

# Using Linear Mixed Models or Normalization of cDNA Microarray Data: supplementary material

Philippe Haldermans *et al.* (2007)

March 23, 2007

## 1 AIC values for the analyses presented in Section 3

Table 1 presents the AIC values for all normalization models for the lung dataset, which is presented in Section 3 (Table 1) in the paper.

Table 1: AIC's of the different normalization models for the lung data (K=20)

array	global	linear	non-linear	random intercept	fixed pins	pin by pin	pin by pin with random intercept
1	3852.931	2307.235	1514.088	<b>1425.095</b>	1505.016	1578.928	1519.225
2	3564.334	1471.842	582.395	<b>244.094</b>	550.289	601.536	342.961
3	4421.930	3454.391	2788.243	<b>2229.368</b>	2669.875	2733.868	2230.233
4	4193.063	2732.885	2272.460	<b>2249.596</b>	2293.866	2349.302	2333.702
5	5018.940	3566.565	3039.857	3041.860	<b>3016.017</b>	3095.093	3097.095
6	4340.061	2918.594	2414.714	2416.681	<b>2291.596</b>	2350.773	2352.775
7	4288.504	2491.712	1280.535	<b>1265.816</b>	1275.108	1347.504	1348.358
8	4142.622	2834.162	2512.835	2506.350	<b>2473.571</b>	2493.959	2495.648
9	3762.824	3074.482	2678.986	2649.610	<b>2634.545</b>	2672.588	2666.214
10	2567.354	1898.216	787.869	<b>691.665</b>	790.795	829.566	773.704
11	3299.267	1545.373	721.720	<b>564.925</b>	717.677	756.777	660.078
12	3444.447	2755.059	2125.906	2063.358	1983.775	1998.086	<b>1982.514</b>
13	3613.006	2138.559	1433.739	<b>1309.447</b>	1439.050	1492.187	1419.746
14	4344.090	2827.837	2083.653	2075.049	<b>2070.041</b>	2123.537	2124.442
15	4865.071	3429.412	2736.558	2737.403	<b>2656.092</b>	2684.612	2686.613
16	5274.871	3576.393	2610.936	<b>2576.107</b>	2624.825	2683.351	2664.771
17	3454.021	2705.642	2438.543	2382.498	<b>2281.321</b>	2315.979	2308.909
18	3660.153	2665.168	2203.511	2118.260	<b>2075.605</b>	2143.415	2105.512
19	3962.601	2748.273	2344.326	2245.656	2179.008	2209.678	<b>2169.037</b>
20	4535.124	3201.359	2219.706	<b>2084.985</b>	2061.585	2153.550	2095.765
21	4453.080	2278.234	1651.126	<b>1609.502</b>	1630.206	1703.247	1686.928
22	4308.329	2881.316	2301.866	2204.395	2202.884	2245.123	<b>2187.949</b>
23	3787.089	2469.299	1925.289	1861.074	<b>1792.275</b>	1848.780	1828.561
24	3970.344	2861.207	2690.113	2692.117	2679.507	<b>2660.712</b>	2662.712

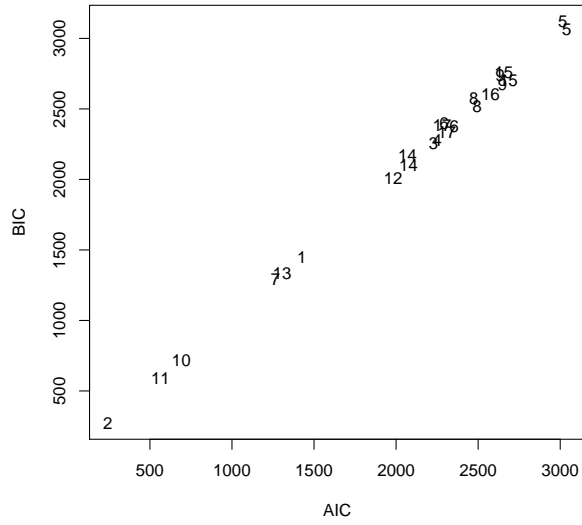
Table 2 presents the AIC values for the apo AI experiment discussed by Yang *et al.*(2002).

Table 2: AIC's of the different normalization models for the second study

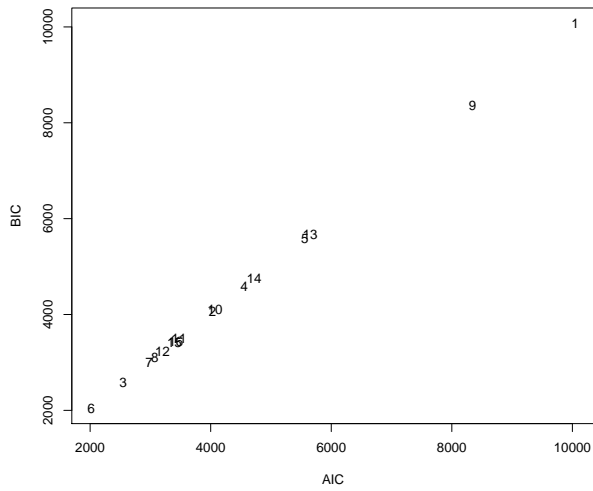
array	global	linear	non-linear	fixed pins	pin by pin
1	13242.475	12527.994	12225.955	10600.132	<b>10042.145</b>
2	8013.272	6336.118	5692.332	4522.263	<b>4020.034</b>
3	9830.152	6654.096	4712.824	3474.075	<b>2549.709</b>
4	8601.862	7505.917	6727.647	5060.814	<b>4547.379</b>
5	10738.048	8899.944	7138.649	6215.575	<b>5560.573</b>
6	9539.005	6920.463	4131.740	2812.745	<b>2004.115</b>
7	8338.348	6714.669	4250.240	3416.357	<b>2976.228</b>
8	7332.593	6443.058	5770.900	4140.125	<b>3071.794</b>
9	10500.368	10011.605	9871.499	8885.492	<b>8343.196</b>
10	6143.106	5498.472	5023.332	4524.790	<b>4070.592</b>
11	6213.324	5660.175	4603.390	3767.839	<b>3473.530</b>
12	6543.331	4853.812	4205.948	3534.885	<b>3203.909</b>
13	7505.036	6929.066	6527.982	5833.490	<b>5650.821</b>
14	6661.483	6198.284	5728.803	4930.171	<b>4723.825</b>
15	5323.179	4969.130	4272.412	3879.359	<b>3398.123</b>
16	5350.521	5261.400	4991.789	3996.540	<b>3424.370</b>

## 2 Model Selection Criteria

Throughout the paper the AIC was used for model selection. Figure 2 presents the AIC and BIC values for lung dataset. For the majority of the arrays AIC and BIC will both select the same model. In the cases were the AIC and BIC select a different model, the values are still comparable. Due to the penalized term, the BIC tends to choose for simpler model, whereas the AIC prefers more complex models.



(a)



(b)

Figure 2: AIC versus BIC values for each array. In case both criteria select the same model, the array number is plotted only once. Otherwise, the best AIC value is plotted against its BIC counterpart, and reversly, the best BIC value and its AIC counterpart. Panel a gives the plot for the lung dataset. Panel b shows the plot for the apo AI experiment.

### 3 The influence of the number of knots on the normalization model

In this section we compare the AIC values obtained for the 24 arrays of the first dataset where the models were fitted with respectively 20, 30 and 40 knots. As can be seen from Tables 1, 3 and 4, the values of the AIC's only have marginal changes. Only in one of the 24 cases, these changes lead to a different model selection. This is mainly due to the fact that the AIC's of two particular models for this array are so close, there is almost no distinction between the two. The choice of the knots  $t_k$  is made in a way that mimics the distribution of the predictor space. Since we only have one predictor, A, a simple solution proposed by Ruppert, Wand and Carroll (2003) is to select equally spaced knots based on the quantiles

$$t_k = \left( \frac{k+1}{K+2} \right) th,$$

which is the sample quantile of the unique age values  $a_i$ , where  $1 \leq k \leq K$ . . Ruppert *et al.* (2003) suggest to choose the number of knots  $K = \min(n/4, 35)$ . Throughout the paper we use  $K=20$  and in this section we present the results for the case  $K=30, 40$ . For all the other cases, the selected model stays the same, illustrating the fact that it is not necessary to use high number of knots, since this will only result in an increase of computational time without any additional benefits.

### 4 The effect of changes in the percentage of upregulated genes

This section reports the results of a simulation study conduct to examine the effect of both the percentage of differentially expressed genes and the amount of differentially expressed genes that are upregulated. For each setting we simulated 100 arrays. In each array we introduced a banana-shaped curve as described in Section 4 of the paper. The goal of the normalization was to

Table 3: AIC's of the different models for the lung data (K=30)

array	global	linear	non-linear	random intercept	fixed pins	pin by pin	pin by pin with random intercept
1	3852.931	2303.235	1514.249	<b>1425.446</b>	1505.262	1579.570	1519.664
2	3564.334	1467.842	581.299	<b>243.829</b>	548.909	604.331	344.175
3	4421.930	3450.391	2788.546	2229.236	2670.124	2732.074	<b>2228.999</b>
4	4193.063	2728.885	2272.688	<b>2249.810</b>	2294.015	2348.312	2332.667
5	5018.940	3562.565	3040.127	3042.127	<b>3016.239</b>	3095.340	3097.340
6	4340.061	2914.594	2415.406	2417.370	<b>2292.392</b>	2353.027	2355.027
7	4288.504	2487.712	1280.634	<b>1265.998</b>	1275.270	1347.820	1348.564
8	4142.622	2830.162	2512.719	2506.250	<b>2473.447</b>	2493.748	2495.428
9	3762.824	3070.482	2678.471	2649.319	<b>2634.044</b>	2671.948	2665.640
10	2567.354	1894.216	787.474	<b>691.427</b>	790.376	827.046	771.301
11	3299.267	1541.373	720.986	<b>565.131</b>	716.901	756.594	660.717
12	3444.447	2751.059	2126.659	2063.693	1984.201	1997.742	<b>1981.911</b>
13	3613.006	2134.559	1433.561	<b>1308.394</b>	1438.016	1488.899	1415.123
14	4344.090	2823.837	2083.872	2075.276	<b>2070.270</b>	2122.670	2123.559
15	4865.071	3425.412	2736.978	2737.842	<b>2656.431</b>	2684.755	2686.755
16	5274.871	3572.393	2610.533	<b>2575.706</b>	2623.601	2683.705	2664.981
17	3454.021	2701.642	2438.880	2383.115	<b>2281.603</b>	2315.912	2308.834
18	3660.153	2661.168	2203.430	2118.087	<b>2075.301</b>	2144.432	2106.048
19	3962.601	2744.273	2344.592	2245.890	2179.317	2210.833	<b>2170.029</b>
20	4535.124	3197.359	2219.323	2084.659	<b>2061.382</b>	2152.636	2095.082
21	4453.080	2274.234	1650.524	<b>1609.360</b>	1628.999	1702.251	1686.021
22	4308.329	2877.316	2301.839	2204.403	2202.955	2244.902	<b>2187.807</b>
23	3787.089	2465.299	1924.246	1860.106	<b>1790.615</b>	1848.063	1827.799
24	3970.344	2857.207	2690.158	2692.158	2679.623	<b>2660.301</b>	2662.3013

Table 4: AIC's of the different models for the lung data (K=40)

array	global	linear	non-linear	random intercept	fixed pins	pin by pin	pin by pin with random intercept
1	3852.931	2303.235	1513.973	<b>1425.009</b>	1504.842	1579.514	1519.541
2	3564.334	1467.842	581.246	<b>244.027</b>	548.851	605.834	344.829
3	4421.930	3450.391	2788.472	2229.251	2670.065	2731.310	<b>2228.218</b>
4	4193.063	2728.885	2272.759	<b>2249.866</b>	2294.035	2347.817	2332.136
5	5018.940	3562.565	3040.268	3042.268	<b>3016.415</b>	3095.388	3097.388
6	4340.061	2914.594	2415.553	2417.518	<b>2292.621</b>	2353.865	2355.865
7	4288.504	2487.712	1280.593	<b>1265.982</b>	1275.158	1348.270	1348.933
8	4142.622	2830.162	2512.679	2506.196	<b>2473.350</b>	2493.640	2495.322
9	3762.824	3070.482	2678.363	2649.198	<b>2633.669</b>	2671.736	2665.490
10	2567.354	1894.216	787.596	<b>691.549</b>	790.638	826.021	770.340
11	3299.267	1541.373	720.625	<b>564.916</b>	716.564	757.024	661.188
12	3444.447	2751.059	2126.931	2063.997	1984.400	1998.174	<b>1982.170</b>
13	3613.006	2134.559	1432.460	<b>1305.036</b>	1436.872	1487.469	1412.660
14	4344.090	2823.837	2083.881	2075.280	<b>2070.295</b>	2121.936	2122.838
15	4865.071	3425.412	2736.938	2737.797	<b>2656.300</b>	2684.868	2686.868
16	5274.871	3572.393	2610.548	<b>2575.723</b>	2623.439	2683.569	2664.8182
17	3454.021	2701.642	2438.874	2383.140	<b>2281.513</b>	2316.012	2308.936
18	3660.153	2661.168	2203.491	2118.141	<b>2075.311</b>	2145.174	2106.490
19	3962.601	2744.273	2344.470	2245.818	2179.165	2211.470	<b>2170.592</b>
20	4535.124	3197.359	2219.492	2084.714	<b>2061.503</b>	2152.455	2094.954
21	4453.080	2274.234	1650.207	<b>1609.054</b>	1628.640	1702.032	1685.799
22	4308.329	2877.316	2302.058	2204.529	2203.133	2245.108	<b>2187.794</b>
23	3787.089	2465.299	1923.669	1859.654	<b>1789.872</b>	1847.309	1827.029
24	3970.344	2857.207	2690.270	2692.270	2679.767	<b>2660.160</b>	2662.160

correctly identify and remove the introduced curve. To quantitate the correctness of the identification we calculate the respective mean square errors between the estimated model and the true function from which the data were generated.

Table 5: MSE for the normalization with LMM and Lowess of an array of 10000 genes with different settings of the percentage of differentially expressed genes (DEG) and the number of upregulated genes.

Percentage of DEG	Method	Number of upregulated genes				
		30%	40%	50%	60%	70%
5%	LMM	0.00205	0.00181	0.00177	0.00179	0.00200
	Lowess40	0.00204	0.00181	0.00175	0.00179	0.00199
	Lowess75	0.00239	0.00217	0.00209	0.00214	0.00234
10%	LMM	0.00695	0.00628	0.00591	0.00694	0.00711
	Lowess40	0.00693	0.00628	0.00588	0.00691	0.00707
	Lowess75	0.00744	0.00678	0.00632	0.00736	0.00753
20%	LMM	0.02309	0.02803	0.02470	0.02312	0.02731
	Lowess40	0.02307	0.02804	0.02468	0.02312	0.02728
	Lowess75	0.02377	0.02888	0.02546	0.02382	0.02800
40%	LMM	0.10138	0.10597	0.11263	0.10258	0.11814
	Lowess40	0.10144	0.10600	0.11271	0.10257	0.11817
	Lowess75	0.10308	0.10743	0.11444	0.10436	0.11965

## 5 Computation time

Figure 3 shows the time needed to normalize one arrays for an increasing number of genes. For each number of genes we simulated 100 arrays and measured the time needed to normalize the array. The elapsed time represented in the figure is the mean of these 100 measures. All simulations were done in R on a Dell Latitude D505 laptop with an Intel Pentium M Processor 1.60 Ghz and 512 Mb of RAM.

## 6 Computational Issues

In this section we discuss the implementation of the models discussed in the paper using the R library `lme()` and the new SAS procedure `GLIMMIX`. All R and SAS code can be downloaded from the website <http://www.censtat.uhasselt.be/software/>.

### 6.1 Normalization using the R library `lme()`

The global normalization model can be implemented by fitted a regression model without covariates, `lmm.global <- lme(M ~ 1)`.

Similarly, the linear normalization model can be implemented by adding the design matrix for the fixed effects discussed in Section 2.2.1 of the paper.

```
lmm.linear <- lme(M ~ A)
```

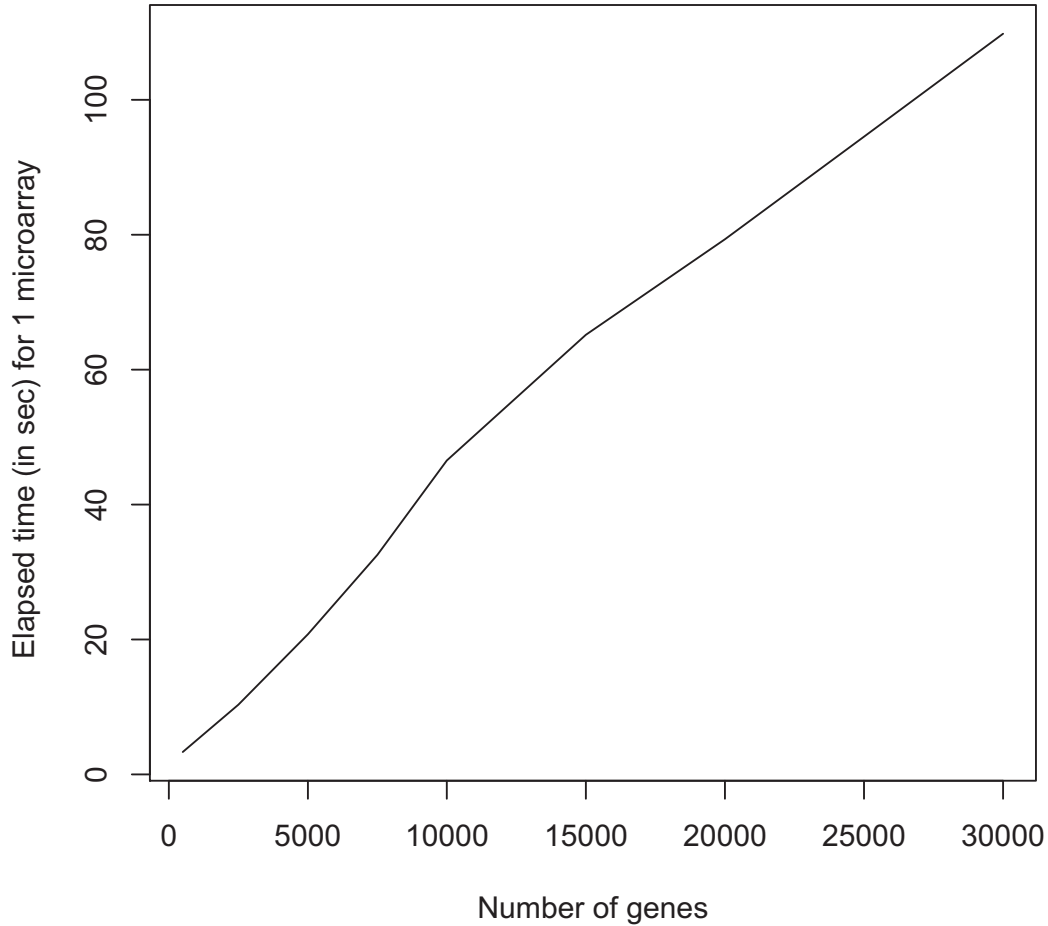


Figure 3: Elapsed time in seconds needed to normalize one array for an increasing number of genes.

The nonlinear normalization model requires introduction of the random effect vector for the smoother.

```
lmm.nonlin <- lme(M ~ A, random = pdIdent (~zmat1))
```

The statement `random = pdIdent(~zmat1)` defines a diagonal covariance matrix for the random effects, while `zmat` is the design matrix for the random effects discussed in Section 2.2.2. For the pin by pin normalization we replace the random statement with `random=list(printTip=pdIdent(~zmat1 -1))`. The object `printTip` defines the different pins for normalization. The complete code for the model is given by

```
lmm.pin<-lme(M~A,random=list(printTip=pdIdent(~zmat1-1))).
```

The random intercept model, which takes into account replicated genes in the array is a multilevel model with two sets of random effects as discussed in Section 2.2.4. For this type of model the random statement is given by `random=list(all=pdIdent(~zmat1-1), geneID= pdSymm(~1))`, where `all` defines a vector of ones and the object `geneID` the different replicates of a gene. The complete code is given by

```
lmm.randInt<-lme(M ~ A,random=list(all=pdIdent(~zmat1-1),  
geneID=pdSymm(~1))).
```

The model which allows for pin by pin normalization with random intercept can be implemented using

```
lmm.randIntPin <-lme(M~A,random=list(printTip=pdIdent  
(~zmat1-1), geneID=pdSymm(~1))).
```

The construction of the design matrix for the random effects (`zmat1`) follows the matrix described in (4), depending only on the number of knots. In the examples presented in the paper we use a value of 20 knots, while the results obtained using 30 and 40 knots were presented in Section 3.

## 6.2 Normalization using the SAS procedure GLIMMIX

The normalization models discussed in this paper can be implemented in SAS using the new procedure GLIMMIX. The statement `type=rsmooth` specifies a radial smoother based on the smoother of Ruppert *et al.* (2003). The option `knotmethod=kdtree` determines the method of construction knots for the smoother, whereas `bucket=20` indicates the number of knots to be used by the smoother. For the nonlinear normalization the code is given by

```
proc glimmix data = theData;  
class geneID;  
model M=A;  
random A /type=rsmooth  
knotmethod=kdtree(bucket=20);  
output out=out pred=p resid=r;  
run;
```

The normalization model with random intercept can be fitted by an extra `random intercept` statement with `subject=geneID` indicating that a random intercept is to be fitted for each gene. This is implemented with the following code:

```

proc glimmix data = theData;
class geneID;
model M=A;
random A /type=rsmooth
          knotmethod=kdtree(bucket=20);
random intercept / subject=geneID;
output out=out pred=p resid=r;
run;

```

Finally, for the print-tip specific random effect normalization we use the SAS procedure MIXED, where the Z-matrix is explicitly defined as zlist. Here  $Z_1, \dots, Z_{20}$  are the columns of the design matrix Z. The procedure performs a separate analysis for each printtip, specified by the **subject=printTip** statement. This resulted in the next program:

```

%macro model2(xlist=,zlist=);
proc mixed data=theData;
class printTip;
model M=&xlist/outp=out;
random &zlist/type=toep(1) subject=printTip;
run;
%mend;

```

```

%model2(xlist=A, zlist= Z1 Z2 Z3 Z4 Z5 Z6
Z7 Z8 Z9 Z10 Z11 Z12 Z13 Z14 Z15 Z16 Z17 Z18
Z19 Z20)

```

Both programming languages yield the same results. However some models are computed in R within seconds, while it may take SAS several minutes to perform the same tasks.

## 7 Data structure

The data needed as input of the previously specified procedures primarily consists out of two data vectors containing the red (R) and the green (G) values on a  $\log_2$  scale. These are then used to calculate the M and the A values, which form the basis of our model statements. Additionally to this, it is possible to specify several characteristics of the microarray. For instance,

one can indicate whether or not there is replicated data in the microarray with a vector containing an ID of each gene. Another possibility is the addition of information about the print-tips. This can be done in two ways. A first option is to give the relative position of the spot through a row and a column indicator, which is then internally converted to the number of the printtip. It is also possible to simply give the list of which printtip was used on each spot. Finally it is also possible to enter multiple arrays at once. Therefore an additional vector is needed which indicates to which array a certain spot belongs. Internally the arrays are splitted up and processed sequentially. An example of a small portion of possible code is given in Table 6.

Table 6: Portion of example data for the R/SAS procedures

$\log_2(R)$	$\log_2(G)$	geneID	row	column	array
14.76664895	14.65674395	228	1	1	1
12.74530539	12.99185661	504	1	2	1
9.097227436	7.923131237	328	1	3	1
9.993095638	9.305829536	38	1	4	1
...	...	...	...	...	...