plot the z scores of original data set against the z scores of the combined data set. We see a good correlation so the combination of the two data sets works quite well. In the next paragraph we want to see how big the benefit of combining data sets really is.

```
> plot(theScores[,"zSco_Ex_3"],theScores[,"zSco"],pch="*",xlab="original score",ylab="co
> abline(0,1,col="red")
```



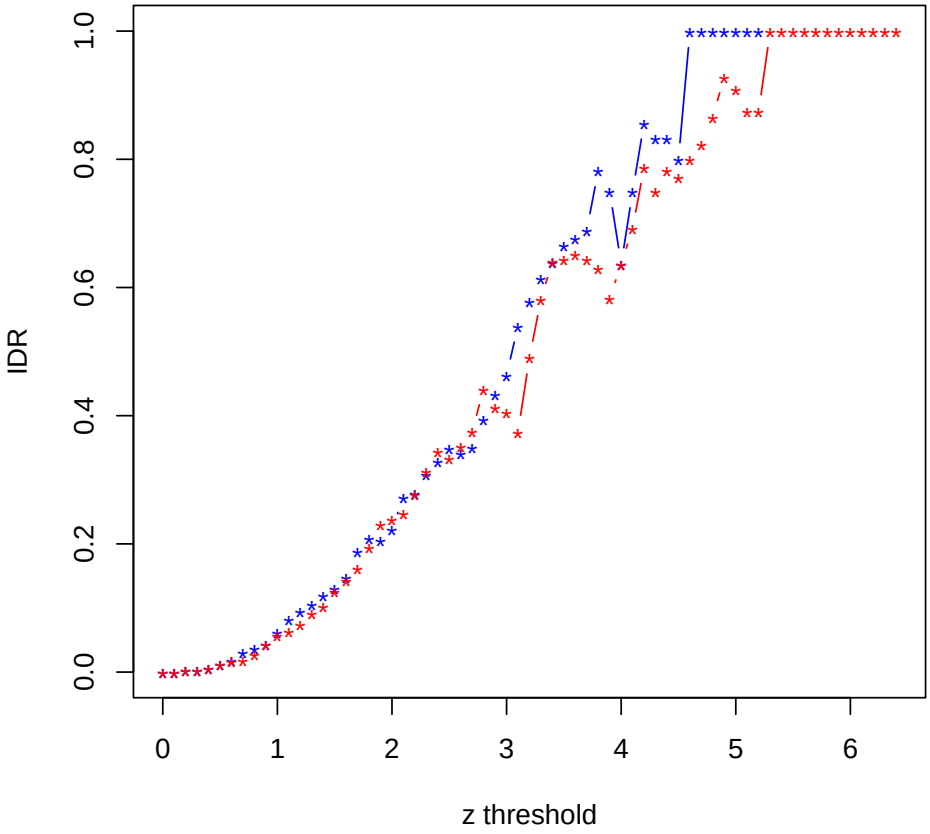
We now will have a look at the IDR plot as it is described in (Choi et al., 2003). For a threshold z_th this plot shows the fraction of the genes that have a higher effect size than the threshold for the combined effect z, but not for any of the experiment specific effects z_i, e.g. we look for genes with

$$z \geq z_{th} \text{ and } \sum_{i=1}^{k} I(z_i \geq z_{th}) = 0 \text{ for } z > 0 \text{ or}$$

$$z \leq -z_{th} \text{ and } \sum_{i=1}^{k} I(z_i \leq -z_{th}) = 0 \text{ for } z < 0$$

The IDR was computed for $z > 0$ (blue) and $z < 0$ (red) separately. We can see that we get higher z scores by combing the sets.

```
> IDRplot(theScores,Combine=1:2,colPos="blue", colNeg="red")
```



Estimating the false discovery rate

Next Choi et al. (2003) discussed using a SAM (Tusher et al., 2001) type analysis to estimate the false discovery rate(FDR). This is implemented in the function `zscoresFDR`.

```
> ScoresFDR <- zScoreFDR(esets, classes, useREM=FALSE, nperm=50,CombineExp=1:2)
```

This object is a list with three slots

```
> names(ScoresFDR)
```

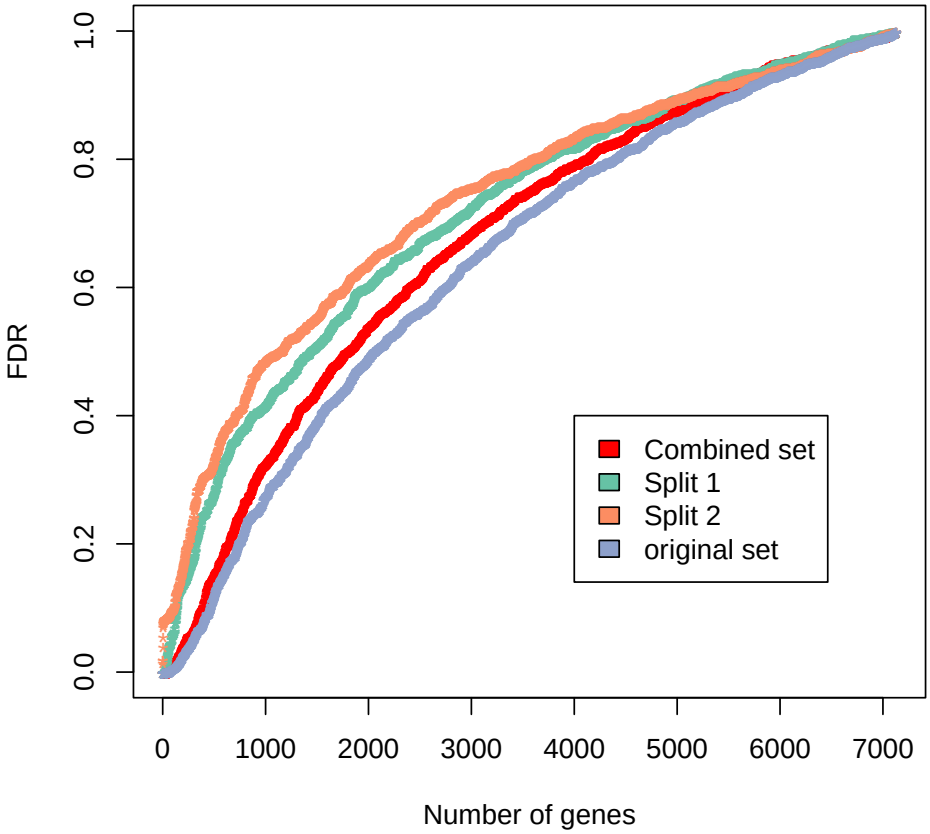
```
[1] "pos"       "neg"       "two.sided"
```

The first slot stores the results of the calculation, if the FDR is computed for the positive scores, the second for the negative scores and the last one for the tow sided situation (i.e. we look at the absolute values of the z scores). Each slot contains a matrix with the values obtained by zScores and additional a FDR for each experiment and the combination of experiments.

```
> ScoresFDR$pos[1:2,]

            zSco_Ex_1  FDR_Ex_1 zSco_Ex_2 FDR_Ex_2  zSco_Ex_3 FDR_Ex_3
A28102_at   0.2933826 0.8432594 -1.518038 1.093965 -0.5794422 1.093660
AB000114_at 0.5850821 0.7700447 -1.486041 1.095971 -0.6361189 1.098239
                 zSco      FDR     MUvals     MUsds    Qvals df  Qpvalues
A28102_at   -0.8500399 1.102665 -0.2542250 0.2990741 1.667944  1 0.1965341
AB000114_at -0.6217634 1.095800 -0.1861197 0.2993417 2.164050  1 0.1412719
                Chisq
A28102_at   0.3953029
AB000114_at 0.5340975
```

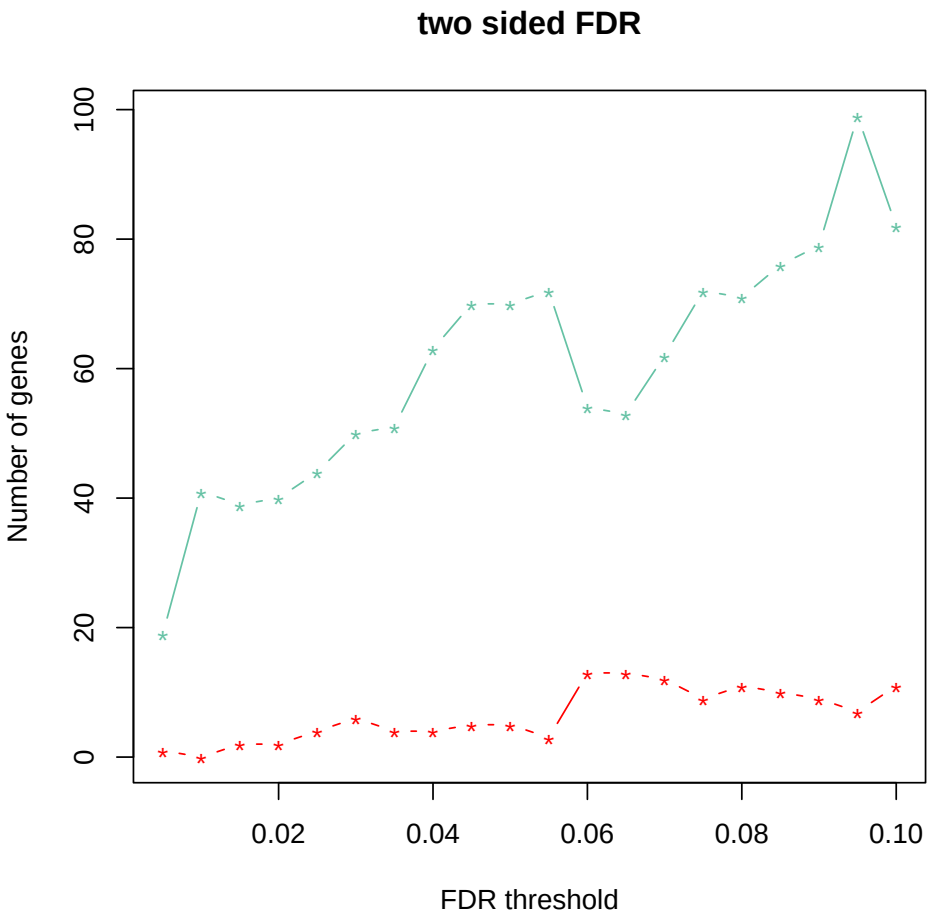
We plot the number of genes and the corresponding FDR. Here the result for the combined set is red and for the result for the original set (without splitting) is blue. We extract the FDR for the two sided situation. It can be see that the combined data set has a lower FDR than the splits and a FDR as good as the original set.

```
> FDRwholeSettwo <- sort(ScoresFDR$"two.sided"[,"FDR"])
> experimentstwo <- list()
> for(j in 1:3){
+ experimentstwo[[j]] <- sort(ScoresFDR$"two.sided"[,paste("FDR_Ex_",j,sep="")])
+ }

> ######################
> #                    #
> #two sided z values #
> #                    #
> ######################
>
> plot(FDRwholeSettwo,pch="*",col="red",ylab="FDR",xlab="Number of genes")
> for(j in 1:3)
+ points(experimentstwo[[j]],pch="*", col=theNewC[j])
> legend(4000,0.4,c("Combined set","Split 1" , "Split 2" ,"original set"), c("red",theNe
```



If we are more interested in the number of gene that are below a given threshold for the FDR we can use the `CountPlot`. Similar to `IDRplot` it shows the following: for each study (indicated by different colors) and various thresholds for the FDR (x axis) the number of genes that are below this threshold in the given study but above in all other studies are shown (y axis). The studies that should be considered (apart from the combined set that is always present) can be specified with `CombineExp`. Here we compare the original data set (green) against the combined data set (red). It can be seen that we do quite well.

```
> #par(mfrow=c(2,2))
> #CountPlot(ScoresFDR,Score="FDR",kindof="neg",cols=c("red",theNewC),
> #        main="Negative FDR", xlab="FDR threshold", ylab="Number of genes",CombineExp=
> #CountPlot(ScoresFDR,Score="FDR",cols=c("red",theNewC),kindof="pos",
> #main="Positive FDR", xlab="FDR threshold", ylab="Number of genes",Combine=1:2)
> CountPlot(ScoresFDR,Score="FDR",kindof="two.sided",cols=c("red",theNewC),main="two sid
```



References

Choi JK, Yu U, Kim S, Yoo OJ. Combining multiple microarray studies and modeling interstudy variation *BI* 19(1): i84-i90 (2003).

West M, Blanchette C, Dressman H and others. Predicting the clinical status of human breast cancer by using gene expression profiles *Proc Natl Acad Sci U S A* 98(20):11462–11467 (2001).

Tusher VG, Tibshirani R, Chu, G. Significance analysis of microarrays applied to the ionizing radiation response *PNAS* 98:5116–5121 (2001).