Preprocessing Microarray Data: Normalization
Comparable numbers, on a scale and format appropriate to the questions and the data analysis tools one intends to use. Possibly no missing values, possibly a reduced set of genes.
**Normalization:**

- Ensure comparability between two sets of expression measurements; experimental condition vs control.

- Mitigate unwanted (non-experimental) variation in expression measurements.

- Important generalization: ensure comparability among several sets of measurements; multiple experimental conditions.
For example: **List of possible sources of unwanted variation in spotted arrays**

**Preparing the samples:**
- mRNA preparation
- Reverse transcription to cDNA
- Dye labeling

**Spotting the chips:**
- PCR amplification
- Pin geometry and surface features
- Amount of cDNA transported by pins
- Amount of cDNA fixated on slide

**Hybridization process:**
- Hybridization parameters (temperature, time, amount of sample)
- Spatial dis-homogeneity of hybridization on the slide
- Non-specific hybridization

**Image production and processing:**
- Non-linear transmission, saturation effects, variations in spot shape
- Global background shining, and local overshining from neighboring spots

Et cetera (source: Schuchhardt *et al.* 2000). Many of these also apply for Affy chips.

Different dyes can also have different decay and optical properties…
Normalization involves creating a correction for one or both sets of numbers. What “genes” (spots/probes) should be used?

**“Bulk” normalization**: use all the “genes”. Underlying rationale:
- only a small share of the “genes” show systematic experimental variation in expression, and/or
- changes in expression tend to compensate as to not sizeably affect the normalizing correction.

Is this reasonable? Not in all experiments (especially not in chips designed for specific purposes; smaller set of genes likely to change)

**“Control-based” normalization**: use only “genes” that ought not to show systematic experimental variation in expression (housekeeping genes, spiked controls)

Is this reasonable? Are controls a reliable tool for normalization? Spiked may carry systematic errors; housekeeping may have very high variability. Also, there may be few controls, and they may be placed at non-random locations on the chip.
1. Computing scaling factors:

Relative “activity” on two colors, or on two chips (normalizing total equivalent to normalizing mean). On log-ratio scale (e.g. base 2, fold-change) we are relocating the distribution to be centered at 0.

\[
\frac{\text{Exper}(i)}{\text{Contr}(i)} \times \rho
\]

\[
\rho = \frac{K_{\text{Contr}}}{K_{\text{Exper}}}
\]

e.g. \( K_w = \sum_i W(i), w = \text{Exper, Contr} \)

\[
\log\left(\frac{\text{Exper}(i)}{\text{Contr}(i)} \times \rho\right) = \log\left(\frac{\text{Exper}(i)}{\text{Contr}(i)}\right) + \log(\rho)
\]

Normalizing median follows a similar logic, but is robust to outliers (fits better with the rationale underlying bulk normalization; some genes do change!)
2. Computing non-parametric trends:

The location of log-ratios may depend on the (average) intensity; relocate at each average intensity level, that is, subtract a trend estimated non-parametrically.

\[ M_i = \log \left( \frac{\text{Exper}(i)}{\text{Contr}(i)} \right) \quad A_i = \frac{1}{2} \log(\text{Exper}(i) \times \text{Contr}(i)) \]

\[ \tilde{M}_i = M_i - \varphi(A_i) \]

\[ i \in s , \tilde{M}_i = M_i - \varphi_s(A_i) \quad s = 1...S \]

If obvious groupings of the “genes” exists (e.g. sectors of the chip; spots produced by the same pin, known to potentially create unwanted variation) can compute group-specific corrections.

From Yang et al. (2001, 2002)
From Yang et al. (2001, 2002)

How normalization “regularizes” the distribution of log-ratios.
Statistics: Non-parametric estimation of trends

**Lowess, or Loess = Locally (weighted) least squares**

References:
• Textbooks on Regression methods.

Create a continuous curve capturing the “systematic” relationship between Y (response, e.g. M) and X (predictor, e.g. A), without introducing a parametric model. A method to create a “smooth” of the scatter-plot of Y vs X, implemented in most statistical packages (including S+ and R)

“Parameters” of the algorithm:
• Fraction of data in each local neighborhood (smoothing parameter)
• Degree of locally fitted polynomial (1=linear, 2=quadratic)
• Weight function for the least square fit
• Number of iterative weighted least square fits (in some packages)
For instance:
Smoothing fraction = 0.2
Degree = 1
# iterations = 2

Weighted least squares linear fit

\[ \min_{\alpha, \beta} \sum_{i \in N(x)} w_i (y_i - (\alpha + \beta x_i))^2 \]

Weight function

\[ w_i = \left( 1 - \frac{d(x_i, x)}{\max_{j \in N(x)} d(x_j, x)} \right)^3 \]

F. Chiaromonte, Sp 07
In R (using the chicken_toy data):

```r
> plot(chicken_toy[,"x"],chicken_toy[,"y"],xlab="log ins ratio",
      ylab="log length ratio")
> lines(lowess(chicken_toy[,"x"],chicken_toy[,"y"],f=0.2,iter=2),lwd=2)
> #implements a degree 1 lowess smooth for y vs one x
> #added as a ‘line’ to the plot, lwd means line width.
> out_lowess <-
      lowess(chicken_toy[,"x"],chicken_toy[,"y"],f=0.2,iter=2)
> out_lowess
$x
[1] 0.34469 0.37752 0.40617 . . .
$y
[1] 0.1170733 0.1414099 0.1654462 . . .
> #the abscisae and ordinates stored for
> #the computed curve (not an equation)
> #see also loess(), more sophisticated.
```

F. Chiaromonte, Sp 07
Fraction controls degree of smoothing:
Quadratic (vs linear) fit allows to better capture “bends”:
3. “Matching” experimental and control distributions:
Reduce both distributions (original or log-scale; equivalent) to a common shape, obtained averaging their $p$ empirical quantiles

$\text{Exper}_i(1) = \text{Contr}_i(1)$
$\text{Exper}_i(2) = \text{Contr}_i(2)$
$\text{Exper}_i(3) = \text{Contr}_i(3)$
$\text{Exper}_i(h) = \text{Contr}_i(h)$

$\text{Exper}_{i(p-1)} = \text{Contr}_{i(p-1)}$
$\text{Exper}_{i(p)} = \text{Contr}_{i(p)}$

$M_i = \log \left( \frac{\text{Exper}(i)}{\text{Contr}(i)} \right) \quad \leftarrow \quad \tilde{M}_i = \log \left( \frac{q_{ave}(\text{rank}_{\text{Exper}}(i))}{q_{ave}(\text{rank}_{\text{Contr}}(i))} \right)$

* Unless $i$ changes rank, normalized log-ratio is 0 (robust; give up signal information… may be problematic for genes in the right tails – few genes, rank changes less likely).

From Bolstad et al. (2003)
Yang et al. (2001, 2002; spotted arrays) and Bolstad et al. (2003; Affy arrays) give much more detail on these procedures and others, and explain how to generalize normalization to many arrays.

Bolstad et al. advocate
- complete methods with no baseline, and
- normalization on individual probe PM values, not probe cell summaries (note MM values are not used); summaries produced after normalization – this allows to recover some “signal” discrimination for genes in the right tail (see *).