Volcano plot: R

topTags

> results=topTags(ms, n=20000, adjust.method="fdr", ~~lfc=0~~)

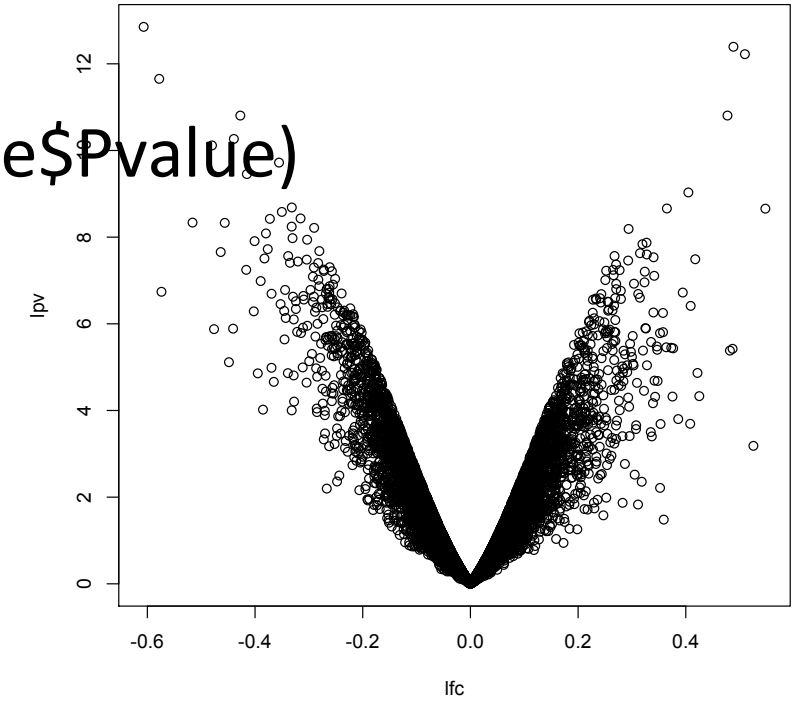
> ~~logfc=results[,2]~~

> logfc=results$table$logFC

~~> logpvalue=-log2(results[,3])~~

> logpvalue=-log2(results$table$Pvalue)

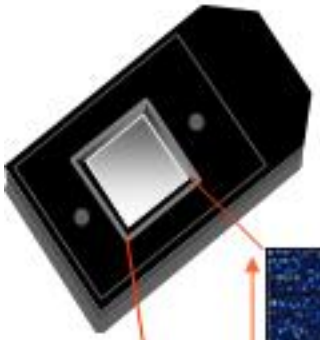
> plot(logfc,logpvalue)



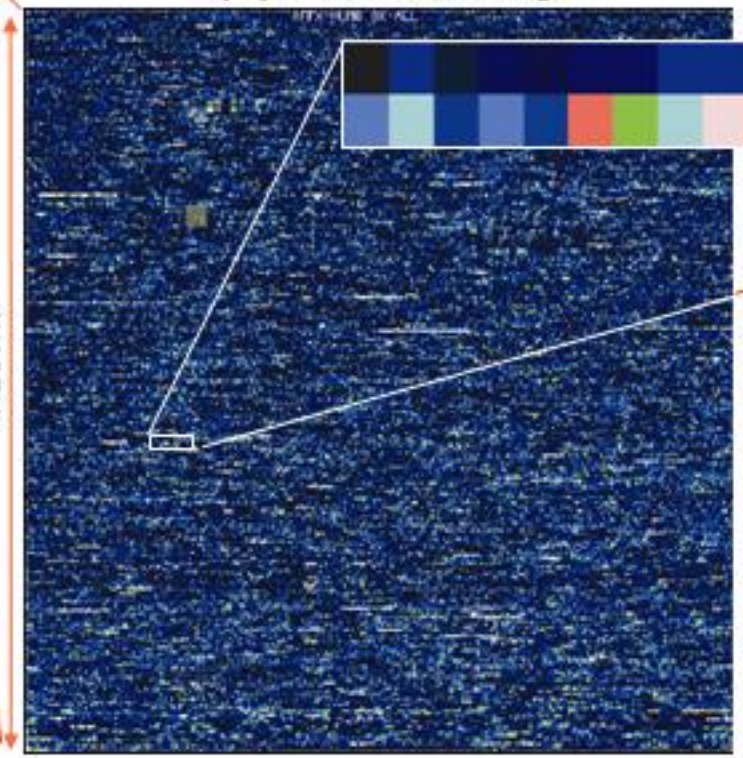
Microarray

Lecture 2

Human Genome U133A GeneChip® Array



(1) Probe Array



(2) Probe Set

Each Probe Set contains 11 Probe Pairs (PM:MM) of different probes

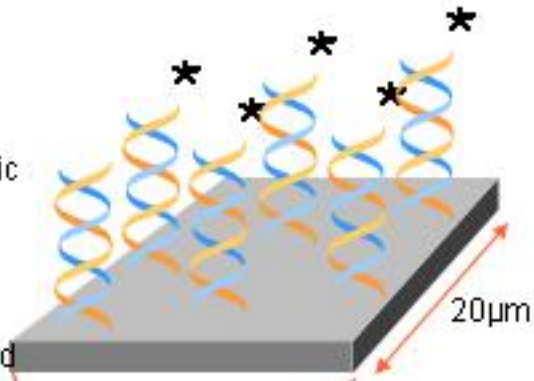
(3) Probe Pair

Each Perfect Match (PM) and MisMatch (MM) Probe Cells are associated by pairs

(4) Probe Cell

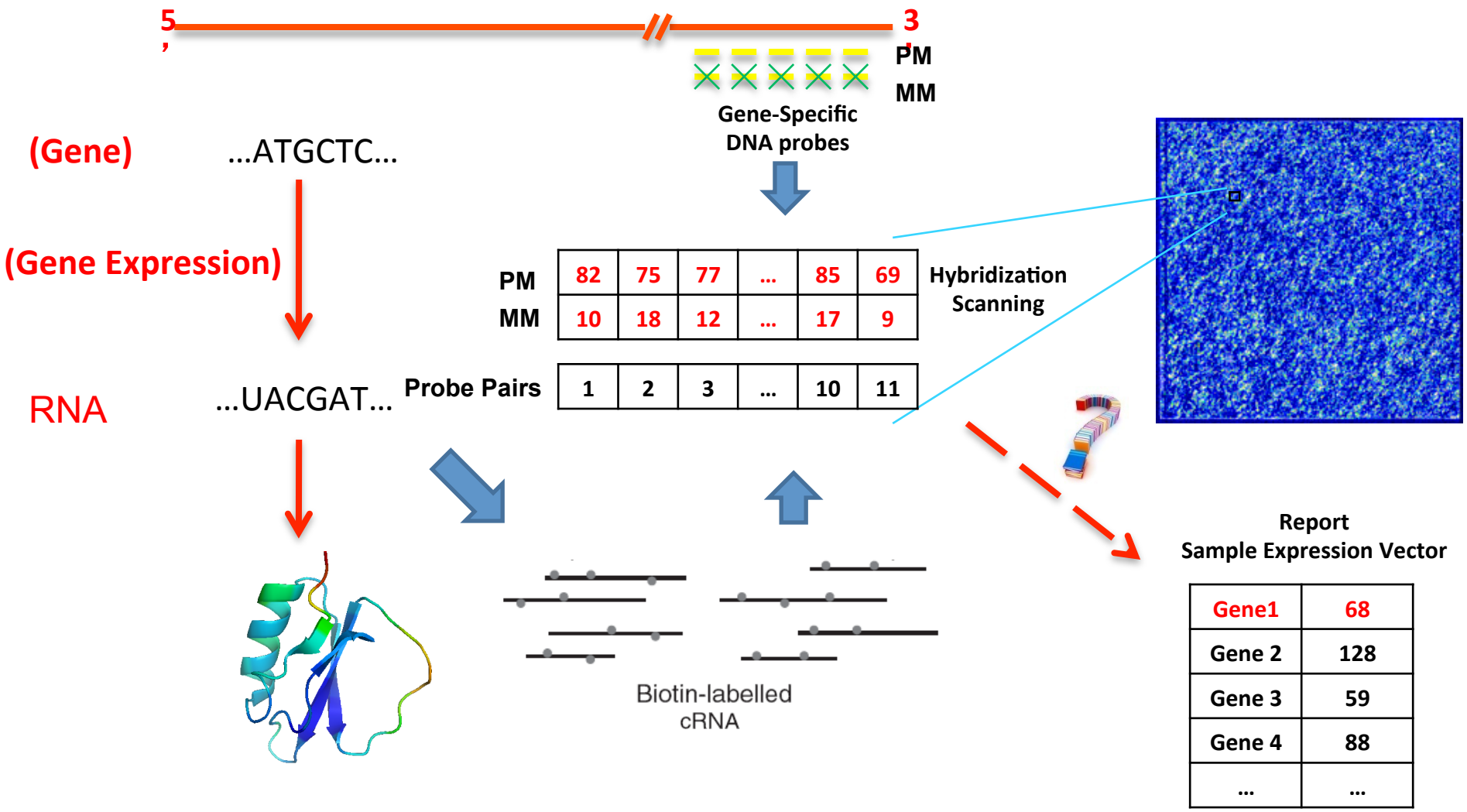
Each Probe Cell contains ~40×10^7 copies of a specific probe complementary to genetic information of interest

probe : single stranded, sense, fluorescently labeled oligonucleotide (25 mers)



The Human Genome U133 A GeneChip® array represents more than 22,000 full-length genes and EST clusters.

Affymetrix Microarray



Bioconductor package for Affymetrix data

- affy: provides a number of statistical methods for the analysis of Affymetrix oligonucleotide arrays
  - library("affy")

- affydata: Affymetrix data for demonstration purposes
  - library("affydata")
  - data(Dilution)
    - The data in Dilution is a small sample of probe sets from 2 sets of duplicate arrays hybridized with different concentrations of the same RNA

Pre-processing affy microarray

BioConductor breaks down the pre-processing of Affy microarray into four steps. Different algorithms can be chosen at each step. It is highly likely that the pre-processing results will change depending on the choices at each steps.

1. Background correction
2. Normalization
3. PM-MM correction (optional)
4. Summarization

Outline

- Background
  - Biology Background
  - Introduction to R and Bioconductor
- Preprocessing of oligonucleotide microarray
  - mas
  - rma
- Quality Assessment for oligonucleotide Microarray
- Differential Expression Testing
- Multiple Testing Procedures
- Data Visualization, Distance Measures
- Clustering
- Gene Annotation and Enrichment Analysis

Outline

- Background
  - Biology Background
  - Introduction to useful packages in Bioconductor
- Preprocessing of oligonucleotide microarray
  - mas
  - rma
- Quality Assessment for oligonucleotide Microarray
- Differential Expression Testing

Preprocess methods in Bioconductor

- MAS (Microarray Analysis Suite) 5.0
- RMA (Robust Multi-array Average)
- These two are the most popular methods for preprocessing Affymetrix data. Each method consists of different algorithms at each step of preprocessing.

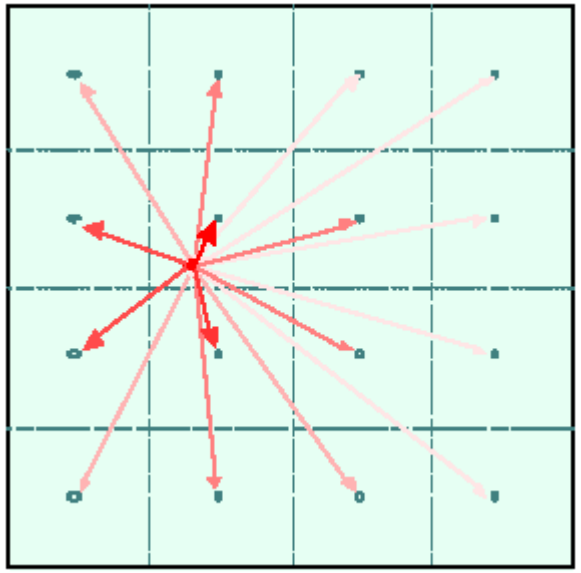
MAS5 Model

- Measured Value = N + P + S
  - N = Noise
  - P = Probe effects (non-specific hybridization)
  - S = Signal

MAS5: Background & Noise

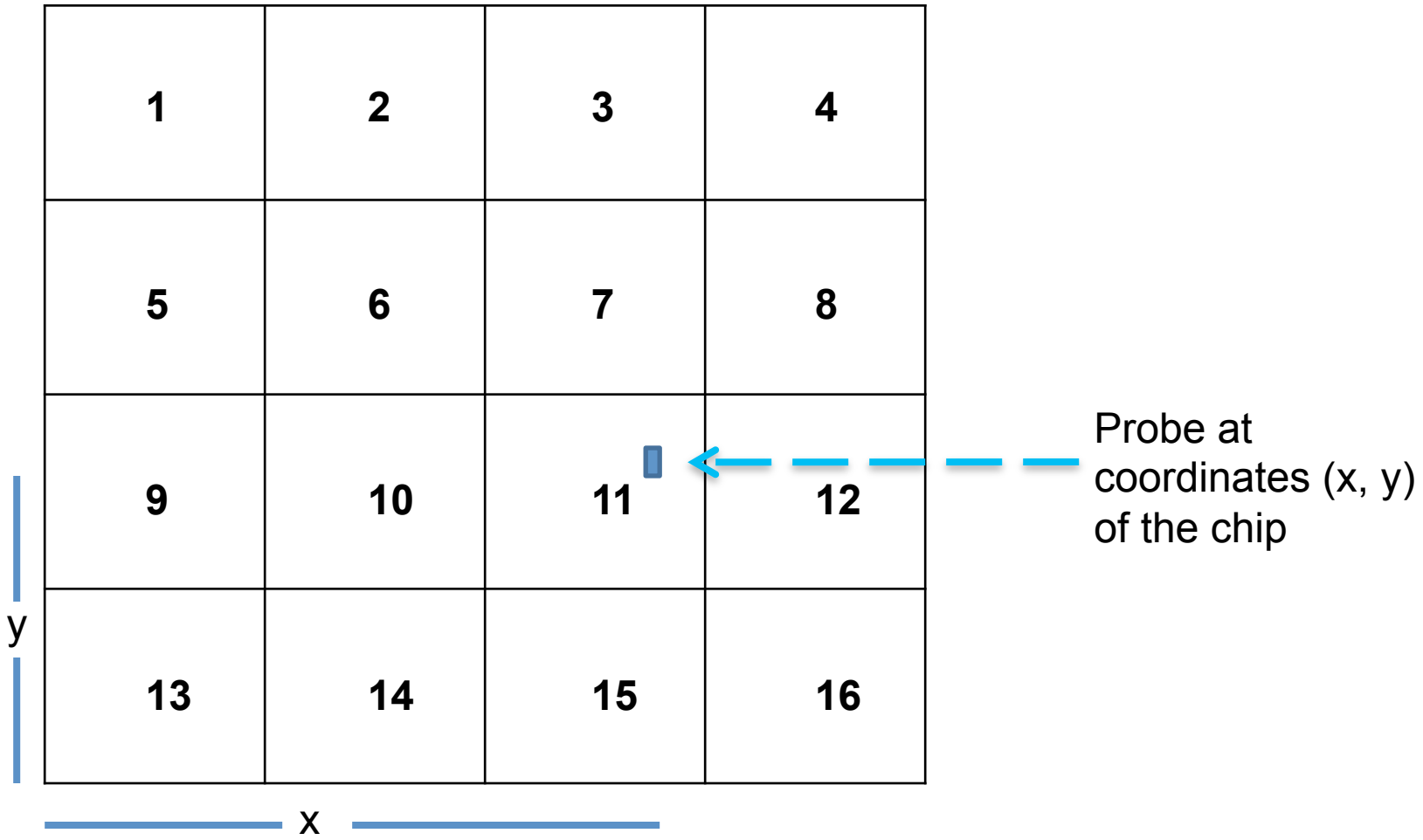
Background

- Divide chip into zones
- Select lowest 2% intensity values
- stdev of those values is zone variability
- Background at any location is the sum of all zones background, weighted by 1/ ((distance^2) + fudge factor)



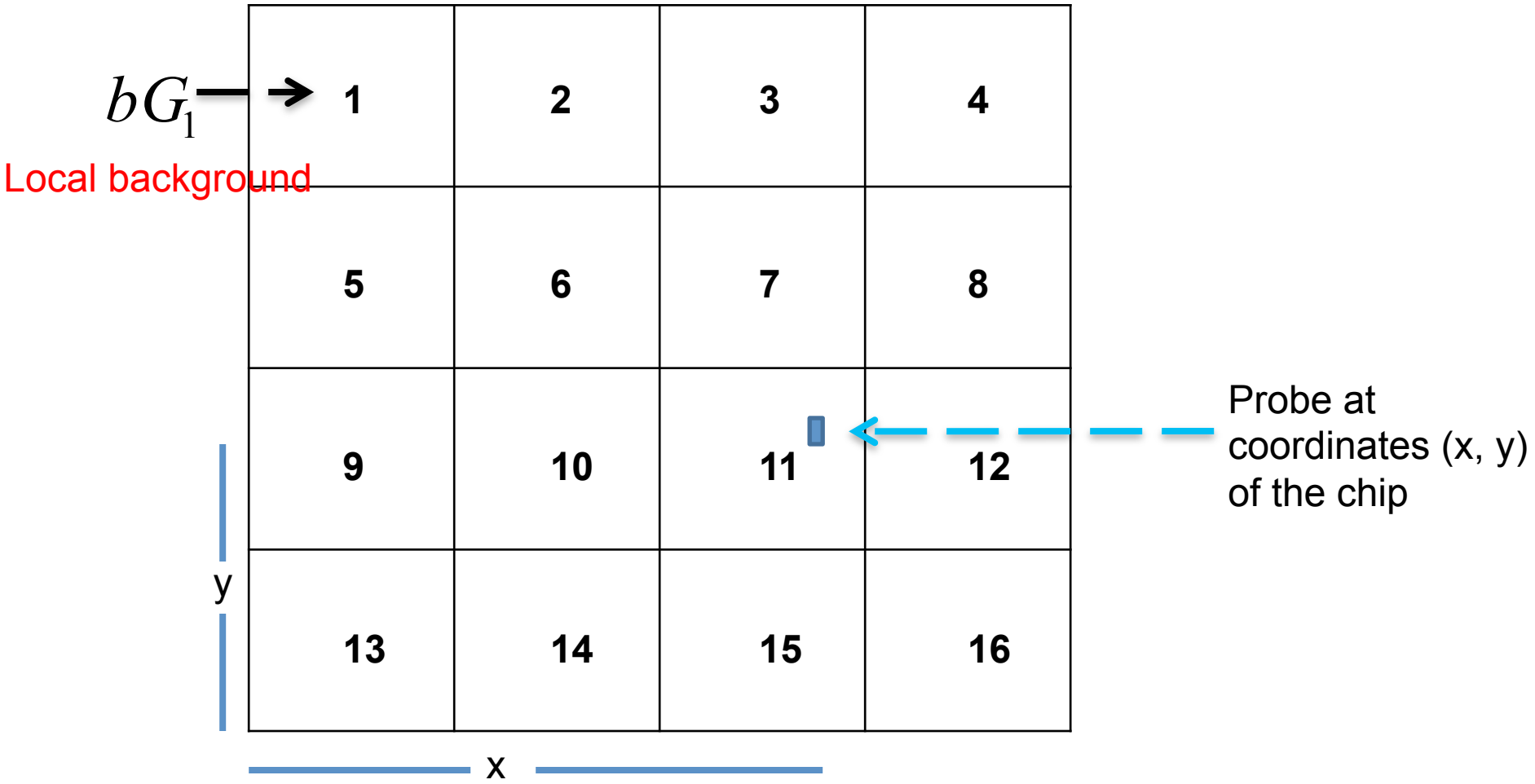
- From http://www.affymetrix.com/support/technical/whitepapers/sadd_whitepaper.pdf

MAS 5.0: Background correction



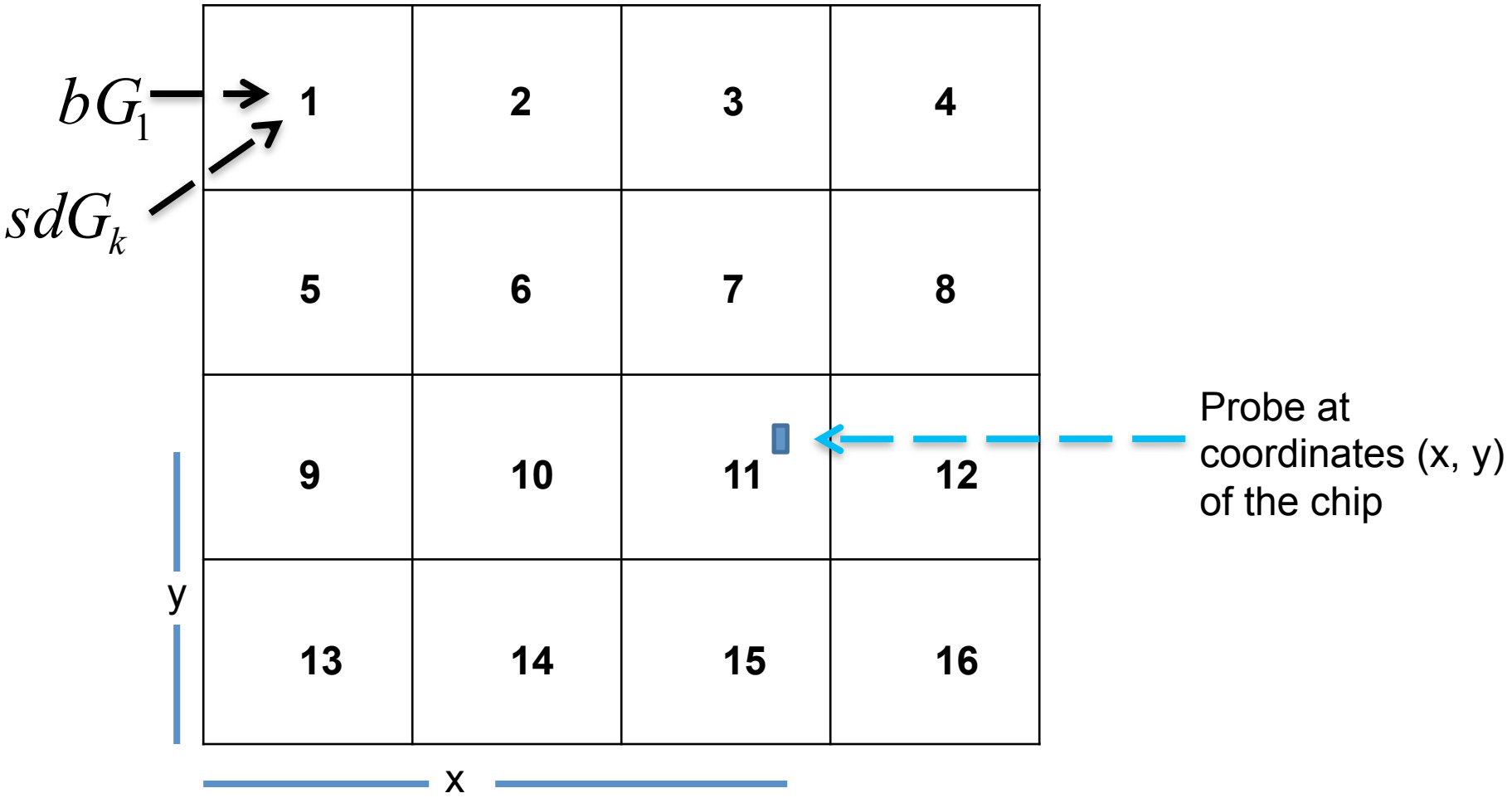
Each chip is evenly divided into 16 grids of rectangular regions.

MAS 5.0: Background correction



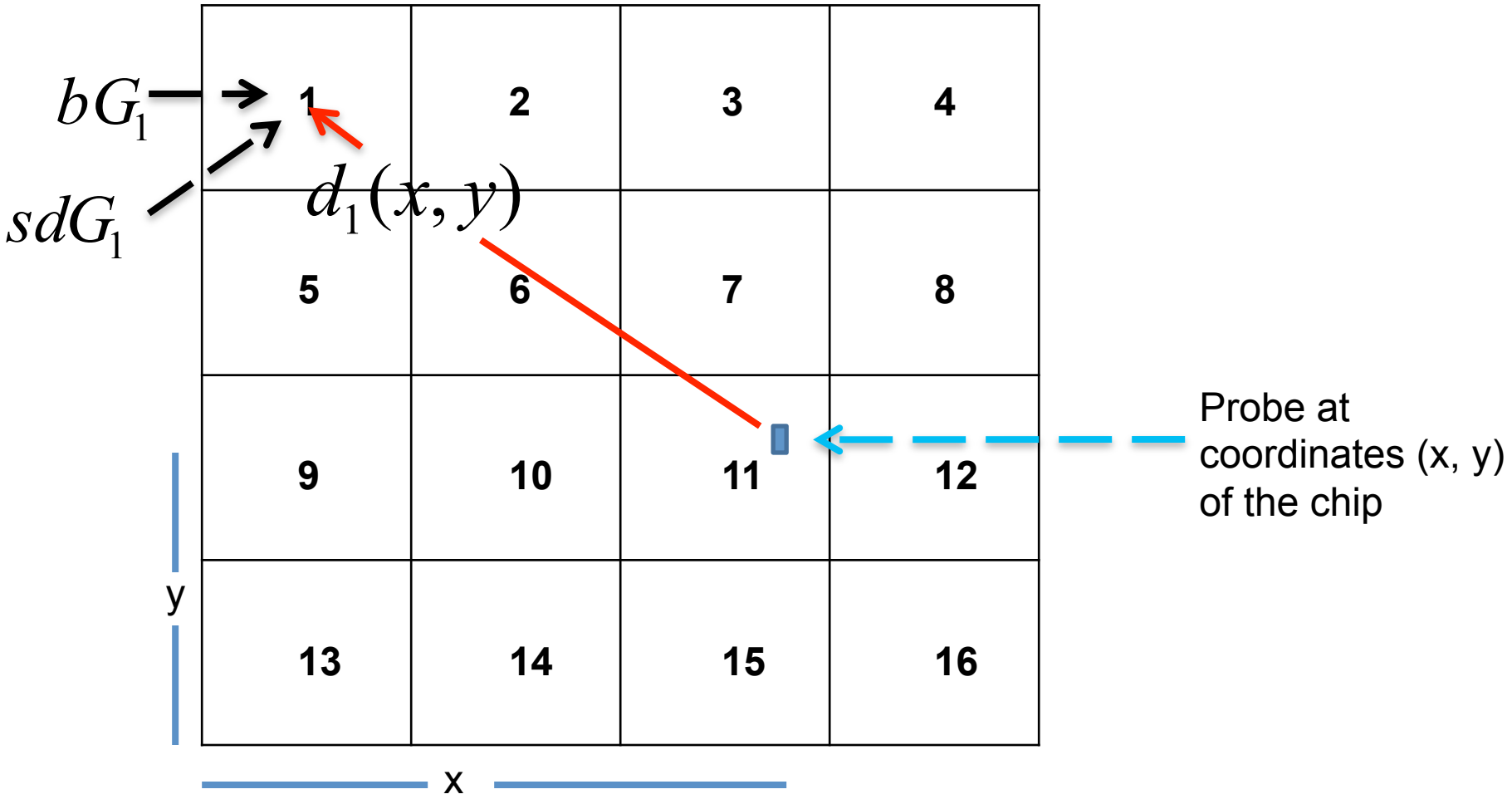
The lowest 2% of intensities in each grid are averaged to form a grid-specific background value denoted bG_k for grids k=1, 2, ..., 16.

MAS 5.0: Background correction



The standard deviation of the lowest 2% of intensities in each grid is calculated and denoted sdG_k for grids k=1, 2, ..., 16.

MAS 5.0: Background correction



Let d_k(x,y) denote the distance from the center of gird k to a probe located at coordinates (x,y) on the chip, k=1,2,…,16.

MAS 5.0: Background correction

- Based on the distance between a given probe (x, y) and the center of a grid K, we define a weight function as

$$\mathbf{W_k(x,y) = \frac{1}{d_k^{\,2}(x,y) + 100}}$$

- background intensity of this probe is the weighted average of background value of each grid:

$$\mathbf{b(x,y) = \frac{\sum_{k=1}^{16} W_k(x,y) \times bG_i}{\sum_{k=1}^{16} W_k(x,y)}}$$

- the standard deviation

$$\mathbf{sdb(x,y) = \frac{\sum_{k=1}^{16} W_k(x,y) \times sdG_i}{\sum_{k=1}^{16} W_k(x,y)}}$$

MAS 5.0: Background correction

- Let *I(x, y)* denote the raw intensity of probe (x, y), that is, the probe located at coordinate (x, y) on the chip, the background corrected intensity will be:

$$\mathbf{CI}(\mathbf{x},\mathbf{y}) = \max(\mathbf{I}'(\mathbf{x},\mathbf{y}) - \mathbf{b}(\mathbf{x},\mathbf{y}), 0.5 \times \mathbf{sdb}(\mathbf{x},\mathbf{y}))$$

$$\mathbf{I}'(\mathbf{x},\mathbf{y}) = \max(\mathbf{I}(\mathbf{x},\mathbf{y}), 0.5)$$

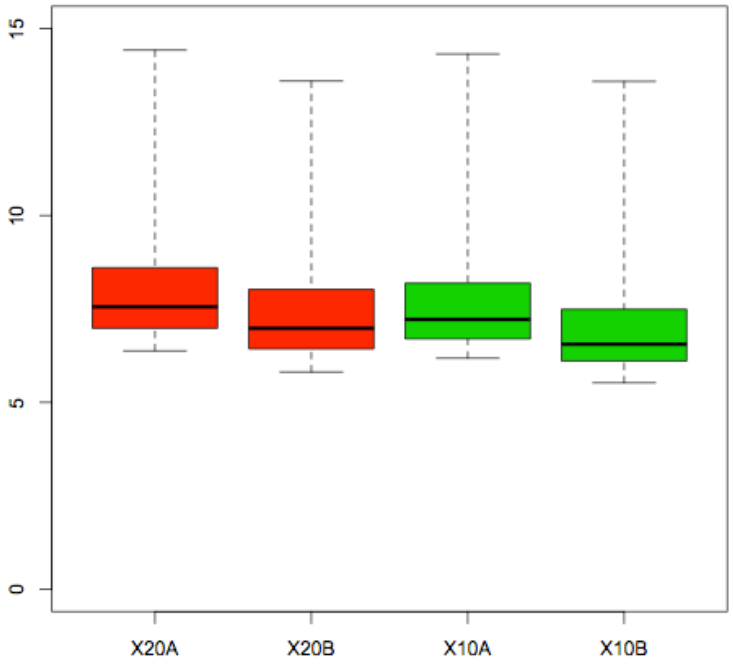
- This can be regarded as that each probe intensity is corrected based upon a weighted average of each of the grid's background values.

- Note that MAS 5.0 correct both PM and MM probes' background.

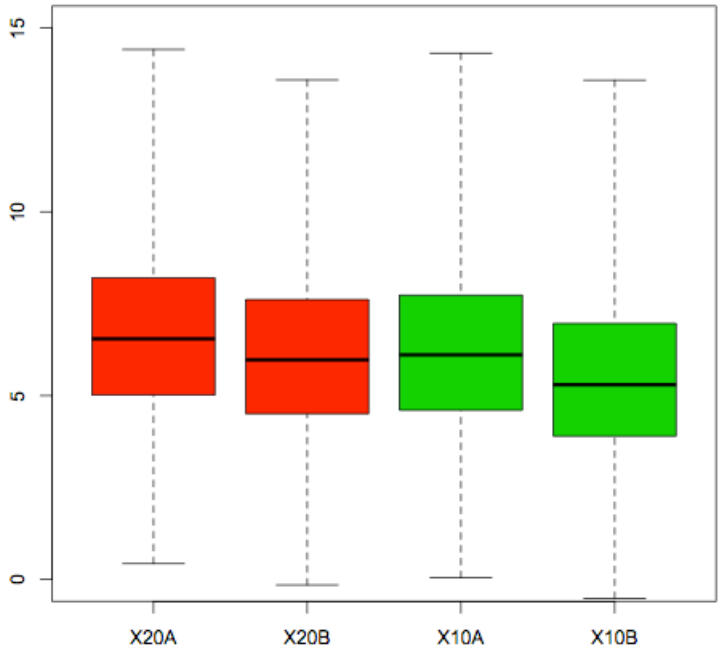
MAS 5.0: Background correction

- Under R:

  > a=bg.correct (Dilution, method="mas")



Before



After

Normalization

- Most approaches to normalizing expression levels assume that the overall distribution of RNA numbers doesn't change much between samples, and that most individual genes change very little across the conditions.

- If most genes are unchanged, then the mean expression intensity should be the same for each sample.
  - Scaling normalization
- An even stronger version of this idea is that the distributions of expression intensity must be similar.
  - Quantile normalization

MAS 5.0: Normalization

- The normalization algorithm in MAS is scaling.

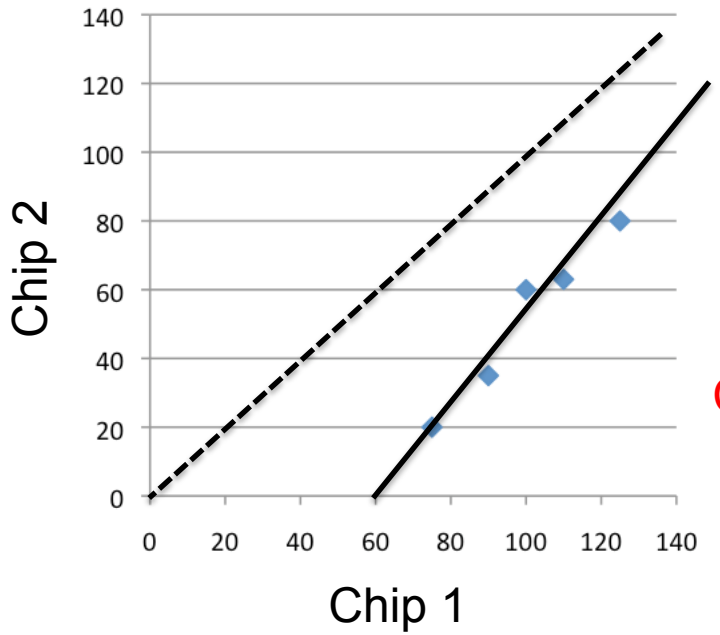
- A baseline array (chip) is chosen and all the other arrays (chips) are scaled to have the same mean (or median) intensity as this baseline array (chip).

- More specifically, a baseline array is selected and a linear regression term is fitted between each array and the baseline array. The fitted line is used as the normalizing relationship.

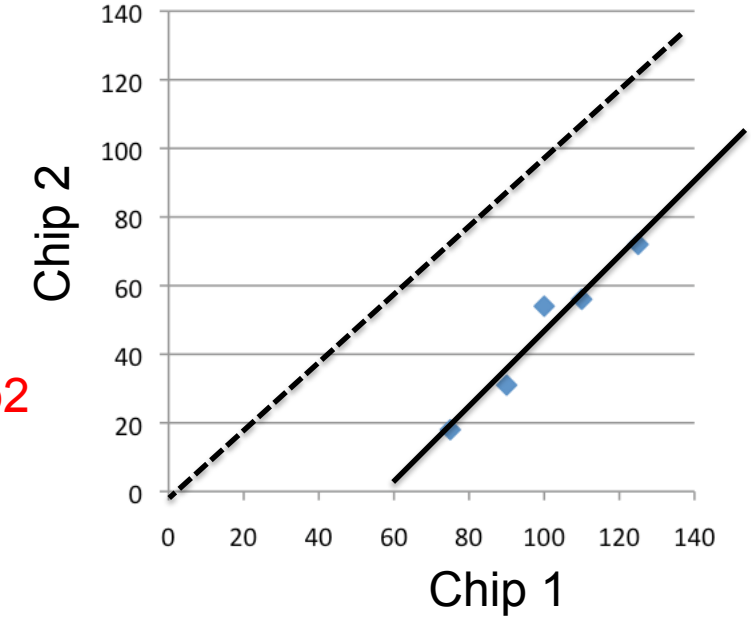
MAS 5.0: Scaling Normalization

	Chip 1	Chip 2	...	Chip n
Probe 1	**100**	**60**	**...**	**25**
Probe 2	**90**	**35**	**...**	**17**
Probe 3	**110**	**63**	**...**	**33**
Probe 4	**75**	**20**	**...**	**35**
Probe 5	**125**	**80**		**10**

Choose chip 1 as the baseline array, do the scaling normalization:



Chip1=k*Chip2



MAS 5.0: Trimmed Mean Scaling Normalization

Pick a column (chip) of expression matrix X to serve as baseline array, say column j.

Compute the trimmed mean of column j. Call this X_j

For i = 1 to n, $i \neq j$ **do**

Compute the trimmed mean of column (chip) i. Call this X_i

Compute $\beta_i = X_j / X_i$

Multiply elements (expression value) of column i by β_i

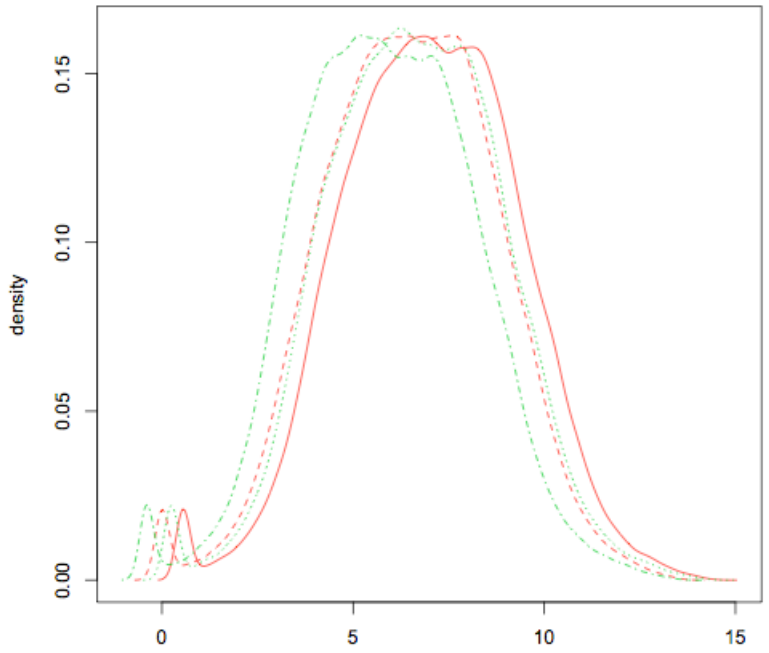
End For

- Trimmed mean: the highest and lowest 2% of the intensity are removed when computing the mean. This way, the mean value is less influenced by extreme (outlier) intensities.

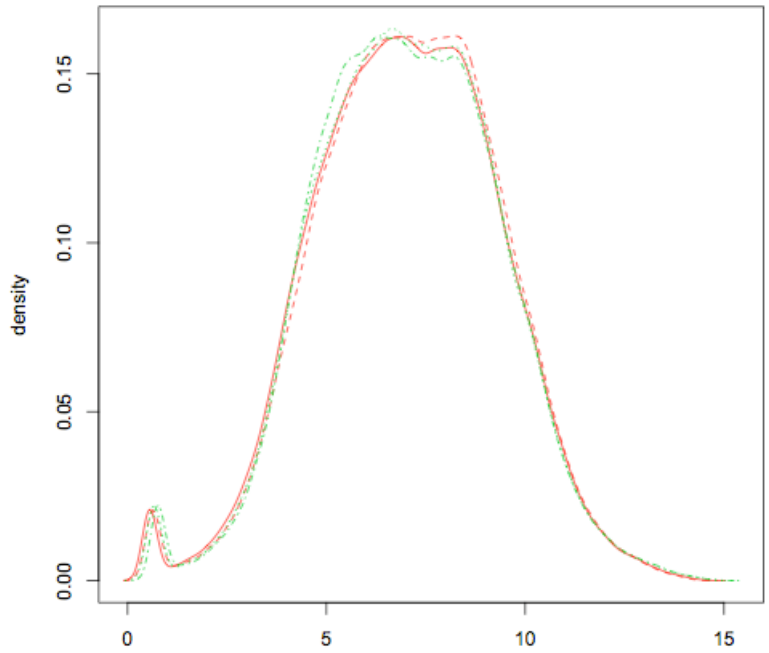
MAS 5.0: normalization

- Under R:

  > b=normalize (Dilution, method="constant")



Before



After

MAS 5.0: PM-MM correction

- The original purpose of designing MM probe is to provide measures of non-specific hybridization which inflate the true intensity of PM probe

- However, some MM values are bigger than the corresponding PM values which make the simple subtracting, (PM – MM), become negative.
  - We cannot use (PM-MM) as the corrected expression values, because it doesn't make sense as an expression value should not below zero…

- Negative (PM-MM) values will make the data interpretation and downstream analysis difficult.

MAS 5.0: PM-MM correction

- MAS introduces **Ideal Mismatch (IM)** computation to replace MM, so that it will remedy the negative impact of using raw MM values.

- The goal is to guaranteed the computed IM value to be smaller than the corresponding PM intensity so that it is usable.

- The principle of IM computation is to calculate a robust average of the log ratios of PM to MM for each probe pair in the probe-set k.

MAS 5.0: PM-MM correction

- For a given probe set k, the robust average of the log2 ratios of PM to MM for each probe pair in the probe-set, (specific background), can be calculated using One-step Tukey's biweight function as:

$$SB_k = turkey.biweight(\log_2(PM_1 / MM_1),...,\log_2(PM_n / MM_n))$$

- The use of one-step turkey's biweight function is to yield a robust weighted mean of log2 ratio of PM to MM that is relatively insensitive to outliers.

One step Turkey's Biweight

```
– Turkey.biweight <- function(x, c=5, epsilon=0.0001)
{
   m <- median (x)
   s <- median(abs(x-m))
   u <- (x-m)/(c*s + epsilon)
   w <- rep(0, length(x))
   i <- abs(u) <= 1
   w[i] <- ((1-u^2)^2)[i]
   t.bi <- sum(w*x)/sum(w)
   return(t.bi)
}
```

x is the vector of the log2 ratios of Pm to MM for each probe pair in a probe set

The probe pair is weighted more strongly if this probe pair's log ratio Is closer to the median value for a probe set.

Once the weight of each probe pair is determined, the mean of the weighted log2 ratio for a probe set is identified.

This mean value is output as specific background for subsequent IM calculation.

MAS 5.0: PM-MM correction

- For a given probe-set k containing n probe pairs, with each probe pair is indexed by i ($i=1,2,\ldots n$), the ideal mismatch, $IM_i^{(k)}$, can be calculated using obtained SB_k as:

$$IM_i^{(k)} = MM_i^k \quad \text{when} \quad MM_i^{(k)} < PM_i^k$$

$$IM_i^{(k)} = PM_i^k / (2^{SB_k}) \quad \text{when} \quad MM_i^{(k)} \geq PM_i^k \text{ and } SB_k > 0.03$$

$$IM_i^{(k)} = PM_i^k / (2^{10/(1+(0.03-SB_k)/10)}) \quad \text{when} \quad MM_i^{(k)} \geq PM_i^k \text{ and } SB_k \leq 0.03$$

- Now the (PM-MM) correction can be done as **(PM-IM)**.

MAS 5.0: Summarization

- After background correction, normalization and PM-IM correction, we obtain a vector of "processed" (background corrected, normalized, and IM corrected) probe values for a given probeset.

- We need to summarize the vector of probe values into one expression value of the stuided probeset.

- We can again use one-step turkey's biweight function to yield a robust weighted mean of probe value, which is relatively insensitive to outliers, to represent the probeset expression value.

expresso()

1. Background correction: weighted average of grid background
2. Normalization: trimmed mean scaling
3. PM-MM correction (optional): Ideal mismatch
4. Summarization: one step Tukey's Biweight function

```
>  set <- expresso(Dilution, bgcorrect.method = "mas",
normalize.method = "constant", pmcorrect.method = "mas",
summary.method = "mas")
```

MAS 5.0

```
> dim(exprs(Dilution))
409600     4
> expression <-mas5(Dilution)
> dim(exprs(expression))
12625     4
> write.exprs(expression, file="mymas5data.txt")
```

MAS5: Summary

- Good
  - Usable with single chips (though replicated preferable)
  - Gives a p-value for expression data
- Bad:
  - Lots of fudge factors in the algorithm
  - Not *exactly* reproducible based upon documentation (source now available)
- Misc
  - Most commonly used processing method for Affy chips
  - Highly dependent on Mismatch probes

Preprocess methods in Bioconductor

- MAS (Microarray Analysis Suite) 5.0
- RMA (Robust Multi-array Average)
- These two are the most popular methods for preprocessing Affymetrix data. Each method consists of different algorithm at each step of preprocessing.

RMA

- Robust Multichip Analysis
- Used with groups of chips (>3), more chips are better
- Assumes all chips have same background, distribution of values.

1. Background correction: RMA convolution
2. Normalization: quantile normalization
3. PM-MM correction (optional): none
4. Summarization: Fitting probe level model

RMA

- One notable point of RMA is that it only uses the PM probes. Hence the PM –MM correction is not applied.
- *This means that **half** the probes on the chip are excluded, yet it still gives good results!*
- Ignoring MM decreases accuracy, increases precision.

Model

- MAS:

  Measured Value = N + P + S

  – N = Noise

  – P = Probe effects (non-specific hybridization)

  – S = Signal

- RMA:

  Measured Value = B + S

  – B = Background

  – S = Signal

RMA: Background correction

- Only PM values are corrected, array by array, using a global model for the distribution of probe intensities.
- The observed raw intensity of PM probes are modeled as the sum of a Gaussian component (with mean u and variance σ^2) for background intensity and an exponential component (with mean α) for true expression intensity.

$$PM_i^{(K)} = S_i^{(k)} + B_i^{(K)} \quad \text{where}$$

$S_i^{(k)} \sim Exp(\alpha)$: true expression intensity

$B_i^{(k)} \sim N(u, \sigma^2)$: background intensity

- For each chip, we can fit the model to the observed PM raw intensity to estimate u , σ , and α

RMA: Background correction

- Once the u, σ, and α are estimated, they can be used in the equation below to obtain the background-adjusted intensity for each PM probe (raw intensity "y").

$$E(S = s \mid Y = y) = A + B \frac{f(\frac{A}{B}) - f(\frac{y - A}{B})}{F(\frac{A}{B}) + F(\frac{y - A}{B}) - 1}$$

where $A = s - u - \sigma^2 \alpha$ and $B = \sigma$

also, $f \sim N(0,1)$ standard normal density function,

$F \sim N(0,1)$ standard normal distribution function

RMA: background correction

- Under R:

  > bg.correct (Dilution, method="rma")

RMA: Normalization

- Most approaches to normalizing expression levels assume that the overall distribution of RNA numbers does not change much between samples, and that most individual genes change very little across the conditions.

- If most genes are unchanged, then the mean expression intensity should be the same for each sample.
  - Scaling normalization
- An even stronger version of this idea is that the distributions of expression intensity must be similar.
  - Quantile normalization

RMA: Normalization

- The normalization algorithm in RMA is quantile normalization. The goal of quantile normalization is to impose the same empirical distribution of intensities to each array.

- The quantile normalization is a specific case of the transformation

$$x_i^{'} = F^{-1}[G(x_i)]$$

where G is estimated by the empirical distribution of each array (x_i) and F is the empirical distribution of the averaged sample quantiles.

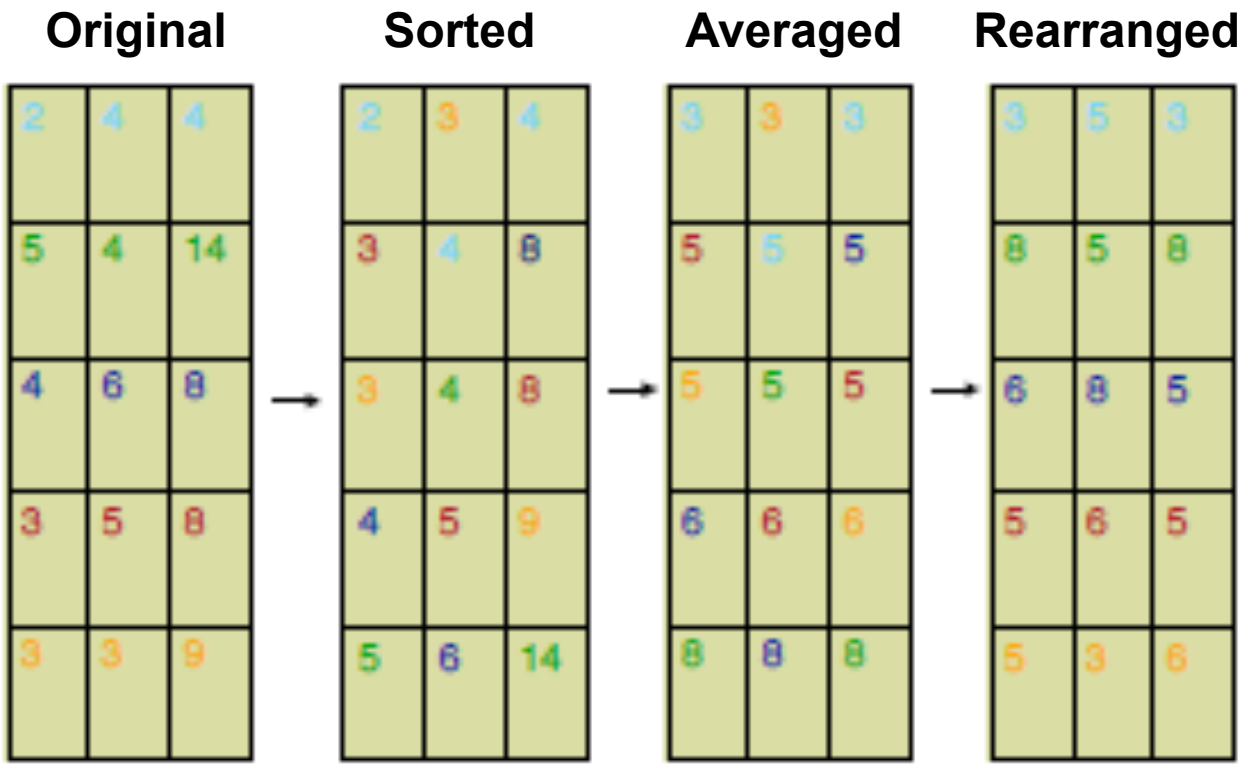
RMA: Quantile Normalization – How?

- Step 1: After background correction, find the smallest corrected intensity on each column (chip)
- Step2: Compute the average values from step 1
- Step 3: Replace each value in step 1 with the average value computed from step 2

  Repeat steps 1 through 3 for the second smallest values, third smallest values, …, largest values.

  Rearrange each column (chip) to have the same ordering as the corresponding input matrix.

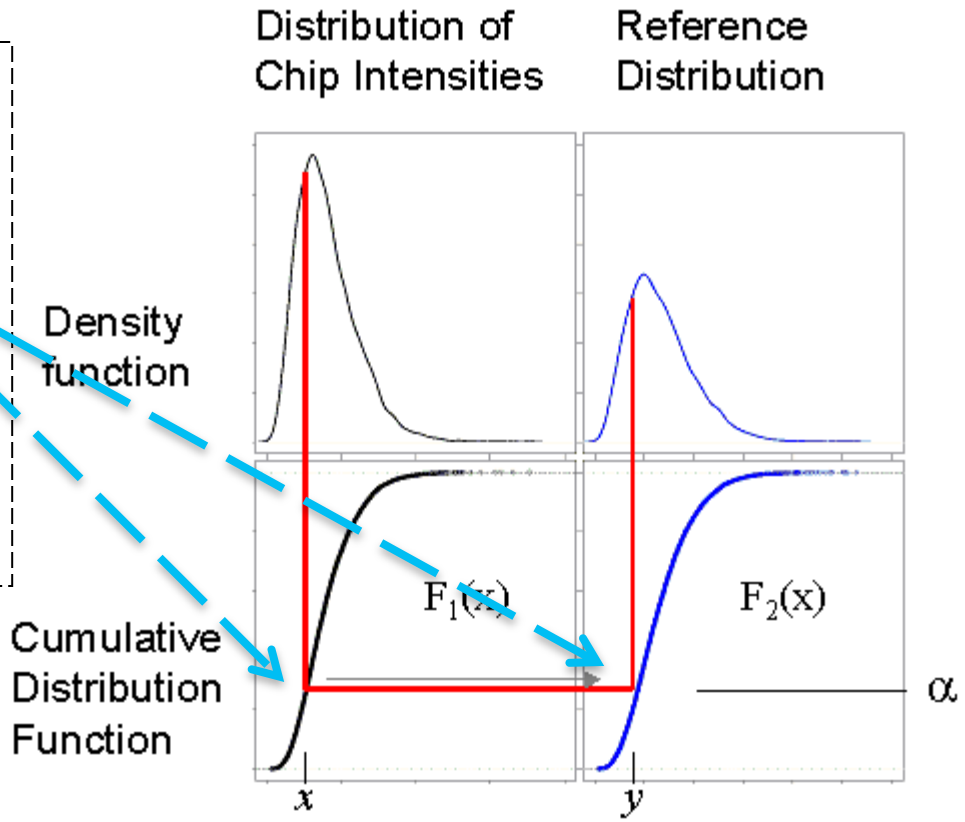
RMA: Normalization



- 5 genes (A, B, C, D, E), 3 chips
- The final rearrangement step is to ensure we are comparing the expression values of the same gene on different chips.

RMA: Normalization

The value ***x***, which is the α-th quantile of all probes on chip 1, is mapped to the value **y**, which is the α quantile of the reference distribution F_2.



- To normalize each chip we compute for each value, the quantile of that value in the distribution of probe intensities; we then transform the original value to that quantile's value on the reference chip, which is the pooled distribution of probes on all chips. The quantile normalization method transforms the distribution of intensities from one distribution to another.

RMA: normalization

- Under R:

  > normalize (Dilution, method="quantiles")

RMA: summarization

We need to combine processed intensities of PM probes to generate an overall expression estimate for the probe set. We do this by fitting a probe level model.

❑Probe level Model

$$\log_2(y_{ij}) = \theta_i + \phi_{ji} + \varepsilon_{ij}$$

Y_{ij} : Background-corrected, quantile normalized intensity of probe *j* in sample (chip) *i*

θ_i : Expression value for the probeset in sample *i*

ϕ_{ij} : probe specific effect of probe *j* *in probeset i*

$\varepsilon_{ij} \sim N(0, \sigma^2)$: residual, in log2 scale

After model fitting, we obtain the estimated expression value of each probe set.

RMA

1. Background correction: RMA convolution
2. Normalization: quantile normalization
3. PM-MM correction (optional): none
4. Summarization: Fitting probe level model

- Under R

> set <- expresso(Dilution, bgcorrect.method = “rma", normalize.method = "quantiles", pmcorrect.method = “pmonly", summary.method =“medianpolish")

> expression<-rma(Dilution)