

Microarray Data Analysis (V)

Preprocessing (i): two-color
spotted arrays

Preprocessing

- **Probe-level data:** the intensities read for each of the components.
- **Genomic-level data:** the measures being used in real research.

probe-level data  genomic-level data

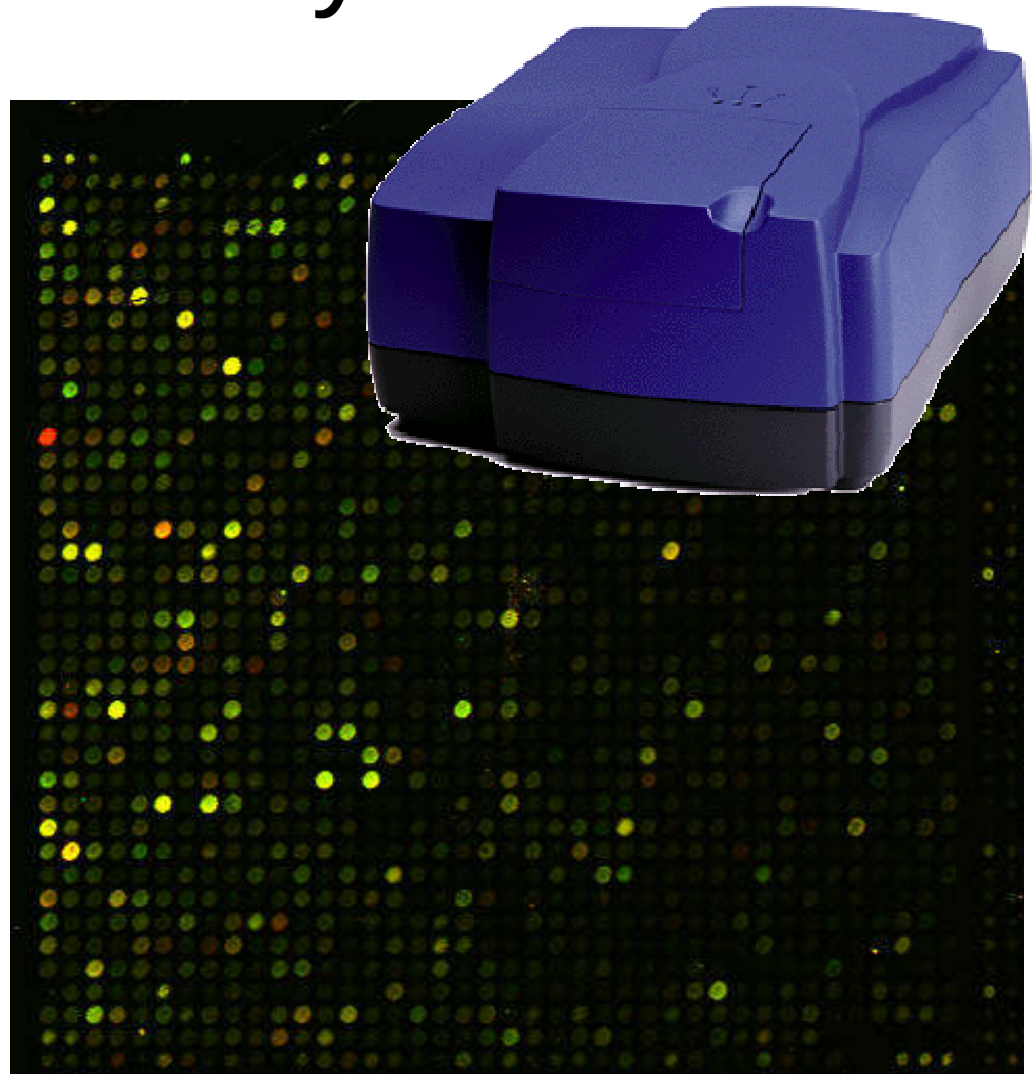
Preprocessing Data

- **Preprocessing:** A procedure that extracts meaningful data characteristics and eliminates unrelated system biases.
 - Image analysis
 - Data import
 - Normalization
 - Summarization
 - Quality assessment

Preprocessing Two-Color Spotted Arrays

Image Analysis

- The **raw data** from a microarray experiment consist of pairs of **image files**, 16-bit TIFFs, one for each of the dyes.
- Image analysis is required to extract measures of the **red** and **green** fluorescence intensities for each spot on the array.



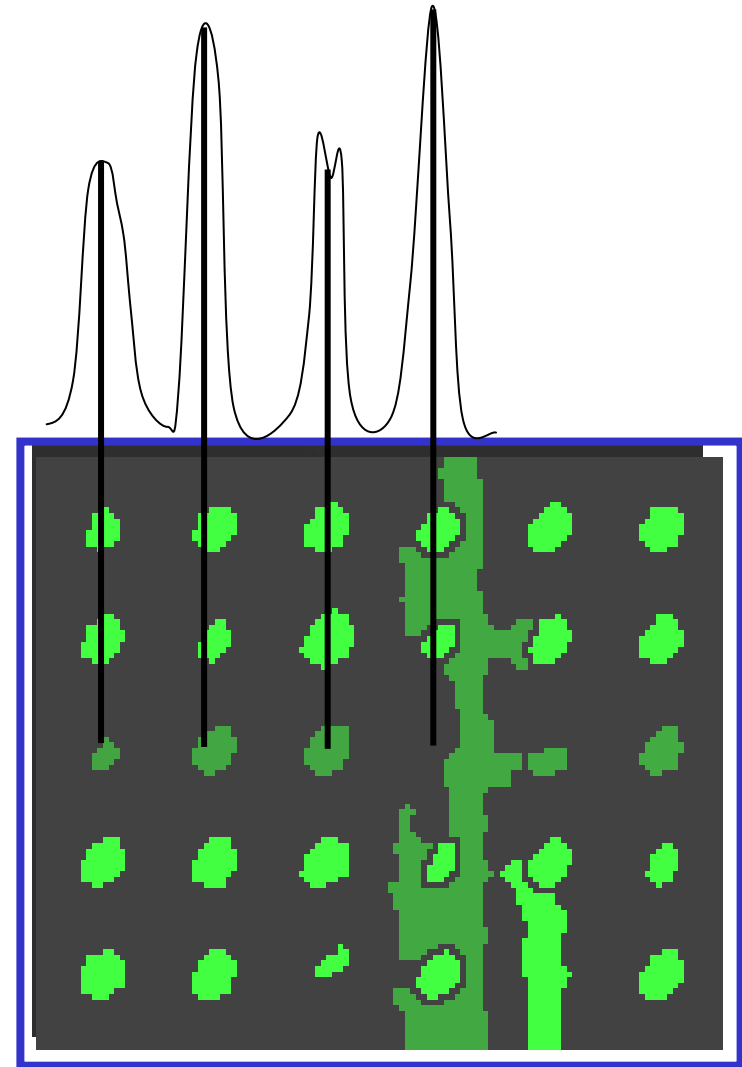
Steps in Image Analysis

1. **Addressing.** Estimate location of spot centers.

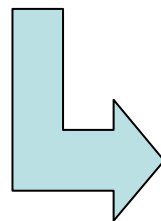
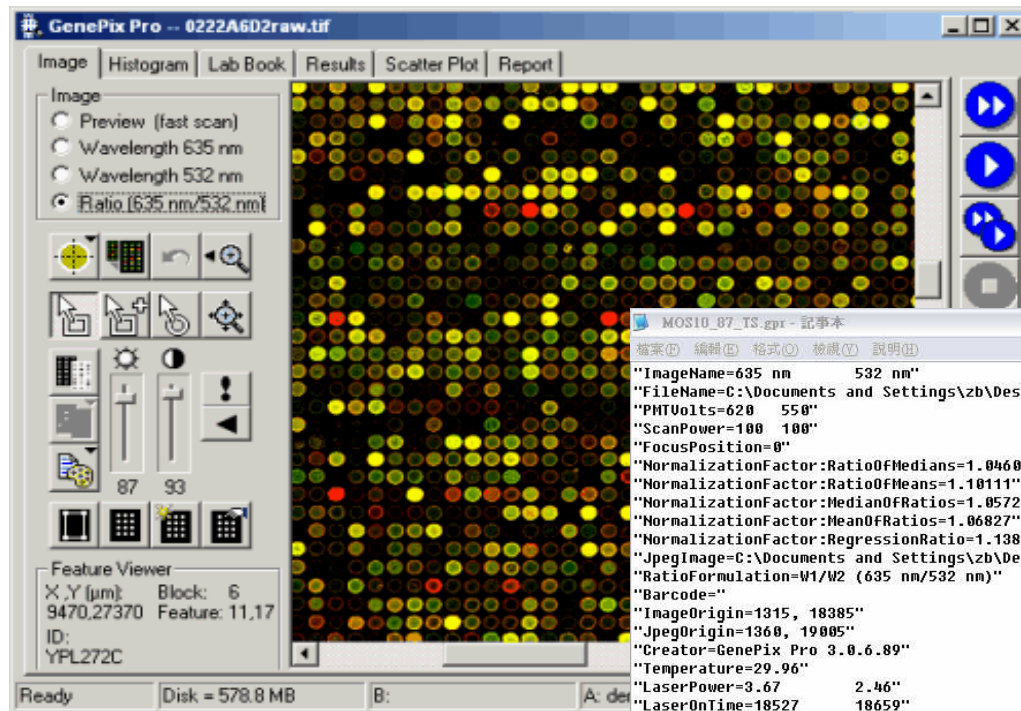
2. **Segmentation.** Classify pixels as foreground (signal) or background.

3. **Information extraction.** For each spot on the array and each dye

- signal intensities;
- background intensities;
- quality measures.



- Bioconductor does NOT provide image analysis utilities and relies on other software.



MOS10_87_TS.gpr - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

"ImageName=635 nm 532 nm"
"FileName=C:\Documents and Settings\zb\Desktop\ctrls\MOS10_87_TS_635_nm.tif C:\Documents and Settings\zb\Desktop\ctrls\MOS10_87_TS.jpg"
"PMTVolts=620 550"
"ScanPower=100 100"
"FocusPosition=0"
"NormalizationFactor:RatioOfMedians=1.04609"
"NormalizationFactor:RatioOfMeans=1.10111"
"NormalizationFactor:MedianOfRatios=1.05722"
"NormalizationFactor:MeanOfRatios=1.06827"
"NormalizationFactor:RegressionRatio=1.13858"
"JpegImage=C:\Documents and Settings\zb\Desktop\ctrls\MOS10_87_TS.jpg"
"RatioFormulation=W1/W2 (635 nm/532 nm)"
"Barcode="

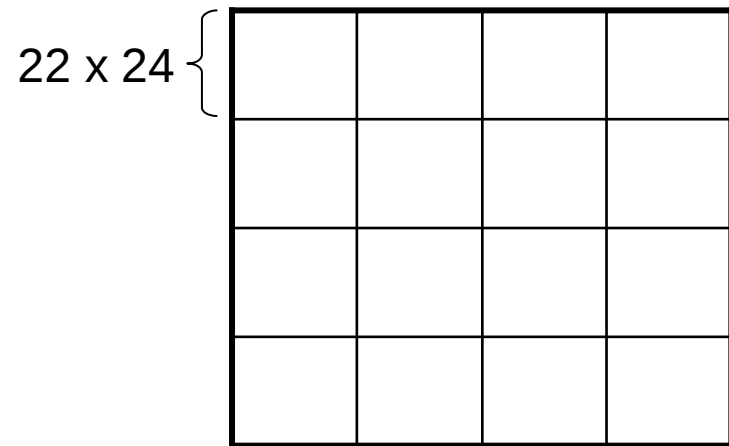
"ImageOrigin=1315, 18385"
"JpegOrigin=1360, 19005"
"Creator=GenePix Pro 3.0.6.89"
"Temperature=29.96"
"LaserPower=3.67 2.46"
"LaserOnTime=18527 18659"
"Supplier="

"Block"	"Column"	"Row"	"Name"	"ID"	"X"	"Y"	"Dia."	"F635 Median"	"F635 Mean"	"F635 SD"	"B635"		
1	1	1	"PFE0192"	"E18031_6"	1705	19300	110	9556	8251	3206	100	141	370
1	2	1	"PFF1587"	"E20827_3"	1900	19300	110	2114	1962	677	98	138	363
1	3	1	"PFE0225"	"E18877_1"	2095	19300	110	1102	1094	362	99	110	77
1	4	1	"PFF0959"	"E18651_4"	2285	19305	115	2730	2533	778	105	124	138
1	5	1	"PFF2755"	"E20360_1"	2480	19305	110	22728	20555	7836	108	138	261
1	6	1	"PFF2734"	"E714_15"	2675	19305	110	2734	2718	917	102	153	335
1	7	1	"PFF1886"	"F7915_1"	2870	19300	105	2812	2634	715	104	139	229
1	8	1	"F11738_1"	3060	19305	115	2093	2064	1008	105	118	68	98
1	9	1	"PFD0255"	"F20673_1"	3260	19305	125	2891	2532	1655	99	110	68
1	10	1	"PFN0232"	"F66030_3"	3450	19310	110	5281	4759	1740	100	109	55
1	11	1	"F54003_1"	3630	19310	105	6269	5688	1838	103	183	590	99
1	12	1	"PFF2402"	"F36648_2"	3830	19310	120	1890	1778	873	103	179	559
1	13	1	"PFF0467"	"F23644_4"	4025	19315	110	65535	56225	16431	110	133	174
1	14	1	"PFF2123"	"F57655_2"	4215	19315	110	2238	2221	964	100	111	79
1	15	1	"PFF0809"	"F36380_1"	4405	19315	110	342	383	206	99	108	62
1	16	1	"PFF0511"	"F25016_1"	4600	19320	110	1706	1675	456	101	117	100
1	17	1	"PFL0361"	"F67192_1"	4795	19320	115	4956	4726	1612	99	382	3606
1	18	1	"PFF2536"	"F39343_3"	4990	19320	120	4096	3499	1605	99	412	3665

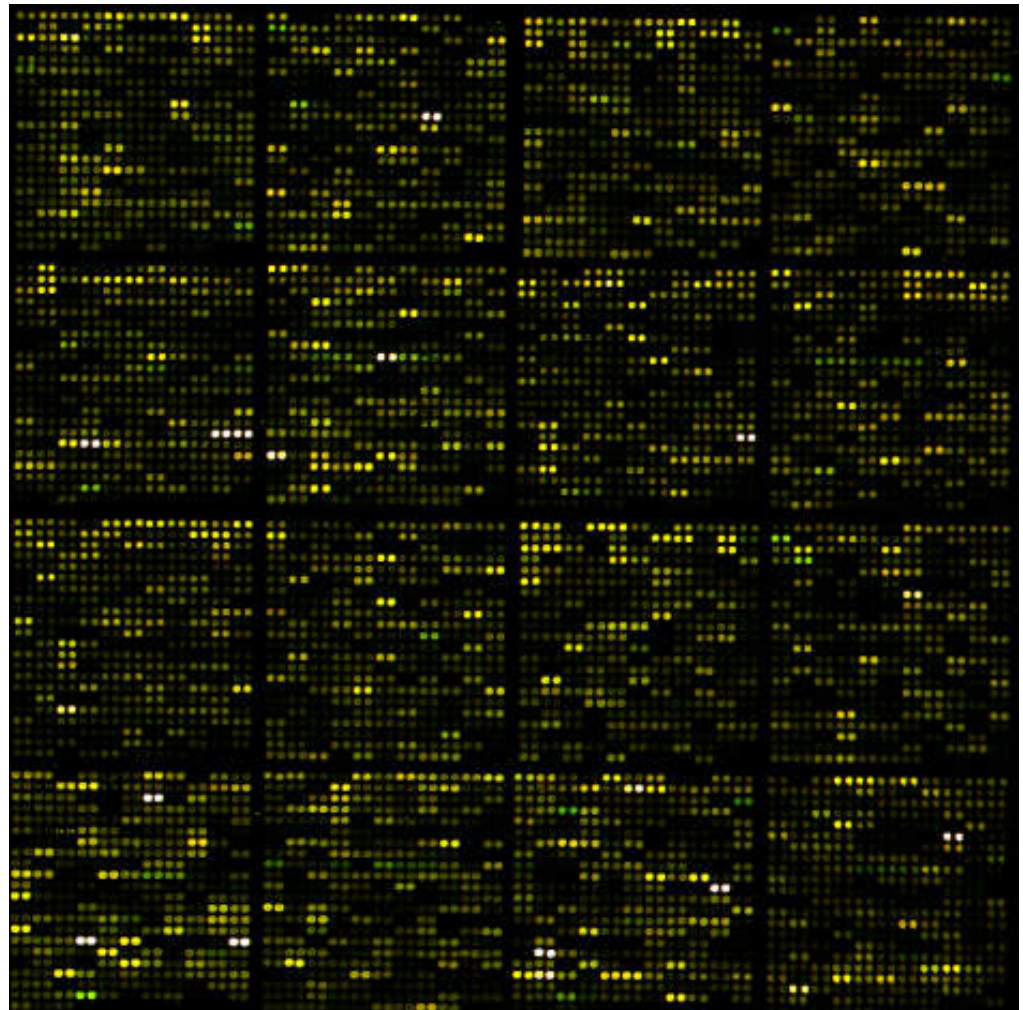
- The **marray package** helps to import the resulting data from different software:
 - GenePix: .gpr
 - Spot: .spot
 - SMD: .xls
 - Agilent: .txt

Data Import

- **Illustrative data:** *Swirl Zebrafish*
 - *Swirl* is a point mutant in the BMP2 gene of Zebrafish that affects the dorsal/ventral body axis.
 - **Objective:** to identify genes differentially expressed in the *Swirl mutant* compared to *wild-type* zebrafish
 - Microarray layout:
 - 8448 probes.
 - 4 x 4 print-tips; each grid consisted of a 22 x 24 spots matrix



$$22 \times 24 \times 4 \times 4 = 8448$$



Data Import

- Information needed for effective statistical analysis:
 - 1) the **sample target** information
 - 2) the **probe** information
 - 3) the probe spot and background **intensities**

Data Import

- 1) **the sample target information** describes which RNA samples were hybridized to each array; typically a tab-delimited text file (.txt)

1	Names	slide number	experiment Cy3	experiment Cy5	date	comments
2	swirl.1.spot	81	swirl	wild type	2001/9/20	
3	swirl.2.spot	82	wild type	swirl	2001/9/20	
4	swirl.3.spot	93	swirl	wild type	2001/11/8	
5	swirl.4.spot	94	wild type	swirl	2001/11/8	

C:\Program Files\R\R-2.5.1\library\marray\swirldata\SwirlSample.txt

1	Names	slide number	experiment Cy3	experiment Cy5	date	comments
2	swirl.1.spot	81	swirl	wild type	2001/9/20	
3	swirl.2.spot	82	wild type	swirl	2001/9/20	
4	swirl.3.spot	93	swirl	wild type	2001/11/8	
5	swirl.4.spot	94	wild type	swirl	2001/11/8	

> read.marrayInfo(*fname*)

```
> swirl.samples = read.marrayInfo(file.path(datadir, "SwirlSample.txt"))
> swirl.samples
An object of class "marrayInfo"
@maLabels
[1] "swirl.1.spot" "swirl.2.spot" "swirl.3.spot" "swirl.4.spot"

@maInfo
      Names slide number experiment Cy3 experiment Cy5      date comments
1 swirl.1.spot          81        swirl      wild type 2001/9/20        NA
2 swirl.2.spot          82      wild type          swirl 2001/9/20        NA
3 swirl.3.spot          93        swirl      wild type 2001/11/8        NA
4 swirl.4.spot          94      wild type          swirl 2001/11/8        NA

@maNotes
[1] "C:/PROGRA~1/R/R-25~1.1/library/marray/swirldata/SwirlSample.txt"
```

2) **the probe information** describes the spotted probe sequences (gene names, annotations, etc.); if using GenePix or Spot, it is a tab-delimited text file (**.gal**) with rows corresponding to spotted probe sequences and columns containing various coordinates and annotations.

C:\Program Files\R\R-2.5.1
\library\marray\swirldata\fish.gal



```
fish.gal - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

ATF      1.0
19      5
"Type=GenePix ArrayList V1.0"
"BlockCount=16"
"BlockType=0"
"Block1= 500, 500, 100, 24, 180, 22, 180"
"Block2= 4996, 500, 100, 24, 180, 22, 180"
"Block3= 9492, 500, 100, 24, 180, 22, 180"
"Block4= 13988, 500, 100, 24, 180, 22, 180"
"Block5= 500, 4996, 100, 24, 180, 22, 180"
"Block6= 4996, 4996, 100, 24, 180, 22, 180"
"Block7= 9492, 4996, 100, 24, 180, 22, 180"
"Block8= 13988, 4996, 100, 24, 180, 22, 180"
"Block9= 500, 9492, 100, 24, 180, 22, 180"
"Block10= 4996, 9492, 100, 24, 180, 22, 180"
"Block11= 9492, 9492, 100, 24, 180, 22, 180"
"Block12= 13988, 9492, 100, 24, 180, 22, 180"
"Block13= 500, 13988, 100, 24, 180, 22, 180"
"Block14= 4996, 13988, 100, 24, 180, 22, 180"
"Block15= 9492, 13988, 100, 24, 180, 22, 180"
"Block16= 13988, 13988, 100, 24, 180, 22, 180"
"Block" "Row" "Column" "ID" "Name"
1      1      1      control geno1
1      1      2      control geno2
1      1      3      control geno3
1      1      4      control 3XSSC
```

> read.Galfile(*fname*)

A screenshot of an R Console window. The title bar says "R Console". The console shows the execution of the `read.Galfile` function. The output is a list object of class "marrayInfo". It contains several components: `$names` (a character vector), `@maLabels` (a character vector with 5 elements, the first being "control"), `@maInfo` (a data frame with columns ID and Name), `@maNotes` (a character vector), `$layout` (an object of class "marrayLayout"), and `@maNgr` (a character vector).

```
> # probe information
> galinfo = read.Galfile("fish.gal",path=datadir)
> galinfo
$names
An object of class "marrayInfo"
@maLabels
[1] "control" "control" "control" "control" "control"
8443 more elements ...

@maInfo
      ID  Name
control control geno1
control.1 control geno2
control.2 control geno3
control.3 control 3XSSC
control.4 control 3XSSC
8443 more rows ...

@maNotes
[1] ""

$layout
An object of class "marrayLayout"
@maNgr
[1] 4
```

3) the probe spot and background intensities: microarray image processing results

```
> read.Spot(fnames)           # Spot  
> read.GenePix(fnames)        # GenePix  
> read.Agilent(fnames)        # Agilent
```

The data is stored as *marrayRaw* class

```
> data = read.Spot(path=datadir, targets=swirl.targets)  
Reading ... C:/PROGRA~1/R/R-25~1.1/library/marray/swirldata/swirl.1.spot  
Reading ... C:/PROGRA~1/R/R-25~1.1/library/marray/swirldata/swirl.2.spot  
Reading ... C:/PROGRA~1/R/R-25~1.1/library/marray/swirldata/swirl.3.spot  
Reading ... C:/PROGRA~1/R/R-25~1.1/library/marray/swirldata/swirl.4.spot  
> slotNames(data)  
[1] "maRf"      "maGf"      "maRb"      "maGb"      "maW"      "maLayout"  
[7] "maGnames" "maTargets" "maNotes"
```

- For *marrayRaw* and *marrayNorm* objects, various methods are defined to extract stored information

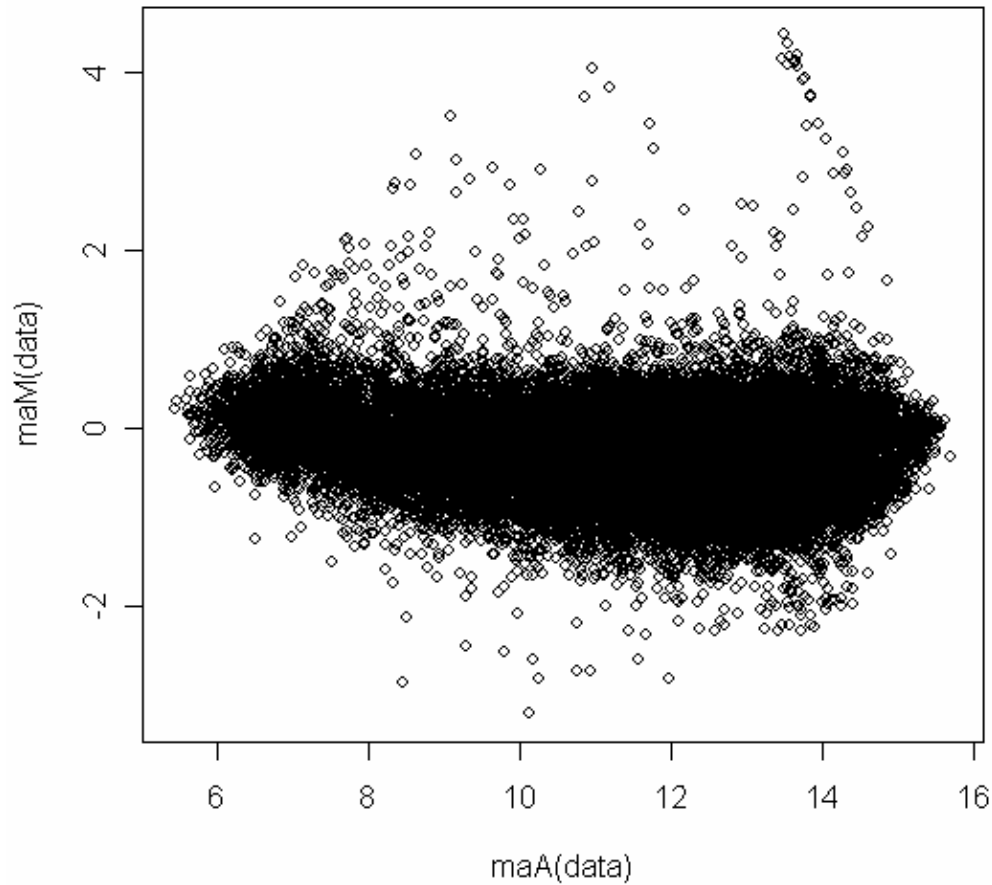
maM: $M = \log_2 \frac{Cy5}{Cy3} = \log_2(Cy5) - \log_2(Cy3)$

maA: $A = \log_2 \sqrt{Cy5 \cdot Cy3} = \frac{\log_2(Cy5) + \log_2(Cy3)}{2}$

maLG: the green log intensities

maLR: the red log intensities

maGeneTable: creates a *data.frame* of spot coordinates and gene names



```
> plot(maA(data),maM(data))
```

```
> maGeneTable(data)[1:4,1:5]
```

	Grid.R	Grid.C	Spot.R	Spot.C	ID
control	1	1	1	1	control
control.1	1	1	1	2	control
control.2	1	1	1	3	control
control.3	1	1	1	4	control

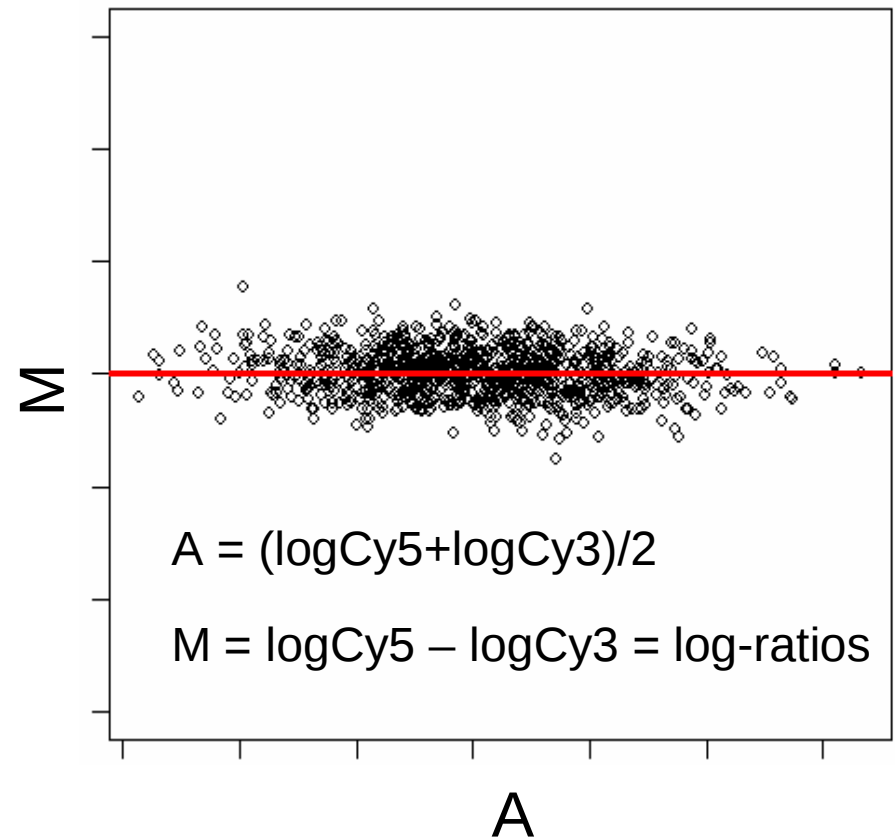
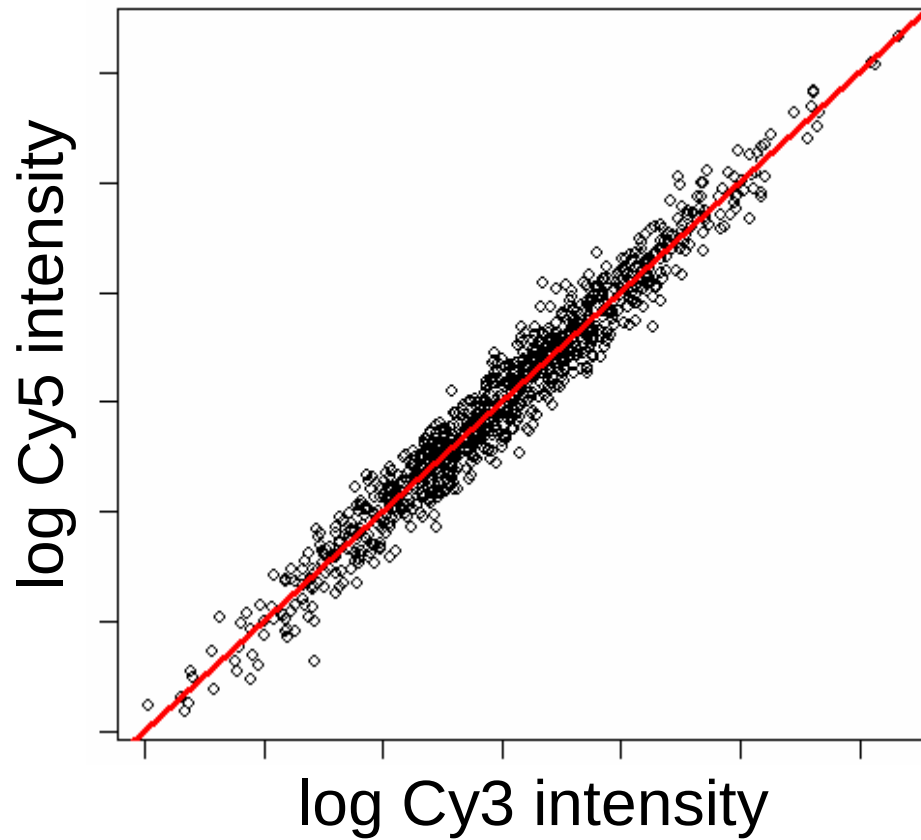
Normalization

- Identify and remove **systematic sources of variation** in the measured fluorescence intensities, other than differential expression, for example
 - different labeling efficiencies of the dyes;
 - different amounts of Cy3- and Cy5- labeled mRNA;
 - different scanning parameters;
 - sector/print-tip, spatial, or plate effects, etc.
- Caution – **Bias-variance trade-off**: more complex normalization procedures tend to be able to remove more of the technical variation but they might also remove more of the biological signals.

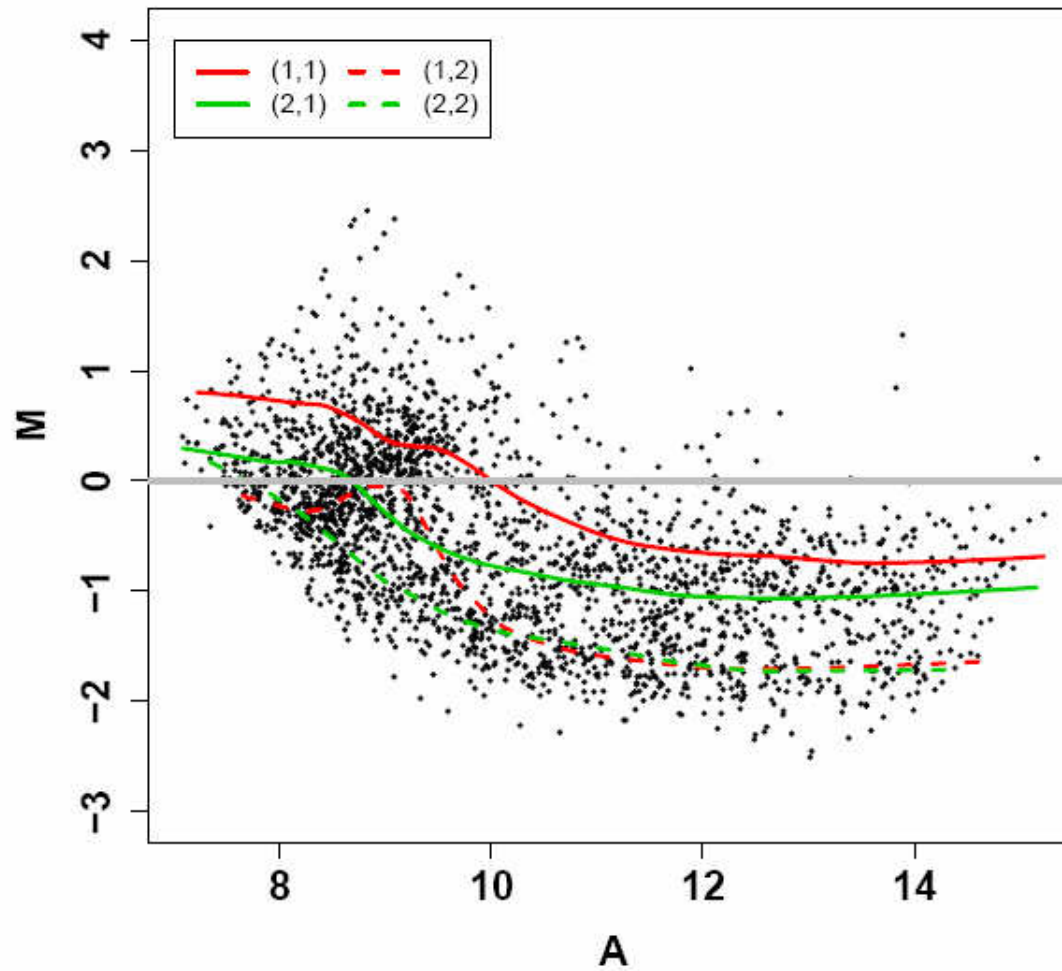
Normalization

- The need for normalization can be seen most clearly in **self-self hybridizations** where the same mRNA sample is labeled with the Cy3 and Cy5 dyes.
- The imbalance in the red and green intensities is usually **not constant** across the spots within and between arrays, and can vary according to overall spot intensity, location, plate origin, etc.
- These factors should be considered in the normalization.

Self-Self Hybridization for Two-Channel Arrays (Ideal case)



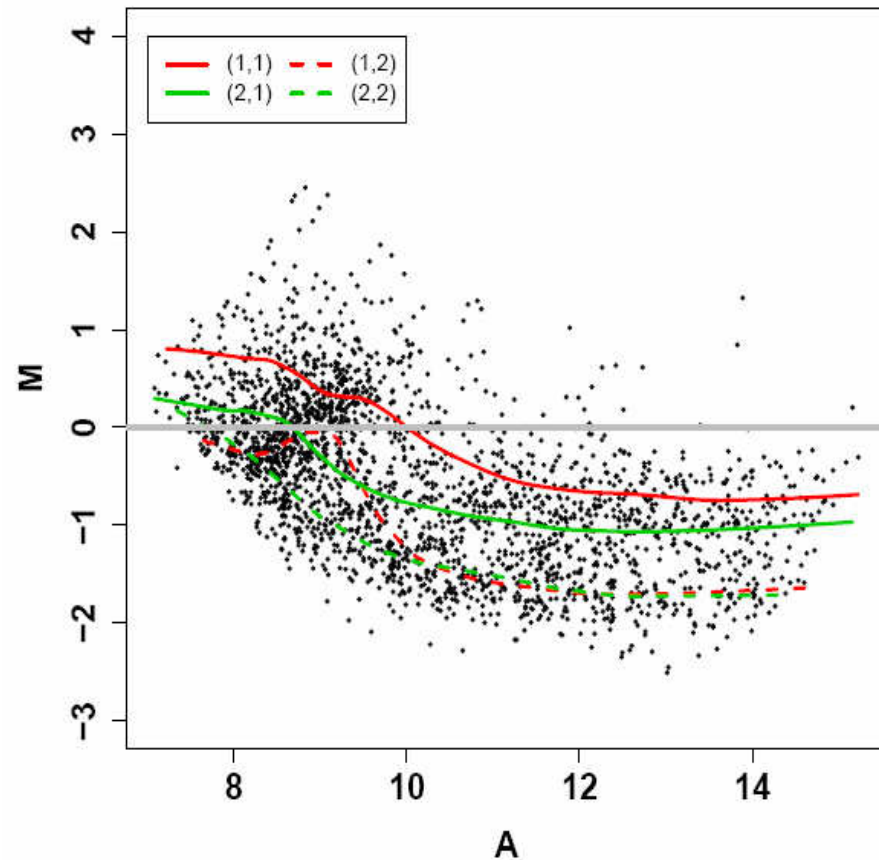
Self-Self Hybridization for Two-Channel Arrays



Source: Yang and Throne (2003)

Within-Array Normalization

- Dealing with absolute log intensities; to eliminate spot-to-spot variation
 - Location normalization:
$$M_{\text{norm}} = M - l$$
 - Scale normalization:
$$M_{\text{norm}} = M / s$$
 - Location-scale normalization:
$$M_{\text{norm}} = (M - l) / s$$



Location Normalization

- $M_{\text{norm}} = M - l$
 - It centers log-ratios around 0.
 - l can be identical for all values of M :
 - Global median normalization: $l = \text{median}(M)$
 - l can vary:
 - Global Lowess/Loess: l = the expected value after locally fitting a smooth regression curve on the global MA plot.
 - Print-tip Lowess/Loess: $l(i)$ = the loess/lowess fit to the MA-plot for the i th grid only.

LOWESS/LOESS

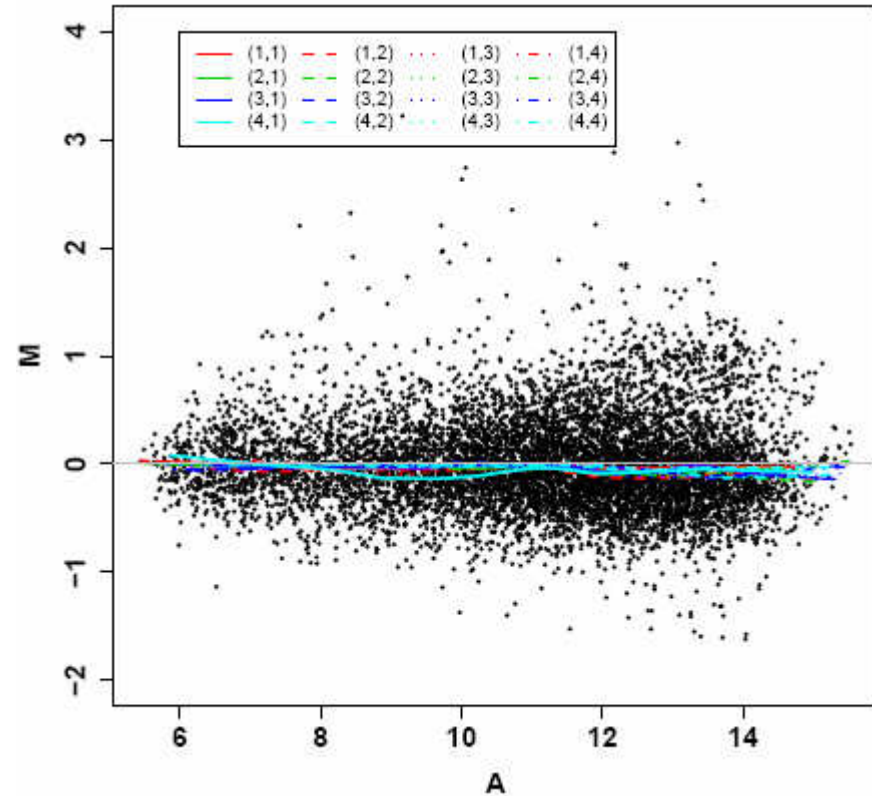
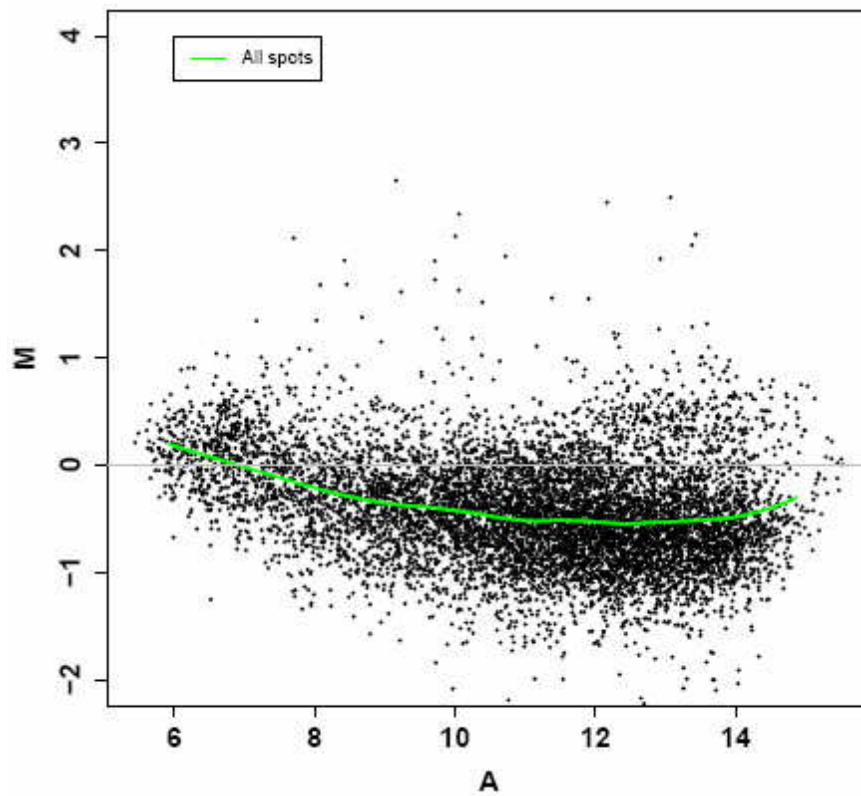
- The function fitted by LOWESS is a polynomial of the form:

$$y = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots$$

- The approach used by LOWESS/LOESS:
 1. The degrees of the polynomials used are limited to 1 (LOWESS) or 2 (LOESS) to avoid over-fitting.
 2. The data points that fall into this interval will be used to fit the first polynomial in a weighted manner.

Loess Normalization Result

Global Loess



Scale Normalization

- Scale normalization can be applied if suspecting unequal variances.
- However, it is recommended to check the need of such normalization; there is a trade-off between the possible gain in bias achieved by scale normalization and the increase in variability introduced by this additional step.
- If the scale differences are relatively small, it is preferable to perform **only** a location normalization.

R: Within-Array Normalization

maNorm (*limma*): returns a *maarrayNorm* object

Procedures	Description	Argument
None	No normalization.	n
Median	Global median location normalization.	m
Loess	Global <i>A</i> -dependent normalization using the scatter-plot smoother <code>loess</code> .	l
Print-tip <i>loess</i>	<i>A</i> -dependent normalization using the scatter-plot smoother <code>loess</code> within print-tip groups.	p
2D loess	2D-spatial normalization using the <code>loess</code> function.	twoD

maNormScale (*limma*)

Between-Array Normalization

- Dealing with **log-ratios**; to make array replicates comparable.
 - Quantile normalization method
 - VSN-method

Quantile Normalization

- **Assumption:** The measurements from different arrays share the same underlying distribution.
- It forces arrays have an identical distribution:

$$x'_i = F^{-1}(G_i(x_i))$$

G_i = empirical distribution functions for the i th array

F = the empirical distribution function of the means of the quantiles over all arrays

0.974[4]	0.341[3]	0.411[2]	-0.951[1]	= (-0.862 - 1.461 - 0.951)/3
1.857[5]	-0.036[2]	0.634[3]	-0.055[2]	
-0.386[3]	-1.461[1]	0.885[4]	0.196[3]	
-0.539[2]	0.368[4]	-0.530[1]	0.742[4]	
-0.862[1]	2.634[5]	2.340[5]	2.277[5]	



0.742[4]	0.196[3]	-0.055[2]
2.277[5]	-0.055[2]	0.196[3]
0.196[3]	-0.951[1]	0.742[4]
-0.055[2]	0.742[4]	-0.951[1]
-0.951[1]	2.277[5]	2.277[5]

Quantile Normalization

- Variants:
 - **Gquantile**: ensures that the **green** (first) channel has the same empirical distribution across arrays
 - **Rquantile**: ensures that the **red** (second) channel has the same empirical distribution across arrays
 - **Aquantile**: ensures that the **A-values** (average intensities) have the same empirical distribution across arrays
 - **Tquantile**: performs quantile normalization separately for the **groups** indicated by 'targets'

Variance Stabilization and Normalization (VSN) Method

- VSN assumes that less than half of the genes on the arrays is differentially transcribed across the experiment.
- The variance of microarray data may depend on the signal intensity; **after normalization the variance is approximately constant**
- x_{ki} = expression level of the k th probe in the i th array.

$$x'_{ki} = \text{glog} \left(\frac{x_{ki} - a_i}{b_i} \right)$$

a_i = background offset

b_i = scale parameter for array i

$\text{glog}(\cdot)$ = generalized logarithm,

R: Within-Array Normalization

normalizeBetweenArrays (limma)

Usage:

```
normalizeBetweenArrays(object, method="Aquantile", targets=NULL, ...)
```

Arguments:

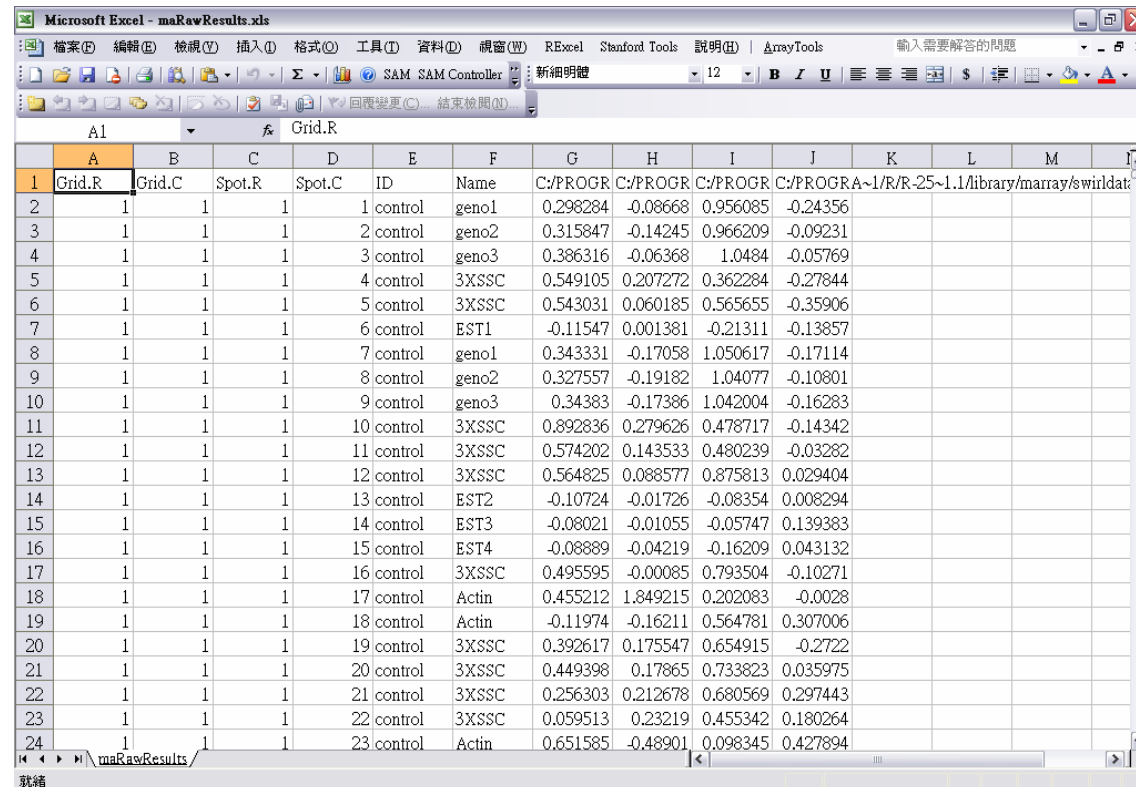
object: a 'matrix', 'RGList' or 'MAList' object containing expression ratios for a series of arrays

method: character string specifying the normalization method to be used. Choices are 'none', 'scale', 'quantile', 'Aquantile', 'Gquantile', 'Rquantile', 'Tquantile' or 'vsn'. A partial string sufficient to uniquely identify the choice is permitted.

Data Output

- Export the normalized data into a file:

> *write.marray(withinNormData)*



Microsoft Excel - maRawResults.xls

Grid.R

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Grid.R	Grid.C	Spot.R	Spot.C	ID	Name	C/PROGR	C/PROGR	C/PROGR	C/PROGR	~1/R/R-25~1.1/library/marray/swirldata		
2	1	1	1	1	1 control	geno1	0.298284	-0.08668	0.956085	-0.24356			
3	1	1	1	1	2 control	geno2	0.315847	-0.14245	0.966209	-0.09231			
4	1	1	1	1	3 control	geno3	0.386316	-0.06368	1.0484	-0.05769			
5	1	1	1	1	4 control	3XSSC	0.549105	0.207272	0.362284	-0.27844			
6	1	1	1	1	5 control	3XSSC	0.543031	0.060185	0.565655	-0.35906			
7	1	1	1	1	6 control	EST1	-0.11547	0.001381	-0.21311	-0.13857			
8	1	1	1	1	7 control	geno1	0.343331	-0.17058	1.050617	-0.17114			
9	1	1	1	1	8 control	geno2	0.327557	-0.19182	1.04077	-0.10801			
10	1	1	1	1	9 control	geno3	0.34383	-0.17386	1.042004	-0.16283			
11	1	1	1	1	10 control	3XSSC	0.892836	0.279626	0.478717	-0.14342			
12	1	1	1	1	11 control	3XSSC	0.574202	0.143533	0.480239	-0.03282			
13	1	1	1	1	12 control	3XSSC	0.564825	0.088577	0.875813	0.029404			
14	1	1	1	1	13 control	EST2	-0.10724	-0.01726	-0.08354	0.008294			
15	1	1	1	1	14 control	EST3	-0.08021	-0.01055	-0.05747	0.139383			
16	1	1	1	1	15 control	EST4	-0.08889	-0.04219	-0.16209	0.043132			
17	1	1	1	1	16 control	3XSSC	0.495595	-0.00085	0.793504	-0.10271			
18	1	1	1	1	17 control	Actin	0.455212	1.849215	0.202083	-0.0028			
19	1	1	1	1	18 control	Actin	-0.11974	-0.16211	0.564781	0.307006			
20	1	1	1	1	19 control	3XSSC	0.392617	0.175547	0.654915	-0.2722			
21	1	1	1	1	20 control	3XSSC	0.449398	0.17865	0.733823	0.035975			
22	1	1	1	1	21 control	3XSSC	0.256303	0.212678	0.680569	0.297443			
23	1	1	1	1	22 control	3XSSC	0.059513	0.23219	0.455342	0.180264			
24	1	1	1	1	23 control	Actin	0.651585	-0.48901	0.098345	0.427894			

Quality Assessment

- Before and after normalization, it is important to consider and unsure the quality of the data. $\Rightarrow$ Visualization tools.
- The package *arrayQuality* gives user a quick visual way to access the quality of individual arrays by providing per-slide diagnostic plots.

```
> maQualityPlots(data, dev="jpeg")
```

dev: A "character" string naming the graphics device. This will take arguments "png", "jpeg" and "ps" only. By default, dev is set to "png".

diagPlot.swirl.1.png : Qualitative Diagnostic Plots

Call: list(maNormLoess(x = "maA", y = "maM", z = "maPrintTip", w = NULL, subset = subset, span = span, ...))

