The problems encountered during microarray data analysis

Joanna Zyprych

Department of Mathematical and Statistical Methods, Poznań University of Life Sciences,
Wojska Polskiego 28, 60-637 Poznań, e-mail: zjoanna@au.poznan.pl

The data are from the microarray experiment. This project was about Acute myeloid leukemia - investigation of transcriptom of cell from the blood and bone marrow. 86 hybridizations were made. On one microarray there was combined experimental probe - RNA sample from a patient or a healthy person, always labelled with Cy5 and control probe - RNA isolated from cell line HL60 (a subtype of AML) labelled with Cy3. Hybridization was made as follows:
1-2 HL60 versus Control
3-68 HL60 versus Leukemia
69-86 HL60 versus Control
From each hybridization using image analysis software genepix we get gpr file. The data from this file are:

Block	Column	Row	Name	ID	X	Y	Dia.	F647 Median	F647 Mean	F647 SD	F647 CV	Flags
1	1	1	ERG_Operon	2078	1220	1450	270	2115	2292	1194	52	100
1	2	1	ERG_Operon	2078	1670	1450	270	2241	2383	1123	47	100
1	3	1	ERG_Operon	2078	2120	1450	260	2164	2273	945	41	100
1	4	1	FLT3_Operon	2322	2570	1450	200	201	259	227	87	-50
1	5	1	FLT3_Operon	2322	3020	1450	200	167	213	149	69	-50
1	6	1	FLT3_Operon	2322	3480	1450	200	184	306	384	125	-50
1	7	1	GAPDHS_Operon	26330	3930	1450	200	237	339	379	111	-50
1	8	1	GAPDHS_Operon	26330	4380	1460	200	232	358	417	116	-50
1	9	1	GAPDHS_Operon	26330	4840	1460	200	261	470	535	113	-50

Figure 1. Gpr file from GenePix for AML experiment.

The last column gives us the specified knowledge which weight should be given to spots. For flags value less than the cutoff value we give weights equal 0 and 1 otherwise. We choose cutoff=-50 to downweight bad or absent spots. As we can see from this data first we should calculate the mean intensity for each gene (here we have 3 replication of each spot). Taking into consideration that the spot has a weight of zero or one we want to determine the average of our repetitions in such way that these spots which have weight zero are not taken into account in our calculations.

We build following code:

```
> # MA.A are the data after preprocessing (dim(MA.A)=3069x137)
> Mean_intesity <- avedups(MA.A, ndups=3, spacing=1, weights=MA.A$weights)
```

To check if the function works right, we can draw two MA plots: before and after using avedups function.

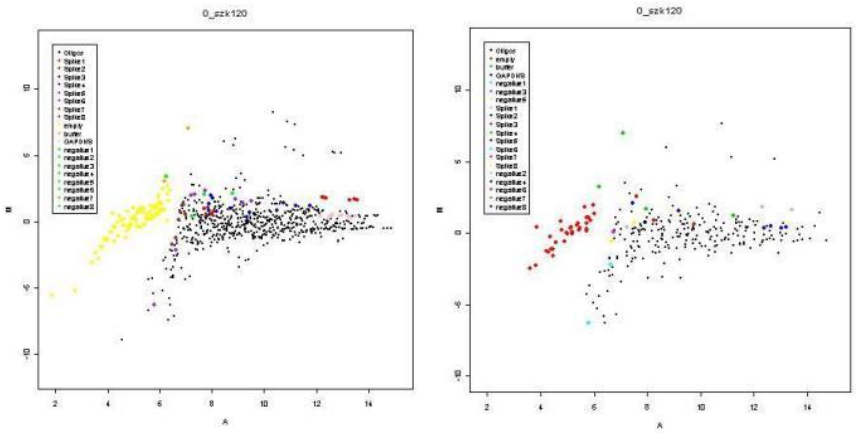

Figure 2. MA plots for one slide before and after using avedups function.

After image analysis and normalization we would like to answer some questions. One of them is: which genes are over(under)expressed comparing leukemia and control probe? Using R software we found the genes which are statistically significant. The statistics used for these calculation are: two sample t-statistics, sam-statistics and fc-statistics. All these calculations were performed using Deds package. Venn diagram shows which genes are common to the three different methods of analysis.

```
> library(DEDS)
> # from targets file 0-control, 1-leukemia
> L<-rep(c(0,1,0),c(2,66,18))
> data<-as.matrix(Mean_intesity)
> d <- deds.stat.linkC(data, L, B=200)
> # for the comparisons between the 3 statistics
> t_genes<-topgenes(d, number = 50,Mean_intesity$genes$Name,sort.by="t" )
> fc_genes<-topgenes(d, number = 50,Mean_intesity$genes$Name,sort.by="fc" )
> sam_genes<-topgenes(d, number = 50,Mean_intesity$genes$Name,sort.by="sam" )
```

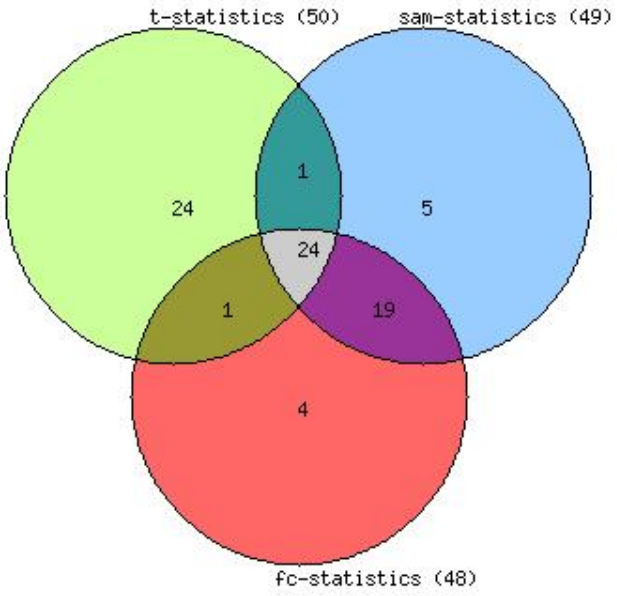

Figure 3. Venn diagram for comaparison of 3 different statistics from DEDS package.

Short description for this function:
deds.stat.linkC(X,L,B=1000,tests=c("t", "fc", "sam","..."))
X: A matrix, in the case of gene expression data, rows correspond to N genes and columns to p mRNA samples.
L: A vector of integers corresponding to observation (column) class labels.
B: The number of permutations.

Another example of the microarray analysis. From each hybridization using image analysis software genepix we get gpr file. The data from this file are:

Block	Column	Row	Name	ID
1	1	1	Dye Marker	97: D-01 Dye Marker
1	2	1	H200000001-NM_001885	01-D01-H200000498-ENSG00000109846
1	3	1	Buffer	96: D-01 Buffer
1	4	1	H200000511-NM_030984;NM_001061	01-D13-H200000511-ENSG00000059377
1	5	1	H200000542-NM_005658	01-H01-H200000542-ENSG00000056558
1	6	1	H200000008-NM_005041	01-H13-H200000557-ENSG00000180644
1	7	1	H200000577-NM_000073	01-L01-H200000577-ENSG00000160654
1	8	1	H200000583-NM_003385	01-L13-H200000583-ENSG00000163032
1	9	1	H200000011-NM_006080	01-P01-H200000613-ENSG00000075213

Figure 4. GPR file.

The data records contain all names and identifier information for each spot. As we can see from the Name column it has only oligo id, here we can not see which sond consist which gene. We have also available a separate gal file in which oligo id and gene symbol are combined. These data look as follows:

384_number	384_position	oligo_id	oligo_sequenc	gene_id	transcript_id	gene_symbol
1	A03	H200000001	TGGGGAGAA	ENSG0000015	ENST0000028	NAT2
1	A05	H200000005	GAAGGCTCT	ENSG0000009	ENST0000020	TGM1
1	A07	H200000006	ATGGGTTACA	ENSG0000006	ENST0000038	FECH
1	A09	H200000007	TATGGAGAT	ENSG0000017	ENST0000038	GLDC
1	A11	H200000008	GTCATCTTCT	ENSG0000014	ENST0000027	MS4A2
1	A13	H200000010	CATGGAGGA	ENSG0000017	ENST0000038	Q6FG55_HUMAN
1	A15	H200000011	GAACAGGAC	ENSG0000007	ENST0000026	ACAT1
1	A17	H200000014	GTGCTGTGG	ENSG0000016	ENST0000037	PTAFR

Figure 5. GAL file.

We are working in R with gpr files and that is why as result we obtain only the names of sond. We need for further analysis gene names and this is the reason why we combine these two files gpr and gal file. Grep searches for matches to pattern (its first argument) within the character vector x (second argument).

R code

```
> gal<-read.table("gal.csv",dec=",", sep=";")
> wyniki<-read.table("gpr.csv",dec=",", sep=";")
> gal<-gal[,3]
> gal<-as.character(gal)
> gpr<-gpr[,4]
> gpr<-as.character(gpr)
> symbol<-gal[,9]
> symbol<-as.character(symbol)
> result<-matrix(0,length(gpr),2)
> result[,1]<-gpr
> colnames(result)<-c("Sonda","Gen_symbol")
> # i is the number of elements in a
> for(i in 1:8){
+        j<-grep(gal[i],gpr)
+        result[j,2]<-symbol[i]
+                  }
```