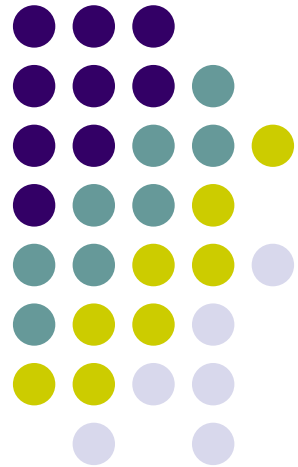
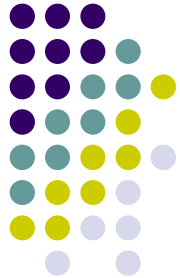
Introduction to Bioconductor

2. Statistical analysis using Bioconductor

Bioinformatics and Biostatistics Lab.,
Seoul National Univ. Seoul, Korea
Eun-Kyung Lee

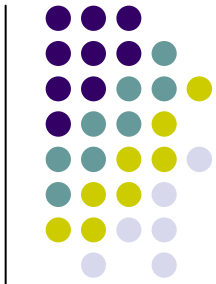


Outline



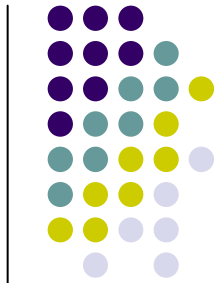
- preprocessing (cDNA, Affy)
  - Normalization
  - Summarization
- Identify significantly different genes(limma, sam)
- classification (tree, randomforest)
- clustering (som)

Normalization



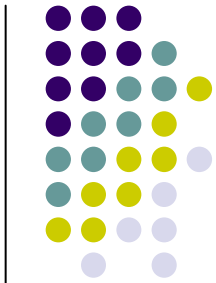
- What is Normalization?
  - How do we compare results across chips?
  - Getting intensity values from one chip to mean the same as intensity values from another chip.
- Why is Normalization an issue?
  - Amount of RNA
  - DNA quality
  - Variation is obscuring as opposed to interesting

Normalization Methods



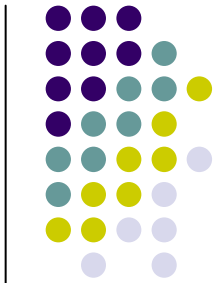
- Old fashioned method
  - Use housekeeping genes : start with a set of genes whose expression shouldn't change
  - Use Spike-ins : Use a set of markers whose relative intensities you can control Cyclic Loess
  - Simple scaling
- Commonly used method
  - Quantiles
  - Cyclic Loess

Normalization Methods



- Quantiles
  - Assume that the distribution of probe intensities should be completely the same across chips
  - Start with n arrays and p probes ; form p*n matrix X
  - Sort the columns of data matrix X so that the entries in a given row correspond to a fixed quantile
  - Replace all entries in that row with their mean
  - Undo sort
  - Sorting and averaging are comparatively fast
  - Projecting the observed n-vector onto this central axis suggests using the mean value

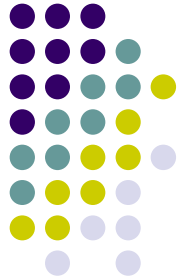
Normalization Methods



- Cyclic Loess
  - Start with MA plots
  - Fit a loess smooth for each pair of chips
  - Let $M_k = \log 2(x_{ki} / x_{kj})$ for arrays i and j.
  - Let $\hat{M}_k$ be the fitted loess curve.
  - Then, the adjusted value is $M_k^{'} = M_k - \hat{M}_k$
  - Repeat for all pairs, the refit and repeat.
  - This is very slow.

(Bolstad et al, Bioinformatics 2003)

Summary measure



- avgdiff
- liwong

$$MM_{ij} = \nu_j + \theta_i \alpha_i + \varepsilon$$

$$PM_{ij} = \nu_j + \theta_i \alpha_i + \theta_i \phi_j + \varepsilon$$

- mas

$$signal = TukeyBiweight(\log(PM_j - CT_j))$$

- medianpolish

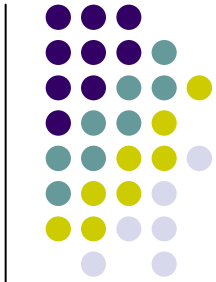
$$\log(PM_{ij} - BG) = \mu_i + \alpha_i + \sigma\varepsilon_{ij}$$

Example 1

- Arabidopsis data
- For each of 22810 genes we have

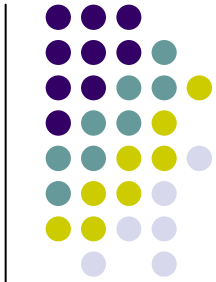
	Replicates
Mutant : IMW	IMW1, IMW2, IMW3
Mutant : NF	NF1, NF3
Wild Type	WT1, WT2, WT3

Read Affymetrix data



```
> library(affy)
Loading required package: Biobase
Loading required package: tools

Welcome to Bioconductor

    Vignettes contain introductory material. To view, type
    'openVignette()' or start with 'help(Biobase)'. For details
    on reading vignettes, see the openVignette help page.

Loading required package: affyio
> cel.path<-"d:/ISM-data/affy"
> celfile.name<-
   list.celfiles(path=cel.path,full.names=TRUE)
```

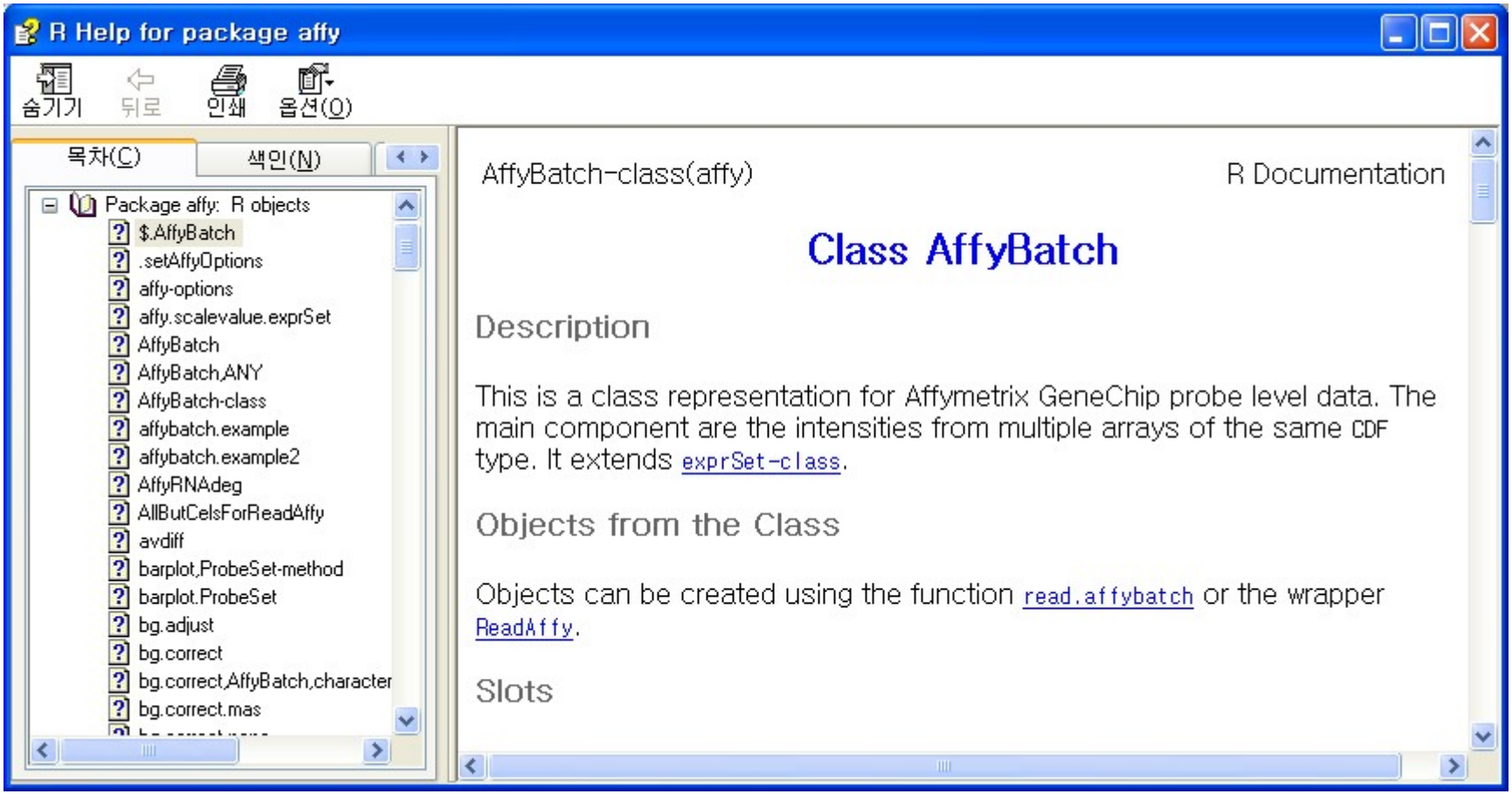
Read Affymetrix data



```
> celfile.name
[1] "d:/ISM-data/affy/IMW1.CEL" "d:/ISM-data/affy/IMW2.CEL"
[3] "d:/ISM-data/affy/IMW3.CEL" "d:/ISM-data/affy/NF1.CEL"
[5] "d:/ISM-data/affy/NF3.CEL"  "d:/ISM-data/affy/WT1.CEL"
[7] "d:/ISM-data/affy/WT2.CEL"  "d:/ISM-data/affy/WT3.CEL"
> affy.testdata<-ReadAffy(filenames=celfile.name)
> class(affy.testdata)
[1] "AffyBatch"
attr(,"package")
[1] "affy"
> slot(affy.testdata,"cdfName")
[1] "ATH1-121501"
> sampleNames(affy.testdata)
[1] "IMW1.CEL" "IMW2.CEL" "IMW3.CEL" "NF1.CEL"  "NF3.CEL"  "WT1.CEL"  "WT2.CEL"
[8] "WT3.CEL"
> geneNames(affy.testdata)[1:5]
[1] "244901_at" "244902_at" "244903_at" "244904_at" "244905_at"
```

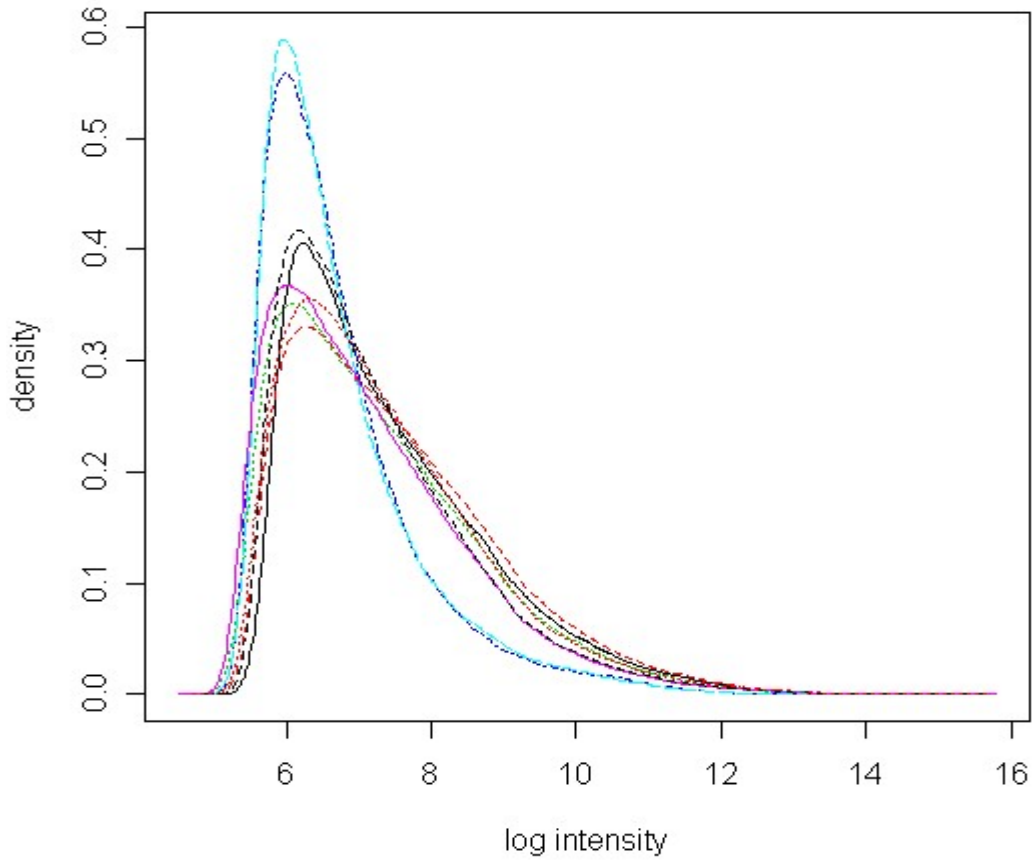
Read Affymetrix data

```
>  class ? AffyBatch
```



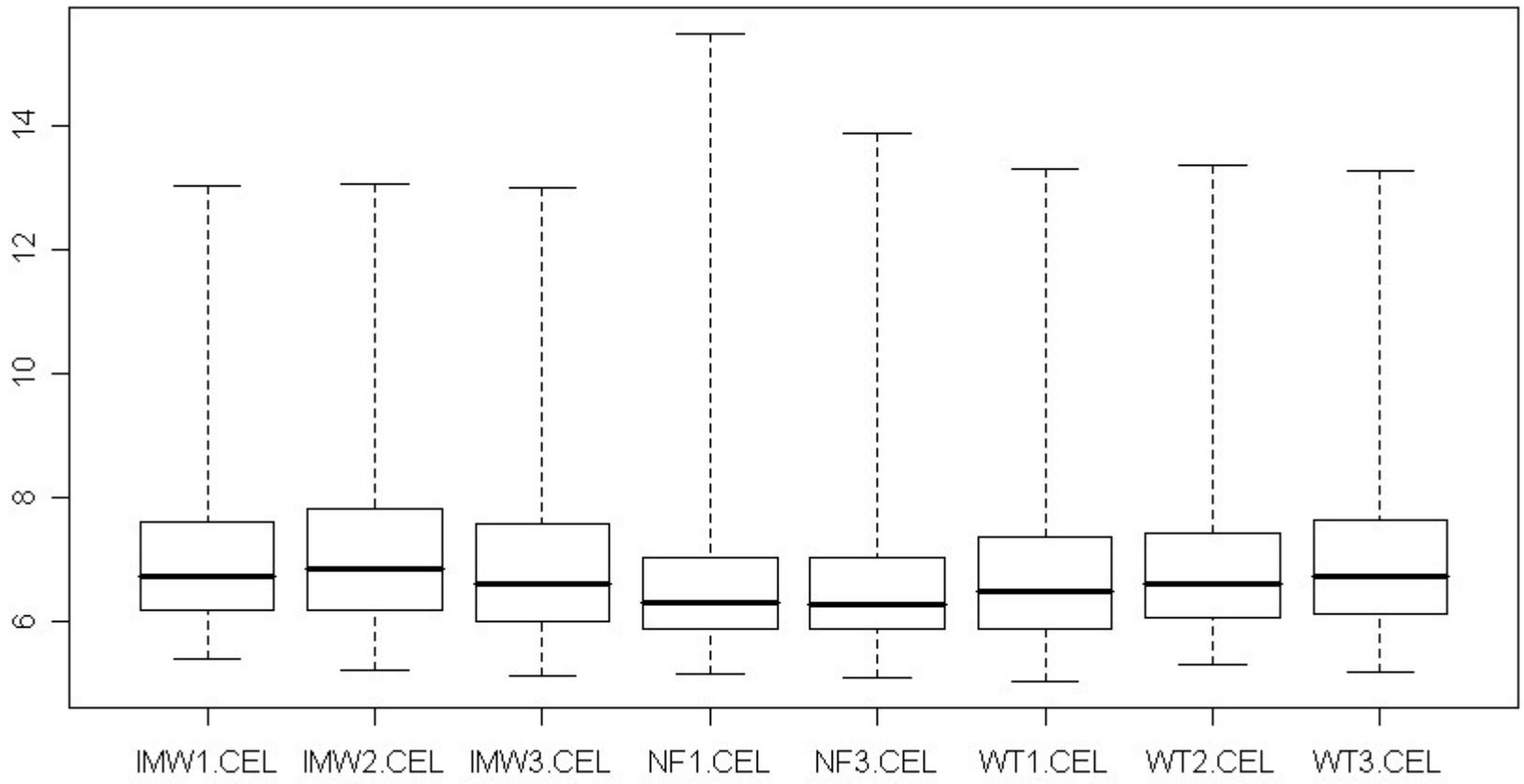
Read Affymetrix data

```
> hist(affy.testdata)
```

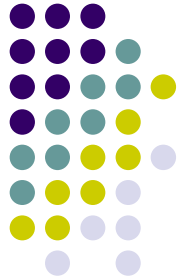


Read Affymetrix data

```
> boxplot(affy.testdata)
```



Examining probe-level data



```
> pm(affy.testdata)[1:5,]
```

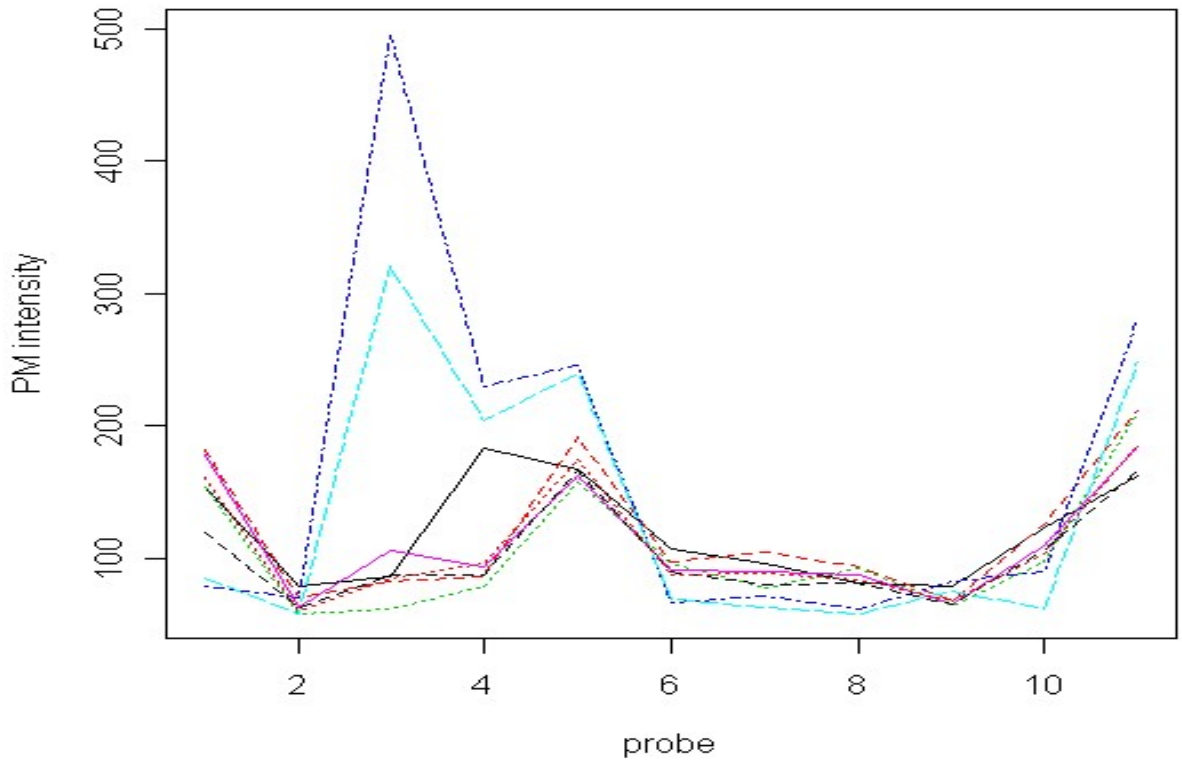
	IMW1.CEL	IMW2.CEL	IMW3.CEL	NF1.CEL	NF3.CEL	WT1.CEL	WT2.CEL	WT3.CEL
[1,]	153.8	182.0	153.3	79.3	84.5	177.8	119.8	161.0
[2,]	79.0	70.5	58.5	70.5	58.0	63.3	63.0	60.8
[3,]	85.8	83.0	61.8	496.3	320.8	106.0	86.8	84.5
[4,]	182.5	86.5	79.3	229.3	204.0	93.5	87.3	95.8
[5,]	167.5	191.5	157.3	245.8	239.5	162.5	166.3	174.3

```
> mm(affy.testdata)[1:5,]
```

	IMW1.CEL	IMW2.CEL	IMW3.CEL	NF1.CEL	NF3.CEL	WT1.CEL	WT2.CEL	WT3.CEL
[1,]	65.5	65.3	62.3	60.0	51.8	51.5	60.0	63.0
[2,]	65.8	66.8	59.8	53.0	72.5	49.8	64.3	63.0
[3,]	82.3	76.0	58.3	583.8	424.0	85.0	83.5	77.8
[4,]	117.3	65.8	57.3	137.0	122.0	63.5	81.3	84.3
[5,]	80.0	76.3	70.0	52.8	64.0	63.3	61.0	70.8

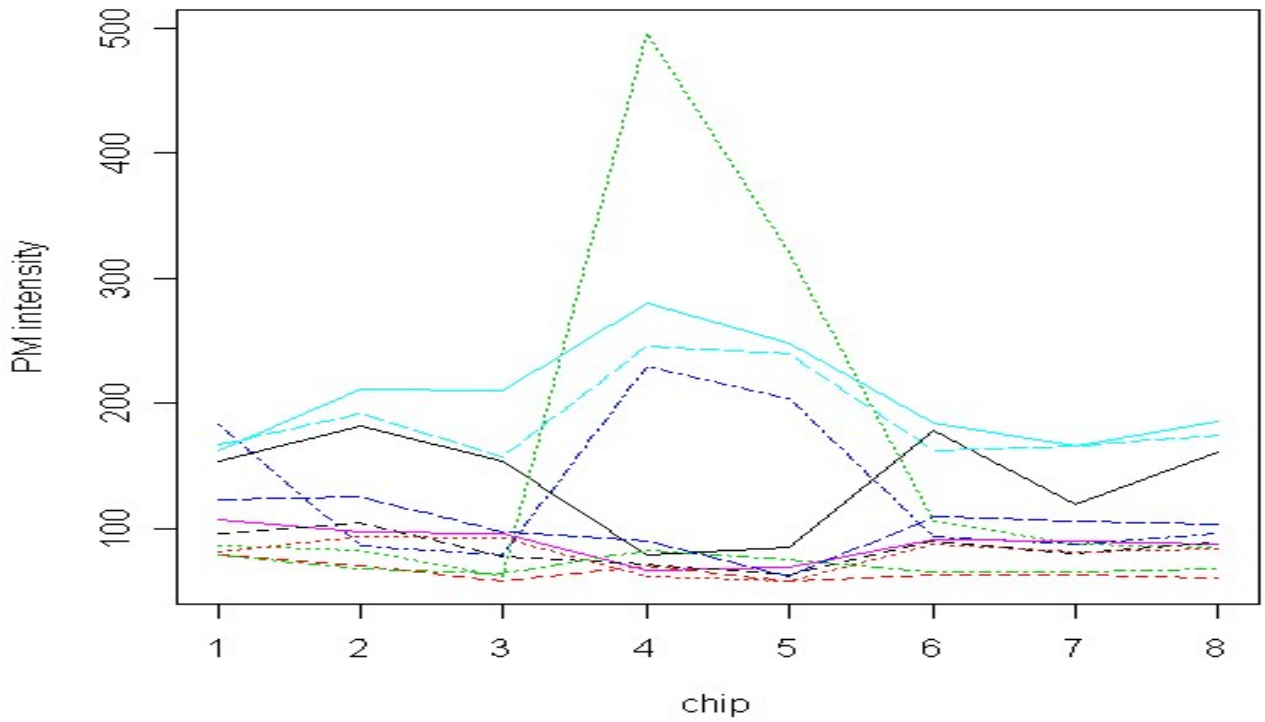
Examining probe-level data

```
>matplot(pm(affy.testdata,"244901_at"),type=
   'l',xlab="probe",ylab="PM intensity")
```

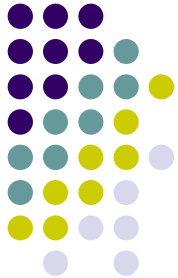


Examining probe-level data

```
>matplot(t(pm(affy.testdata,"244901_at")),ty
  pe='l',xlab="chip",ylab="PM intensity")
```



phenotype data

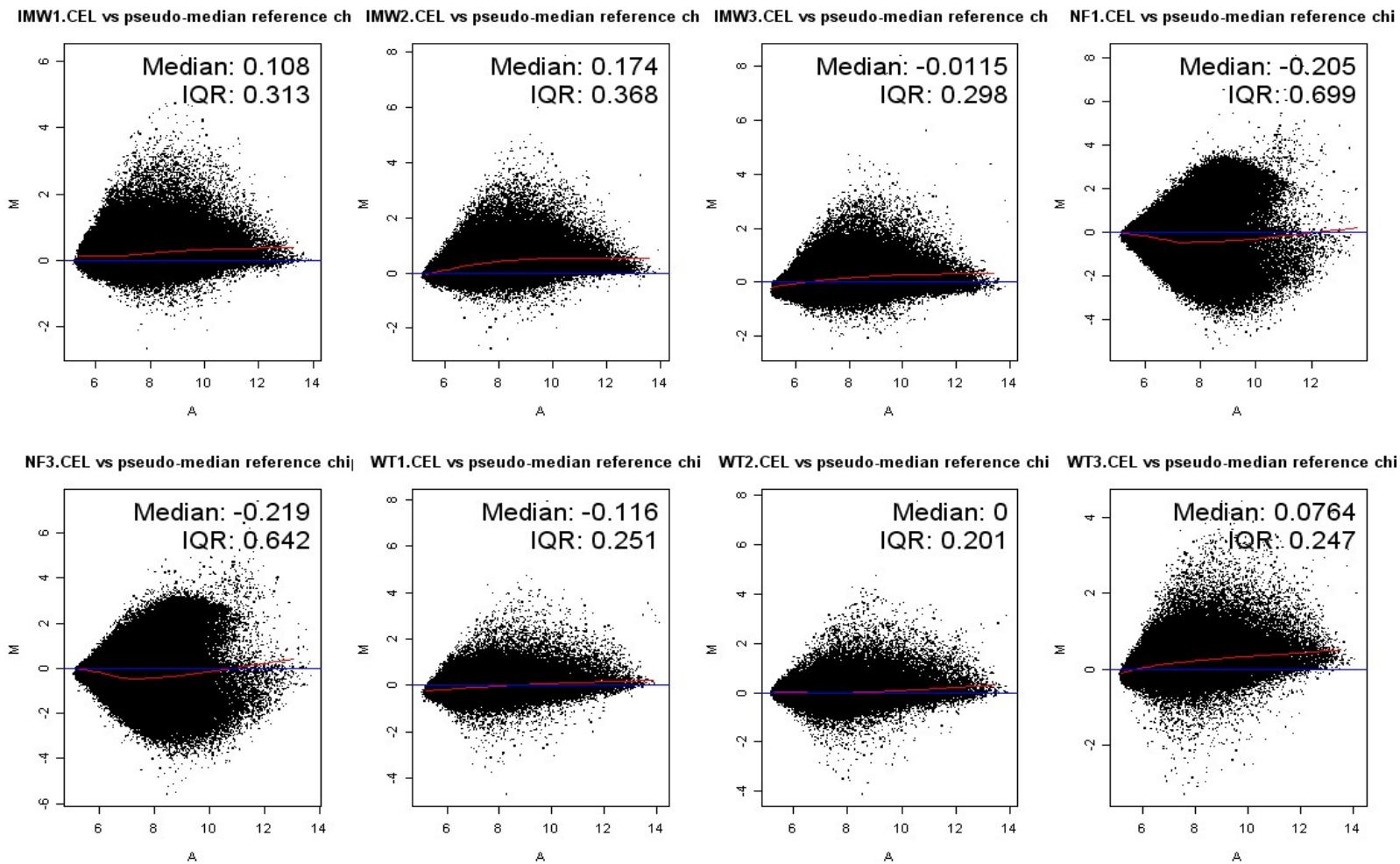


```
> pheno<-data.frame
  (genotype=c("IMW","IMW","IMW","NF","NF","WT","WT","WT
  "),replicate=c(1,2,3,1,2,1,2,3))
> pData(affy.testdata)<-
  cbind(pData(affy.testdata),pheno)
> pData(affy.testdata)
```

	sample	genotype	replicate
IMW1.CEL	1	IMW	1
IMW2.CEL	2	IMW	2
IMW3.CEL	3	IMW	3
NF1.CEL	4	NF	1
NF3.CEL	5	NF	2
WT1.CEL	6	WT	1
WT2.CEL	7	WT	2
WT3.CEL	8	WT	3

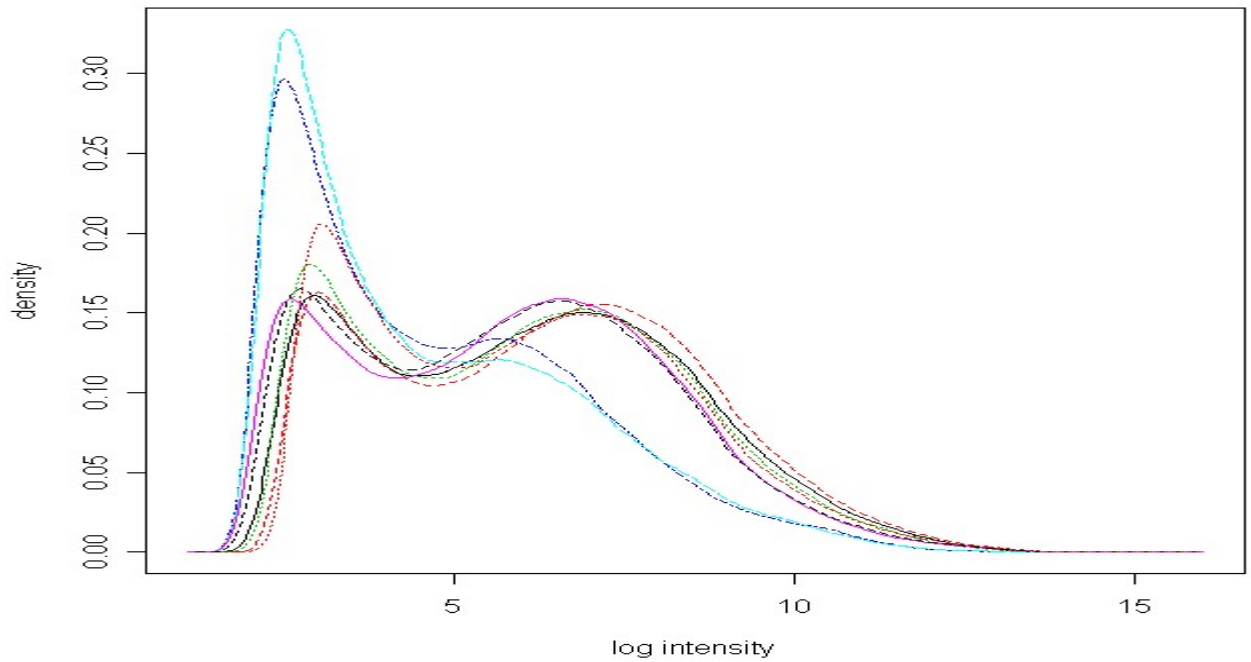
MvA plot

> par(mfrow=c(2,4)); MAplot(affy.testdata)



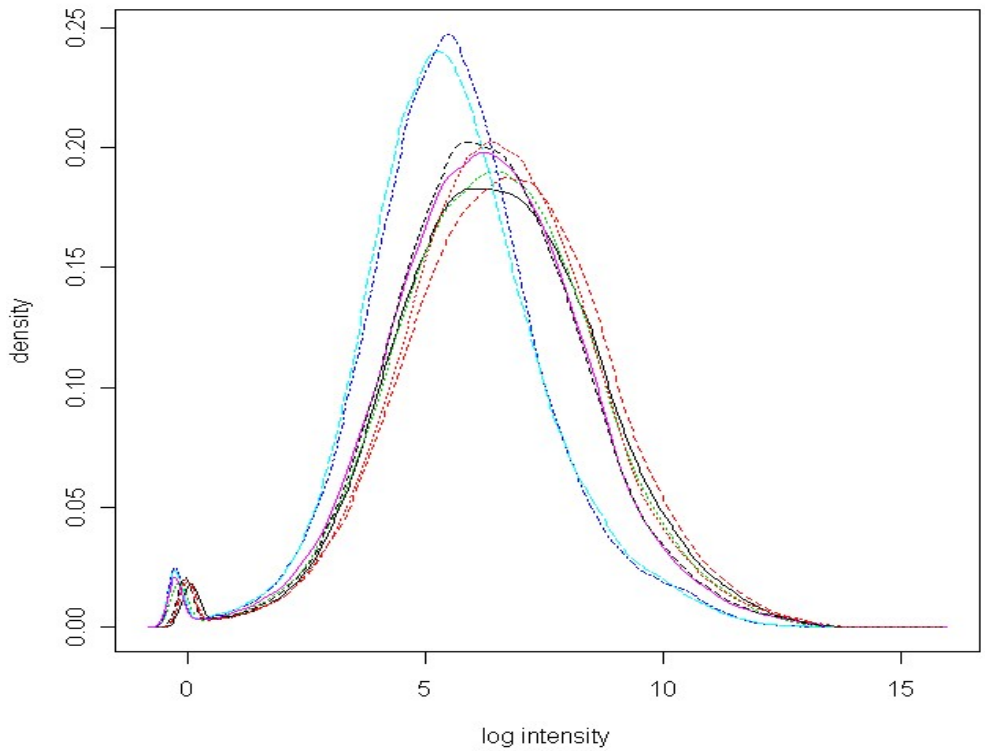
background adjustment

```
> bgcorrect.methods
[1] "mas"  "none" "rma"  "rma2"
> affytest.bg.rma<-bg.correct(affy.testdata,
   method="rma"); hist(affytest.bg.rma)
```



background adjustment

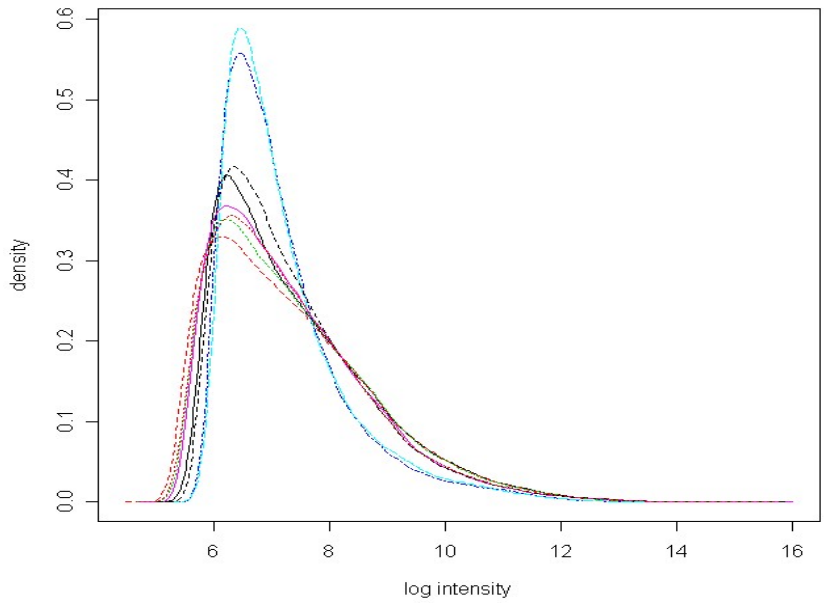
```
> affytest.bg.mas<-bg.correct(affy.testdata,
   method="mas"); hist(affytest.bg.mas)
```



normalization

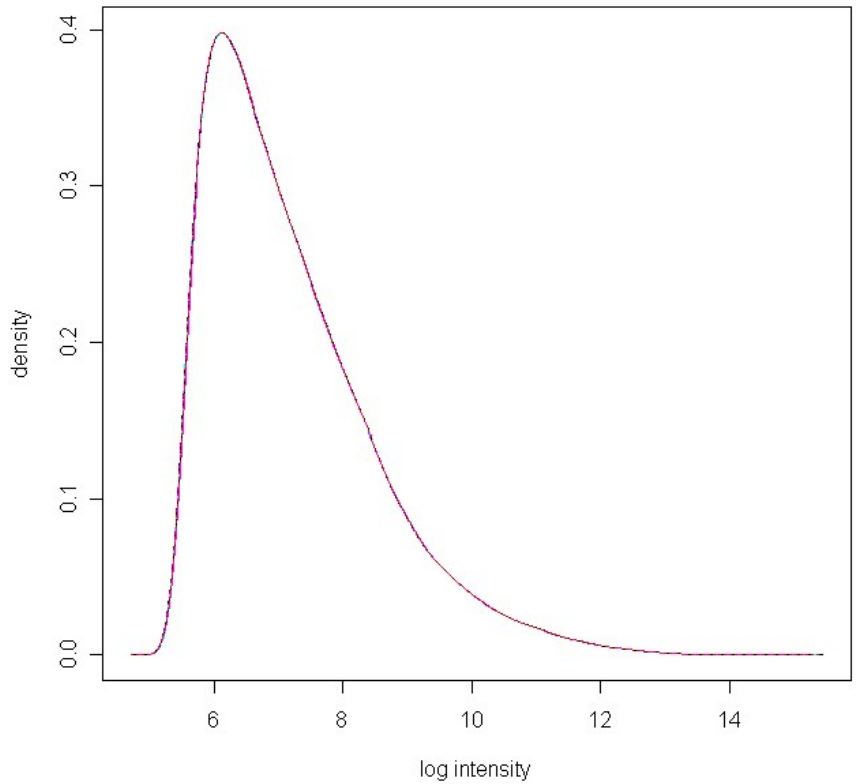
```
> normalize.methods(affy.testdata)
[1] "constant"  "contrasts"        "invariantset"     "loess"
[5] "qspline"   "quantiles"        "quantiles.robust"
```

```
> affytest.norm.constant<-normalize(affy.testdata,
   method="constant"); hist(affytest.norm.constant)
```



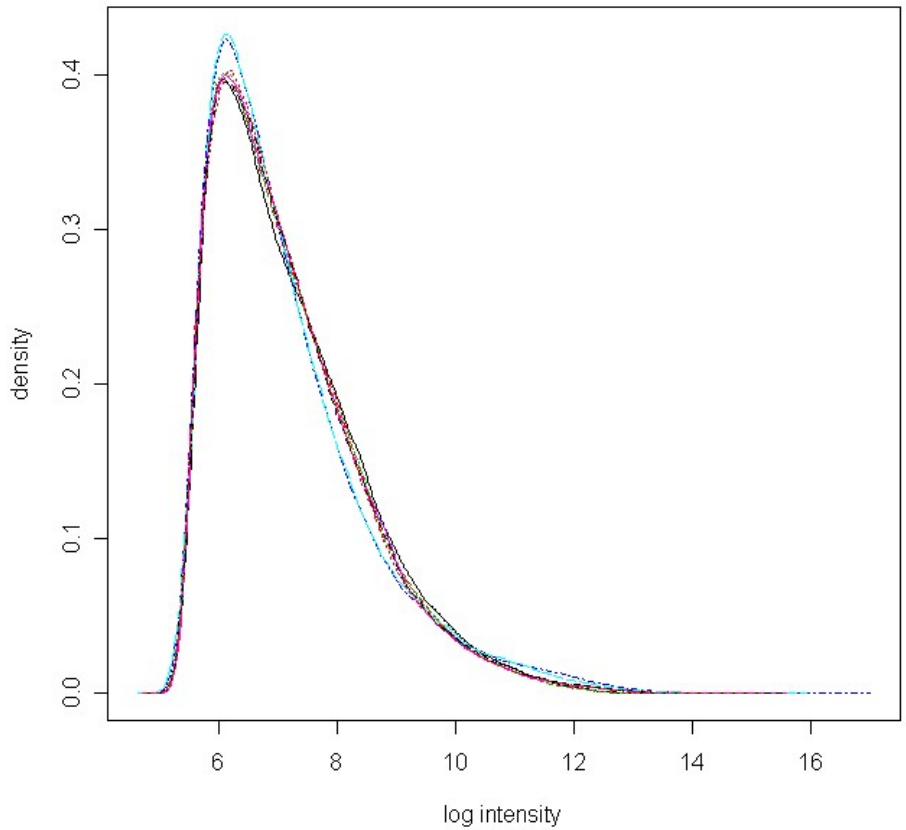
normalization

```
> affytest.norm.quantile<-normalize(affy.testdata,
  method="quantiles"); hist(affytest.norm.constant)
```



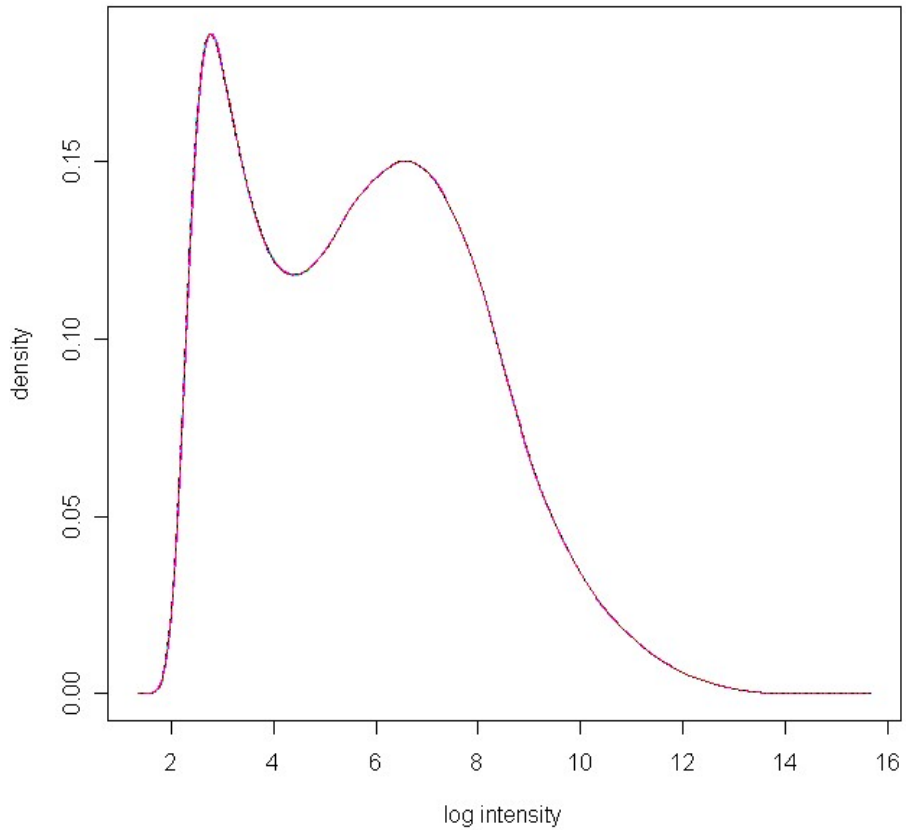
normalization

```
> affytest.norm.loess<-normalize(affy.testdata,
  method="loess"); hist(affytest.norm.loess)
```

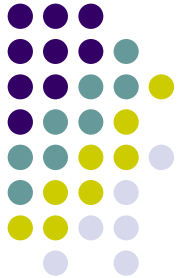


normalization

```
> affytest.bg.norm.quantile<-normalize(affytest.bg.rma,
  method="quantiles");hist(affytest.bg.norm.quantile)
```

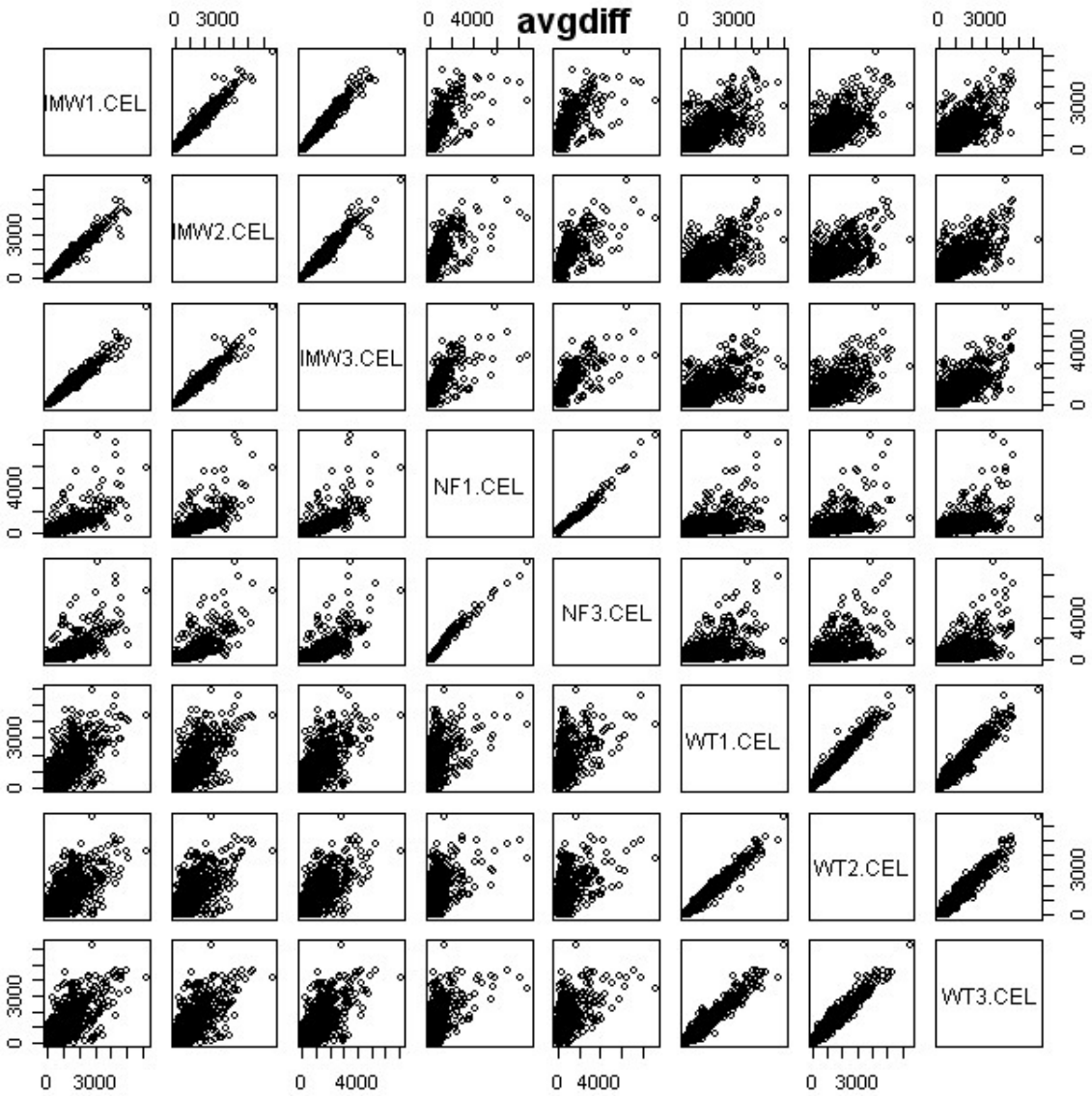


summarization

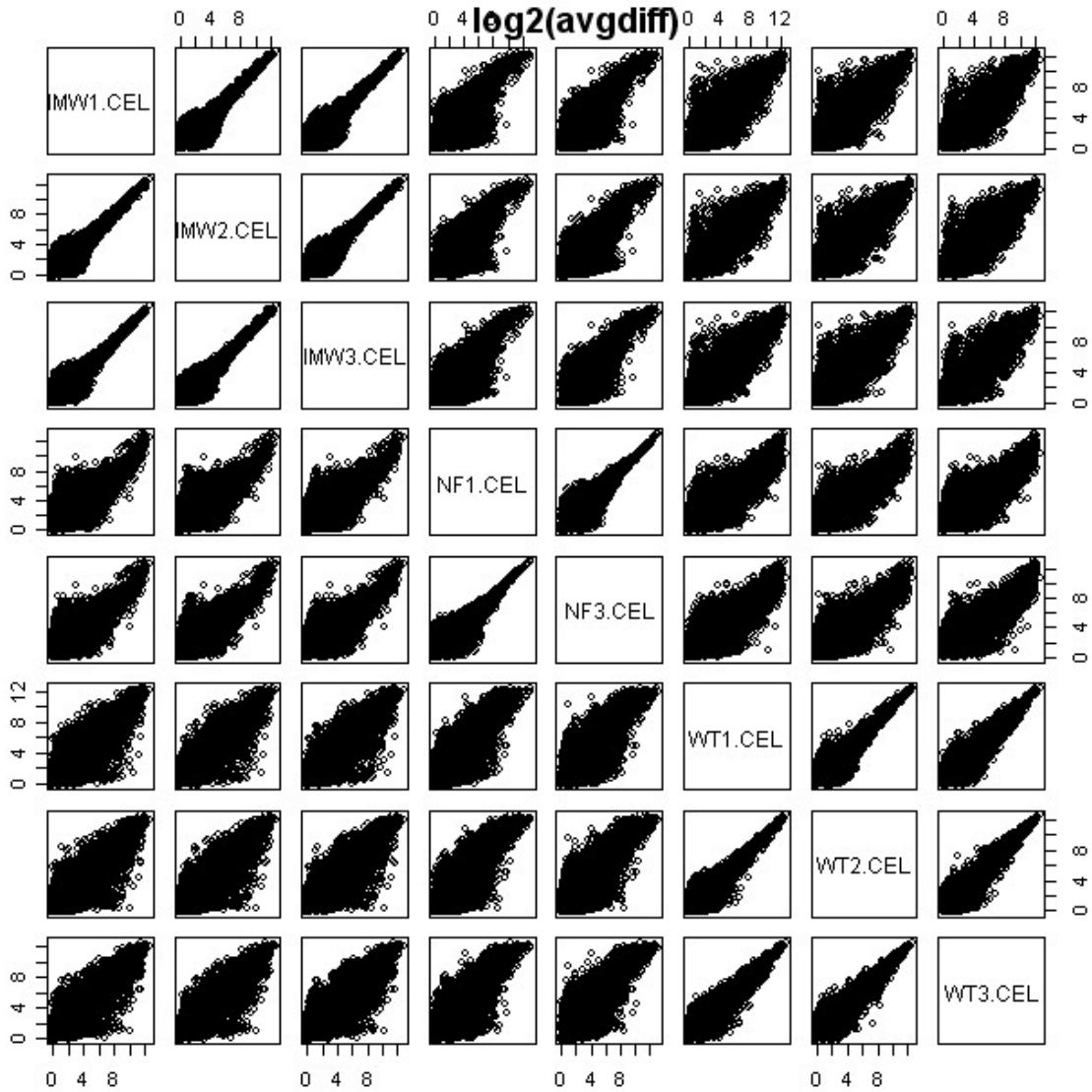


```
> express.summary.stat.methods
[1] "avgdiff"  "liwong"  "mas"  "medianpolish" "playerout"
> affy.avgdiff<-expresso(affy.testdata,
   bgcorrect.method="none",normalize.method="quantiles",
   pmcorrect.method="mas",summary.method="avgdiff")
background correction: none
normalization: quantiles
PM/MM correction : mas
expression values: avgdiff
background correcting...done.
normalizing...done.
22810 ids to be processed
|                    |
|####################|
> affy.rma<-rma(affy.testdata)
```

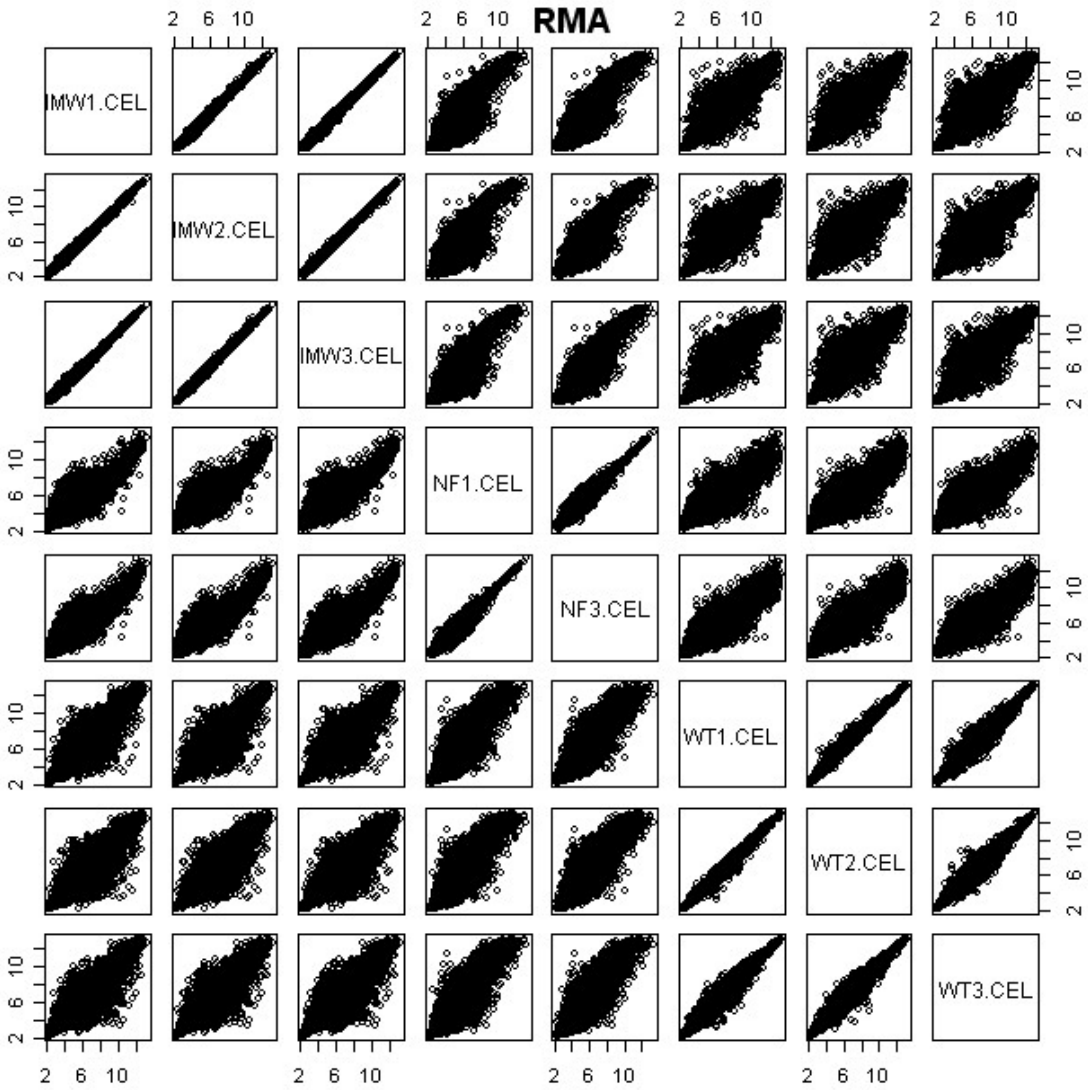
summarization



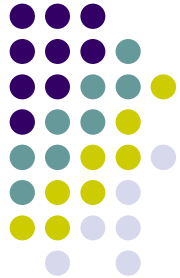
summarization



summarization

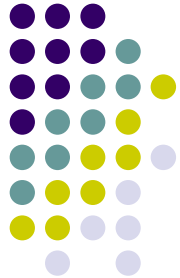


QC：affymetrix quality assessment



```
> library(simpleaffy)
> affy.qc<-qc(affy.testdata)
> avbg(affy.qc)
IMW1.CEL IMW2.CEL IMW3.CEL  NF1.CEL  NF3.CEL  WT1.CEL  WT2.CEL  WT3.CEL
49.52473 44.64997 40.61587 41.24566 42.19821 38.37762 45.36208 42.97333
> sfs(affy.qc)
[1] 0.7761812 0.7370002 0.8946128 4.3103500 3.9894275 1.0923440 1.0578635
[8] 0.9271550
> percent.present(affy.qc)
IMW1.CEL.present IMW2.CEL.present IMW3.CEL.present  NF1.CEL.present
        61.92021         60.91626         60.57869         30.25427
 NF3.CEL.present  WT1.CEL.present  WT2.CEL.present  WT3.CEL.present
        31.87199         57.10653         56.74704         58.73301
```

QC : affymetrix quality assessment



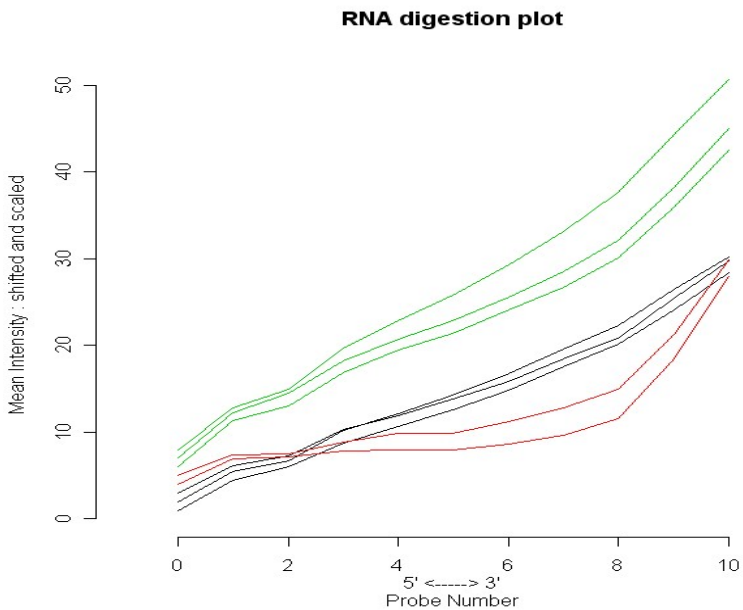
```
> ratios(affy.qc)
```

	AFFX-r2-At-Actin.3'/5'	AFFX-Athal-GAPDH.3'/5'	AFFX-r2-At-Actin.3'/M	AFFX-Athal-GAPDH.3'/M
IMW1.CEL	0.8376161	0.2735591	-0.01481408	-0.77110931
IMW2.CEL	0.8356822	0.7341535	-0.11214855	-0.37997908
IMW3.CEL	0.7701097	0.5322263	-0.16164184	-0.36318236
NF1.CEL	**0.5008100**	**1.8958175**	**-0.24559046**	**0.05781393**
NF3.CEL	**0.2677213**	**2.0154908**	**-0.31682958**	**0.64128519**
WT1.CEL	1.4853941	1.0456613	-0.08798063	-0.60097077
WT2.CEL	1.7968120	0.8101417	0.01598324	-0.58994998
WT3.CEL	1.7572941	1.4382101	0.27197692	0.10590049

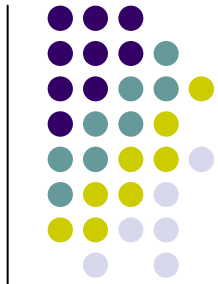
QC : RNA degradation

```
> affy.RNAdeg<-AffyRNAdeg(affy.testdata)
> plotAffyRNAdeg(affy.RNAdeg,col=c(1,1,1,2,2,3,3,3))
> summaryAffyRNAdeg(affy.RNAdeg)
```

	IMW1.CEL	IMW2.CEL	IMW3.CEL	**NF1.CEL**	**NF3.CEL**	WT1.CEL	WT2.CEL	WT3.CEL
slope	2.54e+00	2.68e+00	2.47e+00	**1.67000**	**1.920000**	3.24e+00	3.39e+00	4.00e+00
pvalue	1.66e-09	3.20e-10	1.09e-08	**0.00214**	**0.000306**	2.33e-08	2.37e-08	2.13e-09

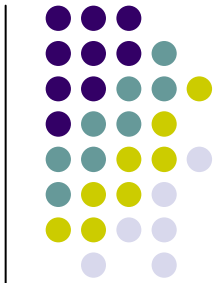


Differentially expressed genes



- Two experimental groups
  - t-test
- Multiple experimental groups
  - Analysis of Variance (ANOVA) models
  - Compare 3 or more groups (eg. dosages, 1-factor design)
  - F-test
  - permutation test
  - can add “fudge factor” if desired

Multiple Testing



- Multiple Testing
  : many hypotheses are tested simultaneously.

- Problems of Multiple Testing
  : It is very likely that a small p-value will occur by chance under null hypothesis when considering a large enough set of hypotheses.

- Notations

H_i^0 : the i-th null hypothesis

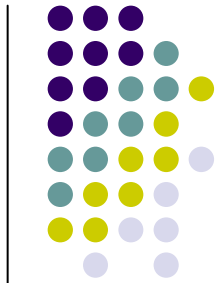
H_i^1 : the i-th alternative hypothesis

Type I and Type II Error

Number of	not rejected	rejected	
True H_0	U	V	m_0
False H_0	T	S	m_1
	m-R	R	m

- False positive (Type I error) : V
  - reject H_0 when H_0 is true
- False negative (Type II error) : T
  - accept H_0 when H_0 is false

Multiple testing problem



- Standard approach

1. Compute a test statistic T_i for each hypothesis H_i^0
2. Apply a multiple testing procedure to determine which H_i^0 to reject while controlling a suitably defined Type I error rate

- Probability of Type I error for testing H_i^0

$$= P(|T_i| \geq C_{\alpha_i} \mid H_i^0)$$

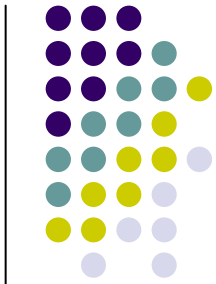
- Testing one hypothesis H_i^0

: control the probability of Type I error at level α

- Testing $\{H_1^0, \cdots H_n^0\}$ hypotheses simultaneously

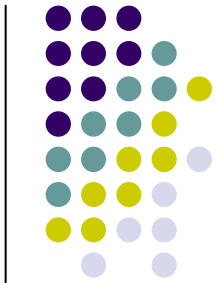
: control a particular Type I error rate at level α

Type I error rates



- PCER $= \frac{E(V)}{m}$ (The per-comparison error rate)
- PFER $= E(V)$ (The per-family error rate)
- FWER $= Pr(V \geq 1)$ The family-wise error rate)
- FDR $= E\left[\frac{V}{R} \cdot I(R > 0)\right]$ he false discovery rate)

Power



- Power of testing H_i^0

  $= P(|T_i| \geq C_{\alpha_i} \mid H_i^1 \text{ is true})$

- Common definitions of Power

1. the probability of rejecting at least one false H^0

   $= Pr(S \geq 1) = Pr(T \leq m_1 - 1)$

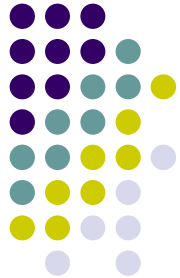
2. the average probability of rejecting the false H^0

   $= E(S)/m_1$

3. the probability of rejecting all false H^0

   $= Pr(S = m_1) = Pr(T = 0)$

Comparison of Type I error rates

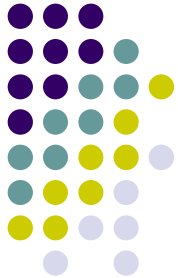


- Suppose each hypothesis H_i^0 is tested individually at level α_i

$$
\begin{aligned}
PCER &= \frac{\alpha_1 + \alpha_2 + \cdots + \alpha_m}{m} \\
&\leq \max(\alpha_1, \cdots, \alpha_m) \\
&\leq FWER \leq PFER = \alpha_1 + \cdots + \alpha_m
\end{aligned}
$$

$$
FDR \leq FWER
$$

p-value



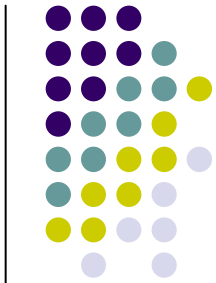
- p-value : the probability of observing a test statistic as extreme or more extreme in the direction of rejection as the observed one.

$$p_i = Pr(|T_i| \geq |t_i| \,|\, H_i)$$

- adjusted p-value : the nominal level of the entire test procedure at which H_j would just be rejected, given the values of all test statistics involved.

❋ An advantage of reporting adjusted p-values : the level of the test does not need to be determined in advance

Control of the FWER : single procedure



1. Bonferroni adjusted p-value

$$\tilde{p}_i = \min(mp_i, 1)$$

2. Šidák adjusted p-value

$$\tilde{p}_i = 1 - (1 - p_i)^m$$

3. minP adjusted p-value

$$\tilde{p}_i = Pr(\min_{1 \le l \le m} P_l \le p_i \mid H_0^c)$$

4. maxT adjusted p-value

$$\tilde{p}_i = Pr(\max_{1 \le l \le m} |T_l| \ge |t_i| \mid H_0^c)$$

- $H_0^c = \cap_{j=1}^m H_j$: the complete null
- P_l : a random variable for the unadjusted p-value

Control of the FWER : step-down

- Holm procedure

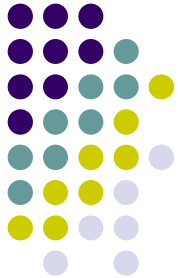
Let $p_{r_1} \leq p_{r_2} \leq \cdots \leq p_{r_m}$ the observed ordered unadjusted p-values and $H_{r_1}, H_{r_2}, \cdots, H_{r_m}$ be the corresponding null hypothesis.

Let $j^* = \min\{j : p_{r_j} > \alpha/(m - j + 1)\}$

Then, reject Hr_j, for j = 1, ⋯, j^*-1.

If no such j^* exists, reject all hypotheses.

Control of the FWER : step-down



1. step-down Holm adjusted p-values

$$\tilde{p}_{r_i} = \max_{k=1,\cdots,i} \{\min((m-k+1)p_{r_k}, 1)\}$$

2. step-down Sidak adjusted p-values

$$\tilde{p}_{r_i} = \max_{k=1,\cdots,i} \{1-(1-p_{r_k})^{(m-k+1)}\}$$

3. step-down minP adjusted p-values

$$\tilde{p}_{r_i} = \max_{k=1,\cdots,i} \left\{ Pr\left(\min_{l \in \{r_k,\cdots,r_m\}} P_l \leq p_{r_k} \mid H_0^c \right) \right\}$$

4. step-down maxT adjusted p-values

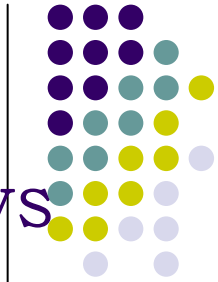
$$\tilde{p}_{r_i} = \max_{k=1,\cdots,i} \left\{ Pr\left(\max_{l \in \{r_k,\cdots,r_m\}} |T_l| \geq |t_{r_k}| \mid H_0^c \right) \right\}$$

eBayes

- Smyth (2004)
- Use the empirical Bayes approach
- shrinkage of the estimated samples variance towards a pooled estimate, resulting in far more stable inference when the number of arrays is small

$$\tilde{t}_{gj} = \frac{\hat{\beta}_{gj}}{\tilde{s}_g \sqrt{v_{gj}}}$$

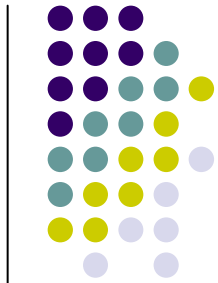
SAM：Significance Analysis of Microarrays



- Tusher, Tibshirani, and Chu (2001)
- SAM assigns score to each gene on the basis of change in gene expression relative to the standard deviation of repeated measurements
- For genes with scores greater than an adjustable threshold, SAM uses permutations of the repeated measurements to estimate the percentage of genes identified by change, the false discovery rate (FDR)

$$d_g = \frac{\overline{x}_{j2} - \overline{x}_{j1}}{s_j + s_0}$$

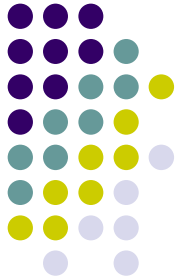
Example 2



- Arabidopsis data
- For each of 8297 genes we have

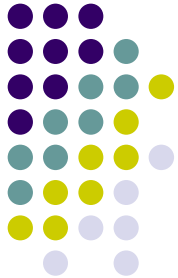
Treatment \ Genotype	Mutant (Bio)	WT
No Biotin	Bio.N.1, Bio.N.2	WT.N.1, WT.N.2
Add Biotin	Bio.B.1, Bio.B.2	WT.B.1, WT.B.2

differentially expressed genes



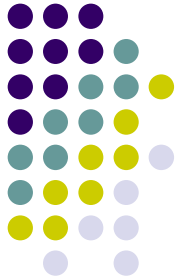
```
> biotin.s[1,]
          Bio.N.1  Bio.N.2  Bio.B.1  Bio.B.2   WT.N.1   WT.N.2  WT.B.1   WT.B.2
11986_at 7.453765 7.550523 7.621419 7.611862 7.666592 7.792472 7.63857 7.555047
> genotype<-factor(c(rep("Bio",4),rep("WT",4)))
> treatment<-factor(c(rep("No",2),rep("Add",2),
  rep("No",2),rep("Add",2)))
> chip<-factor(c(rep("Bio.No",2),
  rep("Bio.Add",2),rep("WT.No",2),rep("WT.Add",2)))
> geno.chip<-factor(c(rep("Bio",2),rep("WT",2)))
> treat.chip<-factor(c("No","Add","No","Add"))
> chip
[1] Bio.No  Bio.No  Bio.Add Bio.Add WT.No   WT.No   WT.Add  WT.Add
Levels: Bio.Add Bio.No WT.Add WT.No
```

eBayes



```
> design<-model.matrix(~0+chip)
> design
  Bio.No Bio.Add WT.No WT.Add
1      0       1     0      0
2      0       1     0      0
3      1       0     0      0
4      1       0     0      0
5      0       0     0      1
6      0       0     0      1
7      0       0     1      0
8      0       0     1      0
attr(,"assign")
[1] 1 1 1 1
attr(,"contrasts")
attr(,"contrasts")$chip
[1] "contr.treatment"
```

eBayes

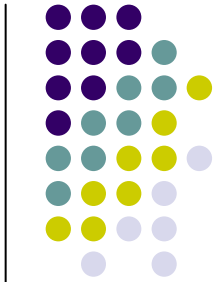


```
> fit<-lmFit(biotin.s,design)
> contrast.matrix<-
  makeContrasts(geno.eff=Bio.No+Bio.Add-WT.No-WT.Add,
+ trt.eff=Bio.No-Bio.Add+WT.No-WT.Add,int.eff=Bio.No-
  Bio.Add- WT.No+WT.Add,levels=design)
> contrast.matrix
```

	Contrasts		
Levels	geno.eff	trt.eff	int.eff
Bio.No	1	1	1
Bio.Add	1	-1	-1
WT.No	-1	1	-1
WT.Add	-1	-1	1

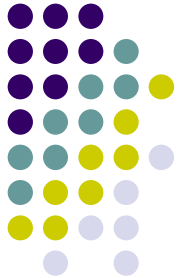
```
> fit<-contrasts.fit(fit,contrast.matrix)
> fit.eBayes<-eBayes(fit)
```

eBayes



```
> summary(fit.eBayes)
                     Length Class  Mode
coefficients         24891  -none- numeric
…
t                    24891  -none- numeric
p.value              24891  -none- numeric
lods                 24891  -none- numeric
F                     8297  -none- numeric
F.p.value             8297  -none- numeric
> sum(fit.eBayes$F.p.value<0.05)
[1] 1127
```

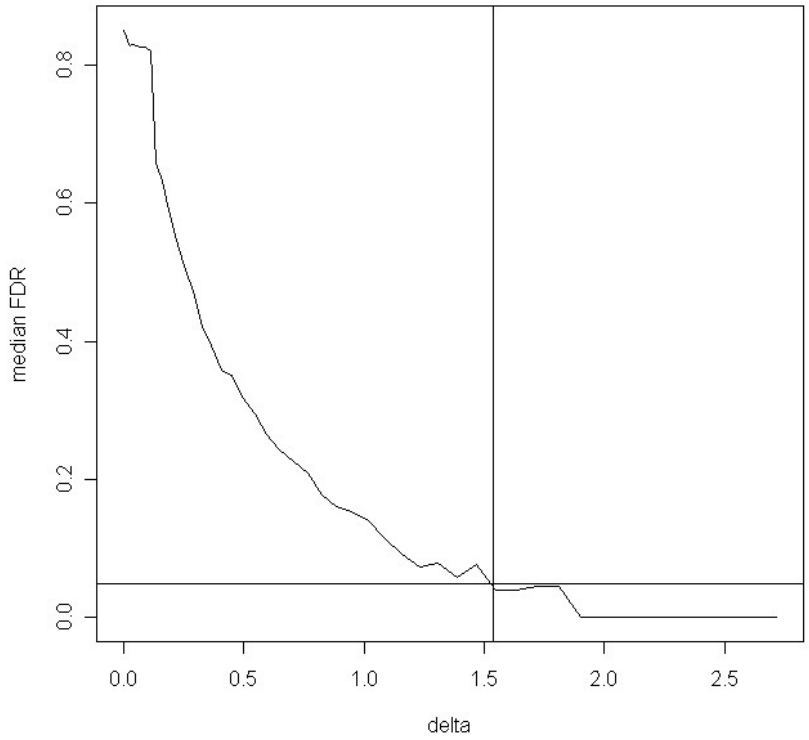
SAM



```
> library(samr)
> y<-c(1,1,2,2,1,1,2,2)
> data<-list(x=biotin.s,y=y,
   geneid=as.character(1:nrow(biotin.s)),
  genenames=colnames(biotin.s),logged2=TRUE)
> samr.obj<-samr(data,  resp.type="Two class
   unpaired", nperms=100)
> delta.table <- samr.compute.delta.table(samr.obj)
```

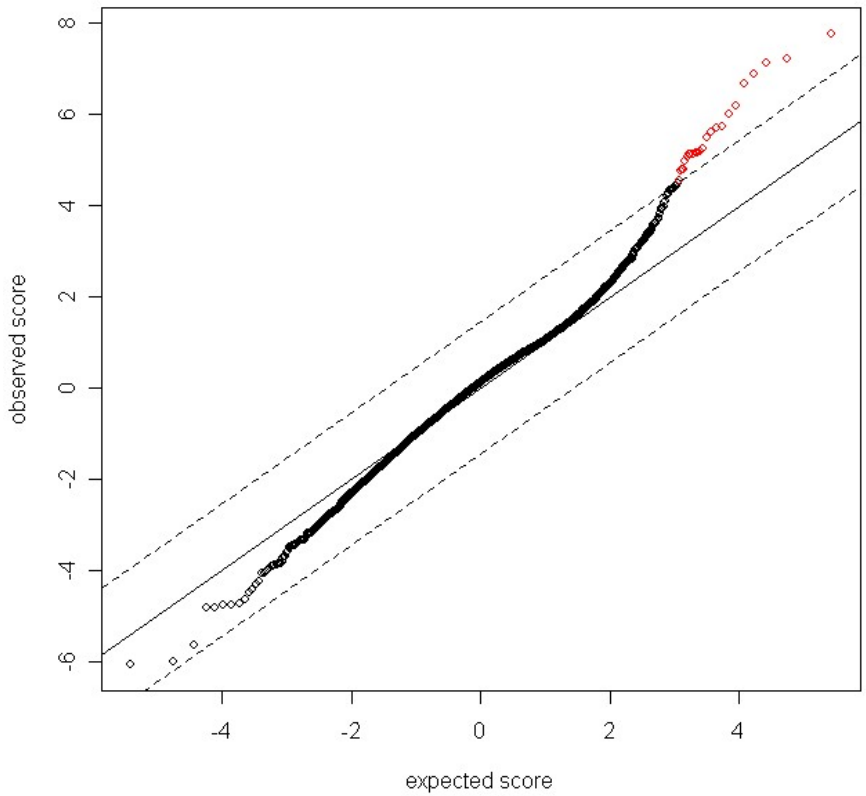
SAM

```
> plot(delta.table[,c(1,5)],type='l')
> abline(h=0.05); abline(v=1.54)
```

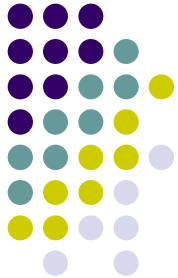


SAM

```
> delta<-1.54
> samr.plot(samr.obj,delta)
```

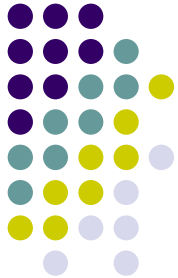


SAM



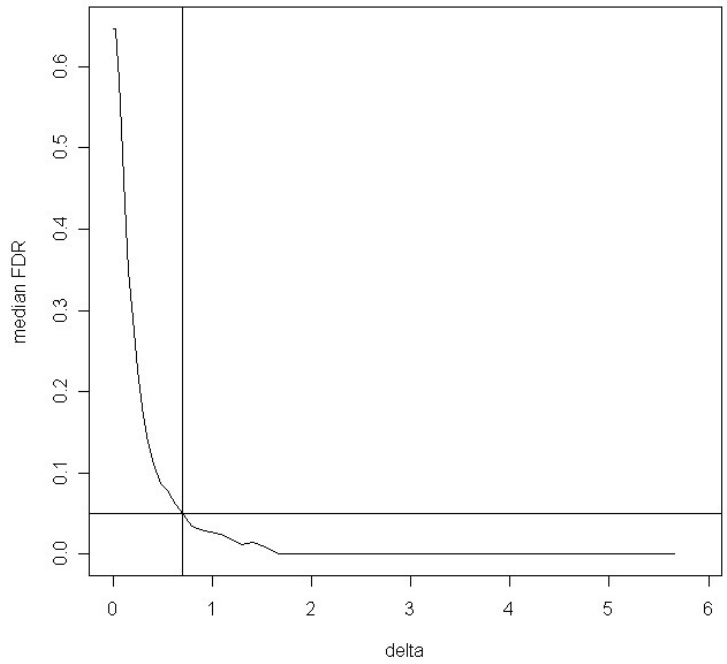
```
> siggenes.table<-samr.compute.siggenes.table
   (samr.obj,delta, data, delta.table)
> siggenes.table
$genes.up
      Row    Gene ID Gene Name Score(d)           Numerator(r)
 [1,] "141"  NA      "140"     "7.7762448041375"  "0.232237782112860"
…
     Denominator(s+s0)    Fold Change        q-value(%)
 [1,] "0.0298650297106508" "1.17501808365725" "0"
 …
$ngenes.up
[1] 24

$ngenes.lo
[1] 0
```

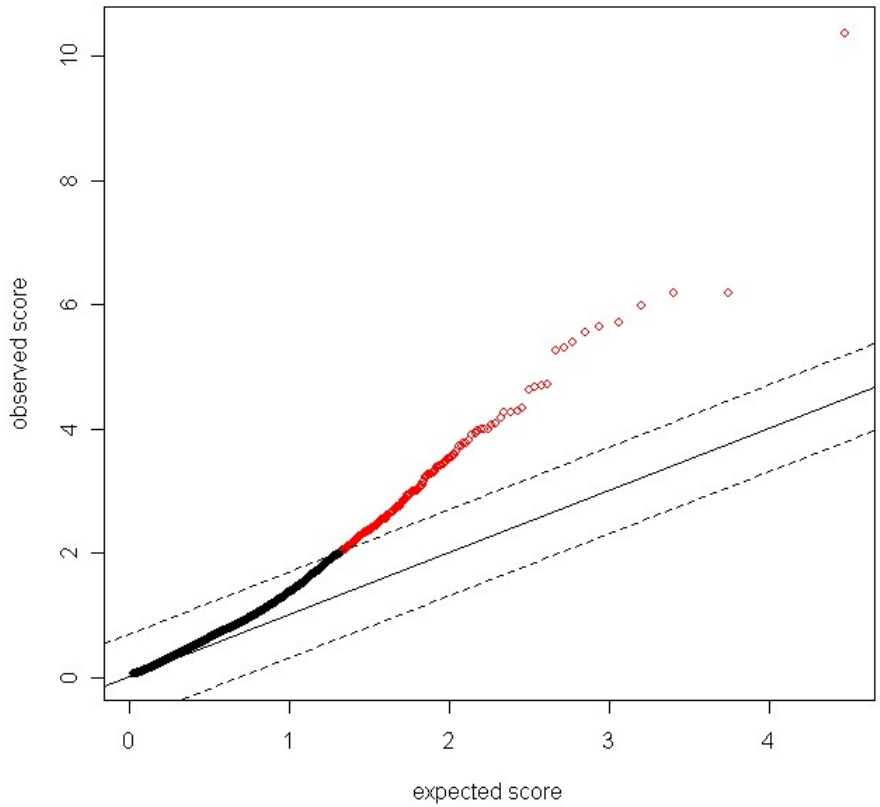
SAM



```
> library(samr)
> y<-c(1,1,2,2,3,3,4,4)
> d<-list(x=biotin.s,y=y,
   geneid=as.character(1:nrow(biotin.s)),
   genenames=colnames(biotin.s),logged2=TRUE)
> samr.obj <- samr(d,  resp.type="Multiclass")
> delta.table <- samr.compute.delta.table(samr.obj)
```

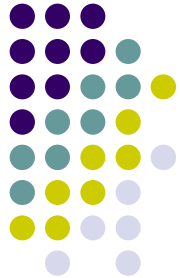
SAM






```
$ngenes.up
[1] 210
$ngenes.lo
[1] 0
```

Other Methods...



- LPE
- classification
  - LDA, QDA, Logistic regression, SVM
  - CART, Random forest, etc
  - kNN, Bagging, Boosting
- clustering
  - hierarchical clustering
  - k-means, SOM
  - PCA, Gene-shaving

Q & A ….

Thank you !!