Supplemental Material: A Bayesian calibration model for combining different pre-processing methods in Affymetrix chips

Marta Blangiardo*[1] and Sylvia Richardson[1]

[1]Centre for Biostatistics, Imperial College. St. Mary's Campus, Norfolk Place London W2 1PG, UK

Email: Marta Blangiardo*- m.blangiardo@imperial.ac.uk;

*Corresponding author

1 Posterior Predictive Check

The Posterior Predictive Check compares the observed sample variance, calculated on the expression values, and the variance of the predicted expression values of each gene under the model using an empirical p-value.
The strategy for calculating the posterior predictive check can be described as follows:

1. A new value for the gene specific component of the variance for each condition ($\sigma_{gk}^{2(pred)}$) is predicted from its prior distribution $Ga^{-1}(a_k, b_k)$

2. new data y_{gjkr}^{pred} are generated under the chosen model, current values of the parameters α_{gj}, δ_g, λ_j, ϕ_j and the predicted variance:

$$y_{gjkr}^{pred} \sim N(\alpha_{gj} + (-1)^k \frac{1}{2} \times \delta_g \times \phi_j, \sigma_{gjk}^{2(pred)}) \qquad k = 1, 2.$$

   where the variability has the same structure presented in (2),

$$\sigma_{gjk}^{2(pred)} = \exp(\lambda_{1jk} + \lambda_{2jk} \times \overline{y}_g + \lambda_{3jk} \times \overline{y}_g^2) \times \sigma_{gk}^{2(pred)}.$$

3. For each gene and pre-processing method the predicted sample variance $(S_{gjk}^{2(pred)} = \frac{1}{2R-1} \sum_{r=1}^{2R} (y_{gjkr}^{(pred)} - \overline{y}_{gjk}^{(pred)})^2)$ is compared to the observed sample variance $(S_{gjk}^{2(obs)} = \frac{1}{2R-1} \sum_{r=1}^{2R} (y_{gjkr} - \overline{y}_{gjk})^2)$ through the statistic $(S_{gjk}^{2(pred)} - S_{gjk}^{2(obs)})$ and a p-value is generated.

Under the null hypothesis of the model being true, the distribution of the p-values should be approximatively uniform, while a poor model fit is indicated by the presence of a notable pattern in the plot, suggesting a systematic difference between the observed values and the predicted ones.

2 Tables and Figures

	$E(\lambda \mid \mathbf{y})$	$CI95\%$
$\lambda_{1,j=1,k=1}$	-5.96	-6.14 , -5.25
$\lambda_{1,j=2,k=1}$	4.30	4.17 , 4.44
$\lambda_{1,j=3,k=1}$	1.66	1.50 , 1.78
$\lambda_{2,j=1,k=1}$	1.22	1.19 , 1.27
$\lambda_{2,j=2,k=1}$	-1.14	-1.18 , -1.10
$\lambda_{2,j=3,k=1}$	-0.08	-0.11 , -0.03
$\lambda_{3,j=1,k=1}$	-0.07	-0.07 , -0.07
$\lambda_{3,j=2,k=1}$	0.07	0.07 , 0.08
$\lambda_{3,j=3,k=1}$	-0.00	-0.01 , 0.01
$\lambda_{1,j=1,k=2}$	-7.07	-7.20 , -6.94
$\lambda_{1,j=2,k=2}$	4.82	4.66 , 4.93
$\lambda_{1,j=3,k=2}$	2.25	2.11 , 2.36
$\lambda_{2,j=1,k=2}$	1.61	1.58 , 1.65
$\lambda_{2,j=2,k=2}$	-1.36	-1.39 , -1.31
$\lambda_{2,j=3,k=2}$	-0.25	-0.28 , -0.21
$\lambda_{3,j=1,k=2}$	-0.10	-0.10 , -0.10
$\lambda_{3,j=2,k=2}$	0.09	0.09 , 0.09
$\lambda_{3,j=3,k=2}$	0.01	0.01 , 0.01

Table 1: Latin Square data set: Posterior Mean and 95% credibility intervals for λ_1, λ_2 and λ_3 in the variability model presented in (2). $j = 1, 2, 3$ indicates the pre-processing method, $k = 1, 2$ indicates the condition. The coefficients satisfy the "sum to 0" constraint.

	$E(\lambda \mid \mathbf{y})$	$CI95\%$
$\lambda_{1,j=1,k=1}$	-4.791	-4.811 , -4.771
$\lambda_{1,j=2,k=1}$	2.550	2.522 , 2.575
$\lambda_{1,j=3,k=1}$	2.241	2.217 , 2.272
$\lambda_{2,j=1,k=1}$	0.743	0.736 , 0.749
$\lambda_{2,j=2,k=1}$	-0.590	-0.598 , -0.576
$\lambda_{2,j=3,k=1}$	-0.153	-0.161 , -0.146
$\lambda_{3,j=1,k=1}$	-0.024	-0.025 , -0.023
$\lambda_{3,j=2,k=1}$	0.028	0.027 , 0.029
$\lambda_{3,j=3,k=1}$	-0.004	-0.005 , -0.003
$\lambda_{1,j=1,k=2}$	-6.604	-6.638 , -6.575
$\lambda_{1,j=2,k=2}$	3.628	3.580 , 3.680
$\lambda_{1,j=3,k=2}$	2.976	2.948 , 3.001
$\lambda_{2,j=1,k=2}$	1.454	1.447 , 1.461
$\lambda_{2,j=2,k=2}$	-0.944	-0.957 , -0.933
$\lambda_{2,j=3,k=2}$	-0.510	-0.515 , -0.502
$\lambda_{3,j=1,k=2}$	-0.081	-0.082 , -0.081
$\lambda_{3,j=2,k=2}$	0.053	0.052 , 0.054
$\lambda_{3,j=3,k=2}$	0.028	0.028 , 0.029

Table 2: HFD vs NFD experiment: Posterior Mean and 95% credibility intervals for λ_1, λ_2 and λ_3 in the variability model presented in (2). $j = 1, 2, 3$ indicates the pre-processing method, $k = 1, 2$ indicates the condition. The coefficients satisfy the "sum to 0" constraint.

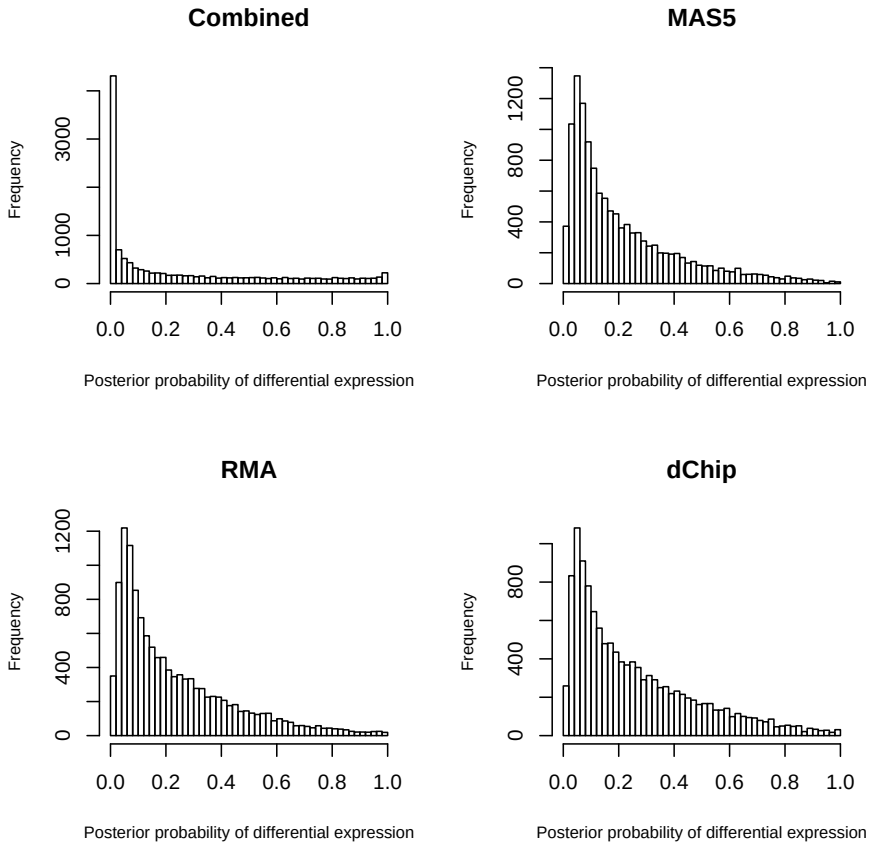

Figure 1: High Fat Diet vs Normal Fat Diet experiment: the histograms show the distribution of the tail posterior probability for the combined method and each pre-processing method separately, considering MAS5, RMA and dChip. In contrast to what happens for each single pre-processing method, the combined model shows a local peak on the right tail of the distribution, indicating evidence of differential expression.

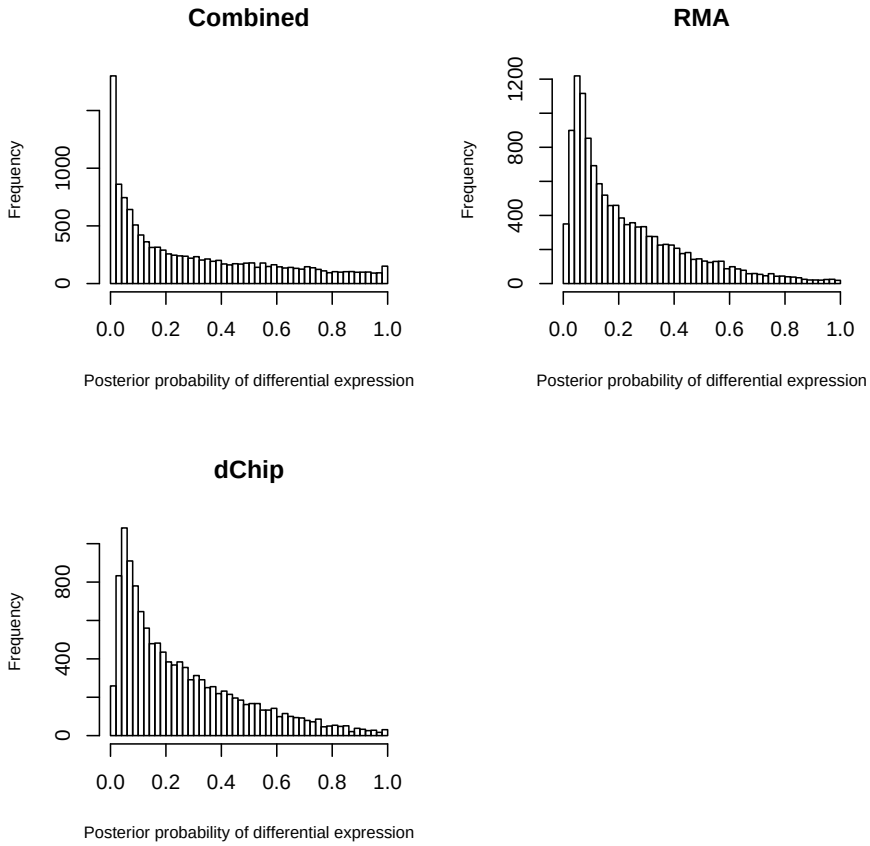

Figure 2: High Fat Diet vs Normal Fat Diet experiment: the histograms show the distribution of the tail posterior probability for the combined method and each pre-processing method separately, when we consider only RMA and dChip. In contrast to what happens for each single pre-processing method, the combined model shows a local peak on the right tail of the distribution, indicating evidence of differential expression.

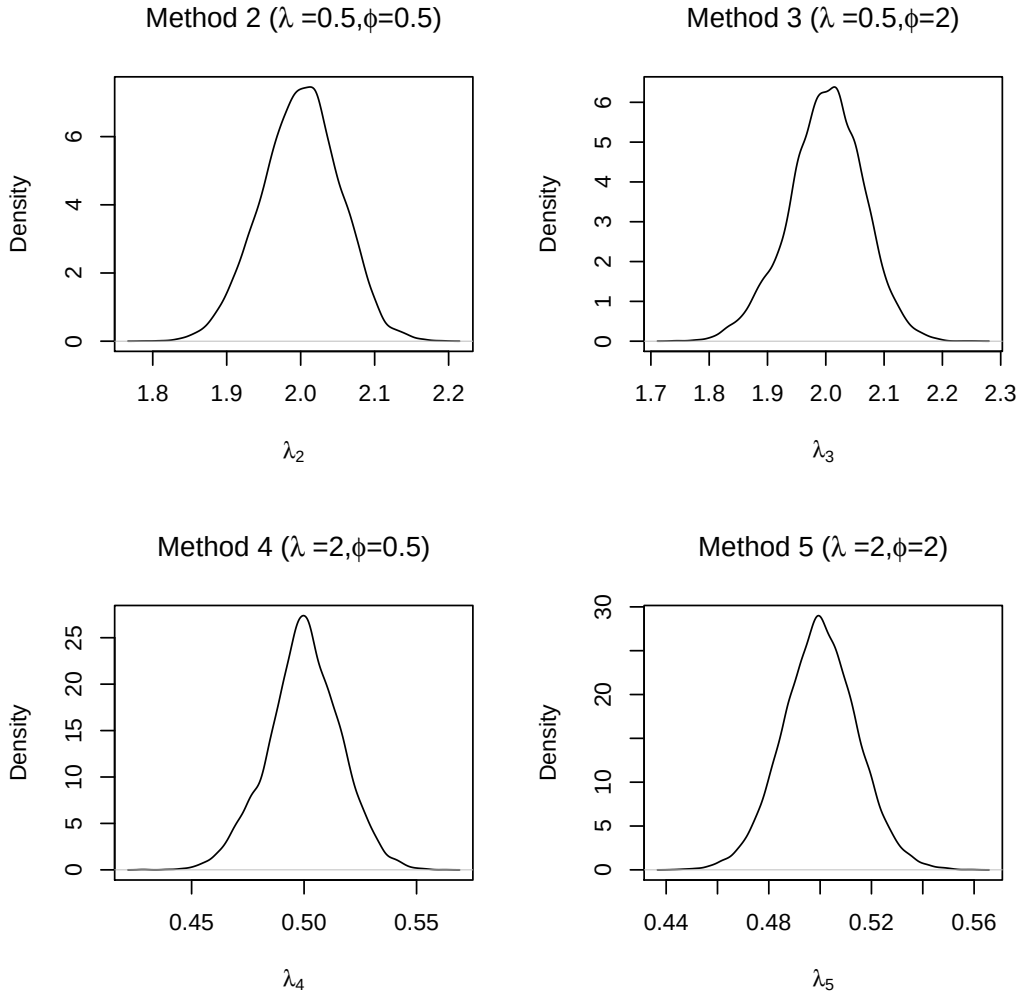

Figure 3: Density plots for λ_j $(j = 1, \ldots, 5)$. The Figure shows how in the simulation study the posterior λ_j distribution for each method is centered around the *true* value suggesting a good performance of the model in estimating the true error terms

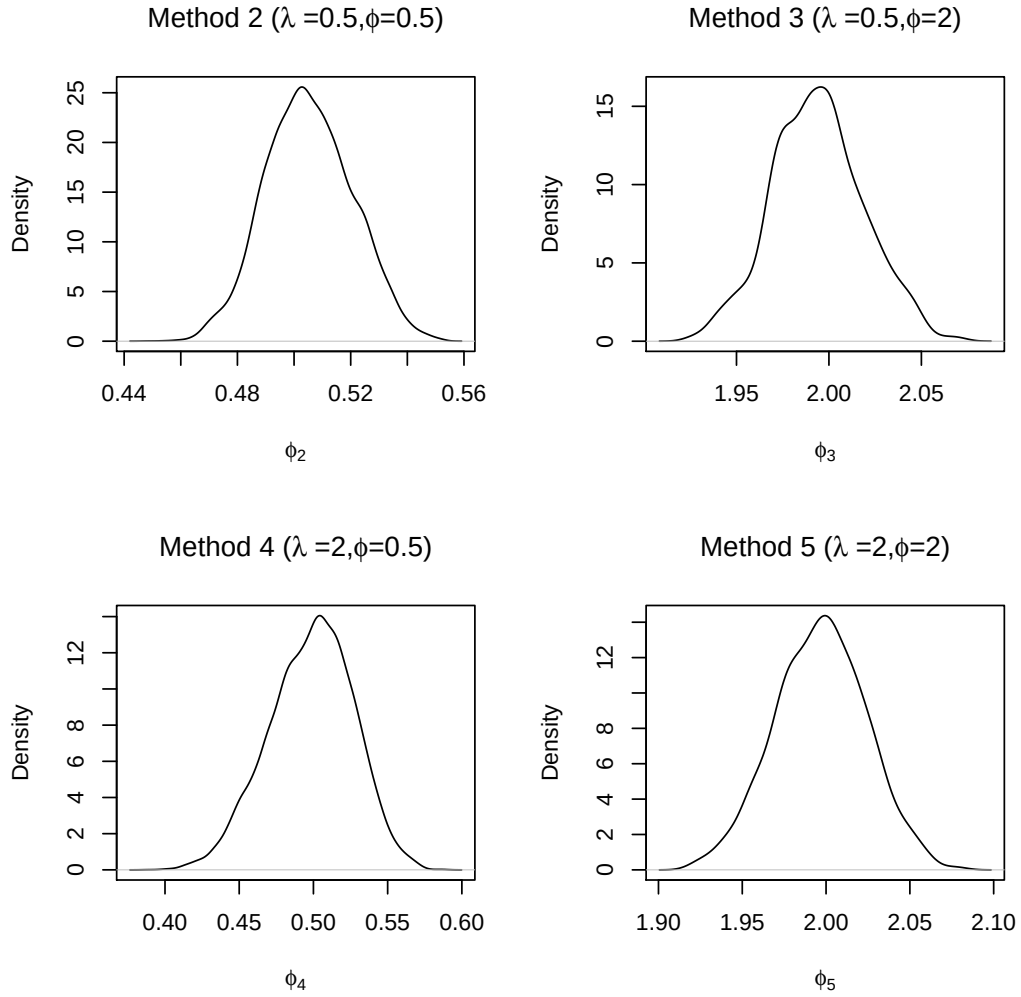

Figure 4: Density plots for ϕ_j $(j = 1, \ldots, 5)$. The Figure shows how in the simulation study the posterior ϕ_j distribution for each method is centered around the *true* value suggesting a good performance of the model in estimating the true relative bias parameters

3 WinBUGS code

```
#Calibration model with the corner point constraint
#G=genes
#J=pre-processing
#K=replicates

model{

    for(g in 1:G){
        for(j in 1:J){
            for(k in 1:K){
            # log gene expression
            y1[g,j,k] ~ dnorm(mu[g,j,1],tau1[g,j])
            y2[g,j,k] ~ dnorm(mu[g,j,2],tau2[g,j])
            #Predicted values under the model
            y1new[g,j,k] ~ dnorm(mu[g,j,1],tau.new1[g,j])
            y2new[g,j,k] ~ dnorm(mu[g,j,2],tau.new2[g,j])

            }
        #ANOVA
        #Condition 1
        mu[g,j,1] <- alpha[g,j]  -  1/2*delta[g]*exp(d[j])
        #Condition 2
        mu[g,j,2] <- alpha[g,j] +  1/2*delta[g]*exp(d[j])


        #VARIANCE gene, method and condition specific
        tau1[g,j] <-exp(lambda1[1,j] + lambda1[2,j]*expr[g] + lambda1[3,j]*pow(expr[g],2))
        *tau.gene1[g]
        tau2[g,j] <-exp(lambda2[1,j] + lambda2[2,j]*expr[g] + lambda2[3,j]*pow(expr[g],2))
        *tau.gene2[g]
        tau.new1[g,j] <-exp(lambda1[1,j] + lambda1[2,j]*expr[g] + lambda1[3,j]*pow(expr[g],2))
        *tau.gene.new1[g]
        tau.new2[g,j] <-exp(lambda2[1,j] + lambda2[2,j]*expr[g] + lambda2[3,j]*pow(expr[g],2))
        *tau.gene.new2[g]

        #Global gene expression for each gene and method
        alpha[g,j] ~ dnorm(0,0.00001)
        }

    #Differential expression for each gene
```

```
    delta[g] ~ dnorm(0,0.0001)

    #Precisions and Variances for each gene and condition
    tau.gene1[g] ~ dgamma(a1,b1)
    sigma.gene1[g] <- 1/tau.gene1[g]

    tau.gene2[g] ~ dgamma(a2,b2)
    sigma.gene2[g] <- 1/tau.gene2[g]

    tau.gene.new1[g] ~ dgamma(a1,b1)
    sigma.gene.new1[g] <- 1/tau.gene.new1[g]

    tau.gene.new2[g] ~ dgamma(a2,b2)
    sigma.gene.new2[g] <- 1/tau.gene.new2[g]
    }

#Corner point constraint for the coefficients of the exponential
component of variability

    for(r in 1:R){
lambda1[r,1]<-0
lambda2[r,1]<-0
        for(j in 2:J){
            lambda1[r,j] ~dnorm(0.01,0.01)
            lambda2[r,j] ~dnorm(0.01,0.01)
            }
        }

#Corner point constraint for the relative bias coefficients

d[1]<-0
    for(j in 2:J){
    d[j] ~ dnorm(0.01,0.01)
    }

#Hyperparameters for the variance

a1~dgamma(0.01,0.01)
b1~dgamma(0.01,bstar1)

a2~dgamma(0.01,0.01)
```

```
b2~dgamma(0.01,bstar2)

bstar1 <- 1/pow(bdistr1,2)
bstar2 <- 1/pow(bdistr2,2)

bdistr1 ~ dunif(0,13.36)
bdistr2 ~ dunif(0,13.80)
}
```

4 Transforming corner point coefficients in sum to 0 coefficients

There are two types of constraints that can be used in the model: (i) corner point, (ii) sum to 0. Both constraints return the same results in terms of differential expression. We have applied the corner point that runs faster and then we have obtained the sum to 0 coefficients that are more easily interpretable. In this paragraph we provide the formula for extracting the "sum to 0" coefficients from the corner point ones. For $j = 1, \cdots, J$ methods we can specify the corner point constraint as follows:

$$\xi_j = \xi_1 + \beta_j \qquad (\xi_1 = 0) \tag{1}$$

Starting from these coefficients we want to apply a transformation to obtain the coefficients that satisfy the sum to 0 constraints:

$$\xi_j = \mu + \alpha_j \qquad (\sum_j \alpha_j = 0) \tag{2}$$

Equalizing equation 1 and 2 when $j = 1$ we obtain:

$$\alpha_1 = -\mu$$

Equalizing equation 1 and 2 when $j > 1$ we obtain:

$$\alpha_j = \alpha_1 + \beta_j$$

Finally μ can be obtained equalizing $\sum_j \xi_j$ for the two constraints:

$$\begin{aligned} \sum_j \xi_j &= 3\mu && \text{From equation 1} \\ \sum_j \xi_j &= \sum_j \beta_j && \text{From equation 2} \\ \mu &= \frac{\sum_j \beta_j}{J} \end{aligned}$$