Lecture 4  Processing the scanned image

Processing Steps:
- Microarrays are scanned  =>  Raw Image, array of pixels recording the intensity level of the fluorescence at each pixel.
  - The image must be converted into spot intensities for analysis.
- Spot intensity: Reflects the amount of labeled probe that hybridized to it.
- Run a series of quality checks on the data.
- The spot intensity is adjusted for background effects.

Converting the scanned image to the spotted image

Gridding: assigning coordinates to the center of each spot
Segmentation: signal is separated from the background
Quantification: spot is assigned an intensity value

Gridding
Arraying process: spots in a perfect rectangular grid
verlay an appropriately-sized grid on the microarray and align it.
In practice the grid tends to be a slightly deformed
need some adjustment. manually shifting the rows and columns

A more rigorous method: locally smooth the image using (Gaussian kernel)
Use the mode of the smoothed region as the spot center.
Modify the grid to minimize the distance from the spot centers to the centers of the rectangular or square regions containing each spot.
Segmentation

- **fixed circle segmentation**: by fitting a circle with a constant diameter to all the spots
- **adaptive circle segmentation**: fitting circles with different diameters to different spots
- Neither of these segmentation methods works particularly well.
- **seeded region growing algorithm** (Computer vision)
  
  (i) **Seed specification**.
  
  seeds for signal be the estimated spot locations from gridding .
  seeds for background could the midpoints.
  
  (ii) **Region Growing**.
  
  Candidate pixel: next to an allocated pixel
  
  At each step, among the pixel candidates the pixel that is the closest to the average intensity of the corresponding region is assigned to that region.
  
  (iii) **Stopping Rule**. The process continues until all the pixels have been allocated to one of the regions.

- **histogram segmentation**.
  
  A mask is placed over each spot.
  
  Graph histogram of pixel intensities within the mask
  
  Determine a threshold value
  
  Each pixel within the mask is then classified as signal or background

Quantification

- Average intensity of the pixels, other spot–related statistics.
- The **spot intensity**. Array \( \{X_{rc}\} \) of spot intensities: here
  
  \( X_{rc} \) Average intensity of the spot \( r \)th row and \( c \)th column of the array
The mean or median or trimmed mean or biweight or mode

- **Spot background**: average intensity of pixels designated as background. \( \{B_{rc}\} \)
- **Signal cv** and **background cv**: quality of the spot.
- **Circularity** is \( 4\pi \) times the area of the spot divided by its perimeter\(^2 \) the closer this is to 1, the more circular the spot.

**The spotted image**

Once the spotted image and related statistics are obtained it is necessary to evaluate the quality of the spot observations by following a quality control procedure that is comprised of the following steps.

**Visualization**

- Visual inspection: evaluate the quality of the spotted image.
- Use image plot. Searched for non-random patterns
- If this is not the case the image is passed to the next step.

**Quality issues**

- spatial patterns in a spotted image are not visible in the image graph because the variation maybe small enough not show in the color scale or the color scale maybe not be sensitive enough to show the pattern.
- Automatic methods are useful in order to process large number of microarray images without need for individual visual inspection.

**Finding area problems**

(i) Large spots covering a good part of the area of the background image. These spots show higher or lower intensities than the rest of the image.
(ii) Vertical or horizontal strips on the background image that show higher or lower intensities.

(iii) Diagonal strips again showing higher or lower background intensities.

(iv) A ramp in the background intensities going across the array.

(v) Bleeding in the spotted image showing sequences of consecutive.

- **Algorithm:**

(i) Split the image into high intensity and low intensity spots. This is a binary split similar to the ones performed by a regression tree algorithm at a single node. CART (Breiman & Friedman (1978) does it by finding the cutoff that minimizes the within group sum of squares. However this split is not robust against outliers. A simple alternative that is resistant to outliers is to set the cutoff to the mid point between to quantiles (say 5% and 95% quantiles). Then define the response at the spot at row $r$ and column $c$ is $Y_{rc}=1$ for high intensities spots and $Y_{rc}=0$ for the rest.

(ii) Fit a quadratic discriminant function to the binary response $\{Y_{rc}\}$ using the spot coordinates $(r,c)$ on the microarray as predictors. Suppose that $Z_{rc}$ are the predicted responses by the discriminant function. In order to asses the good ness of the fit we calculate the proportion $p$ of correctly predicted spots, that is the proportion of spots with $Y=Z$. The null distribution of the $p$ statistic can be simulated by a simulation the images. To do this generate 300 images by random permutations of the spot intensities and calculate the value of $p$ for each image, resulting in the set \{p_1,\ldots,p_{300}\}. Estimate the p–value by the proportion of $p_i$ greater than the observed $p$. This p–value measures the performance of the quadratic discriminant analysis and it is used to determine the overall quality of the microarray.
The above procedure is summarized in an image quality graph such as the one in Figure 4.2. The figure contains a central panel showing a color image and four graphs on the right side of the figure.

(i) The main panel displays an image representing the background intensities that are being analyzed. The color scale corresponding to the main panel is shown on a narrow horizontal strip below the main panel.

(ii) The right side of Figure 4.1 shows a column of four graphs:

- The two graphs at the top of the right side show the average profiles of the rows and columns of the main panel respectively.
- The third and fourth graphs show the image graphs of the arrays \( \{Y_{rc}\} \) and \( \{Z_{rc}\} \) respectively.

**Outlier Removal:**

Another way of seeing whether any part of an array is emitting higher signals compared to the rest of the array is to check whether the “outliers” (as obtained by ----) are randomly scattered throughout the array or whether they are generally clustered together or distributed according to some pattern. In the algorithm in the previous section we could use this as (ii). This could be checked using a simple test of complete spatial randomness, such as that proposed for a problem in ecology by Clark and Evans (1954).

Suppose that there are \( n \) spots on the array, \( r \) of which are “outliers”. For the \( i \)th “outlier”, let \( d_i \) be the distance to the “outlier” closest to it, so that

\[
\bar{d} = \frac{1}{n} \sum_{i=1}^{n} d_i
\]

is the average nearest neighbor distance between the “outliers”. The test statistic for complete spatial randomness is \( \bar{d} \) or its standardized form

\[
T_{CSR} = \frac{\bar{d} - 1/(2\sqrt{\rho})}{\sqrt{(4 - \pi)/4n\pi\rho}}
\]
which has a standard normal distribution under complete spatial randomness. The parameter can be estimated as \( \hat{\rho} = r/n \). Note that two aspects of the data are being ignored in doing this test: non-independence of some nearest neighbor distances and edge effects. More complex tests that adjust for these aspects of the data have been developed in the spatial data analysis field.

This method would produce good results for arrays where the outlier appear in small clusters, such as the bleeding spots case (v) in the list in section 4.2.2. If the spot covers a large part of the array then it would help to smooth the image and it will be harder to detect.

Quality control

(i) Image quality graph, to detect specific problems with the microarray.
(ii) Draw a boxplot of the sequence of arrays that have been observed up to this point and check if there are any changes from the previous arrays to the current one.

Adjusting for background

Global background adjustment: average intensity of all background pixels
Spot background adjustment: subtracted from the spot intensity value
Smoothed background adjustment:
Zonal background adjustment: Affymetrix uses a variation of smoothed background adjustment called zonal background adjustment for its oligonucleotide microarrays. This approach can also be used with cDNA microarrays.

First, the microarray is split up into \( K \) rectangular zones, \( Z_k, k=1,.., K \) (Affymetrix uses the default \( K=16 \)). For each zone, a low percentile of the spot intensities, \( S_{lg} \), is chosen as the background, \( BZ_k \), for that zone (Affymetrix uses the second percentile as its default; this is the value such
that 98 percent of spot intensity values are larger than it and 2 percent are
smaller).

- A smoothed version of these background estimates are used instead.
- $d_{gk} =$ the distance between the $g$th spot and the center of $Z_k$
- weight: $w_{gk} = 1/d_{gk}^2$. $d_0$, is added to the denominator to ensure that it
will never be zero, so that $w_{gk} = 1/d_{gk}^2 + d_0$.

$$BSZ_g = \frac{\sum_{k=1}^{K} w_{gk} BZ_k}{\sum_{k=1}^{K} w_{gk}}.$$ 

**Adjusting for the estimated background**

$Al_g = Sl_g - Bl_g.$

*Background adjusted thresholded spot intensity value, $Al_g,$*

$Al_g = \max ( Sl_g - Bl_g, T )$

Affymetrix uses the following approach. The standard deviation, $NZ_k,$ of
the of all spot intensities lower than $BZ_k$ is used as an estimate of the
variability of the background for each zone. Calculate a weighted
variability estimate: $BNZ_g = \frac{\sum_{k=1}^{K} w_{gk} NZ_k}{\sum_{k=1}^{K} w_{gk}}.$ Then a threshold and a floor are set at
some fraction of the local noise value, so that no value is adjusted below
that threshold. That is, for a cell intensity $I'(x, y)$ at chip coordinates $(x, y),$
we compute an adjusted intensity.

$Al_g = \max ( Sl_g - Bl_g, NoiseFrac \cdot n(x, y) )$

where $I'(x, y) = \max (I'(x, y), 0.5)$

$NoiseFrac$ is the selected fraction of the global background variation.

$(defaultNoiseFrac = 0.5)$
Expression level calculation for oligonucleotide microarrays

A probe set of twenty or so oligonucleotides, the perfect match probes, along with a set of paired mismatch probes. The expression level for a gene, which Affymetrix calls its Signal, is calculated from the combined background adjusted PM and MM values of the probe set for the gene.

In Affymetrix’s MAS 5.0 software, Signal is calculated as follows:
1. Cell intensities are preprocessed for global background.
2. An ideal mismatch value is calculated and subtracted to adjust the PM intensity.
3. The adjusted PM intensities are log-transformed to stabilize the variance.
4. The biweight estimator is used to provide a robust mean of the resulting values. Signal is output as the antilog of the resulting value.
5. Finally, Signal is scaled using a trimmed mean.

The reason for including a MM probe is to provide a value that comprises most of the background cross-hybridization and stray signal affecting the PM probe. It also contains a portion of the true target signal.

- If the MM value is larger than the PM value, it is a physically impossible estimate for the amount of stray signal in the PM intensity. Instead, an idealized value can be estimated based on our knowledge of the whole probe set or on the behavior of probes in general.
- To calculate a specific background ratio representative for the probe set, we use the one-step biweight algorithm ($T_{bi}$), which is described in Appendix I. We find a typical log ratio of PM to MM that is simply an estimate of the difference of log intensities for a selected probe set. The biweight specific background ($SB$) for probe pair $j$ in probe set $i$ is:

$$SB_i = T_{bi} (\log_2(\text{PM}_{i,j}) - \log_2(\text{MM}_{i,j}) : j = 1, \ldots, n_i)$$

If $SB_i$ is large, then the values from the probe set are generally reliable, and we can use $SB_i$ to construct the ideal mismatch $IM$ for a probe pair if needed. If $SB_i$ is small ($SB_i \leq contrast \tau$), we smoothly degrade to use more of the PM value as the ideal mismatch. The three cases of determining ideal mismatch $IM$ for probe pair $j$ in probe set $i$ are described in the following formula:

$$Scale \tau (\tauau)$$

is the cutoff that describes the variability of the probe pairs in the probe set.
The first case where the mismatch value provides a probe-specific estimate of stray signal is the best situation. In the second case, the estimate is not probe-specific, but at least provides information specific to the probe set. The third case involves the least informative estimate, based only weakly on probe-set specific data.

Given the ideal mismatch value, the formula for the probe value (PV) is fairly simple. To guarantee numerical stability, we use the formula:

$$V_{i,j} = \max(PM_{i,j} - IM_{i,j}, \delta) \text{ default } \delta = 2^{-20}$$

Now we calculate the probe value PV for every probe pair j in probeset i, n is the number of probe pairs in the probeset.

$$PV_{i,j} = \log_2 (V_{i,j}), j = 1, \ldots, n_i$$

We then compute the absolute expression value for probe set i as the one-step biweight estimate (see Appendix I) of the in adjusted probe values:

$$SignalLogValue_i = T_{bi}(PV_{i,1}, \ldots, PV_{i,n_i})$$

Note: the scaling (sf) and normalization factors (nf) computed in this section are reported by the software.

If the algorithm settings indicate scaling all probes sets or selected probe sets to a target we calculate a scaling factor (sf)

$$sf = \frac{Sc}{TrimMean(2^{SignalLogValue_i}, 0.02, 0.98)}$$

where Sc is the target signal (default Sc = 500) and the SignalLogValues in the SignalLogValues, set are the probe sets indicated in the algorithm settings. The trimming function here takes the average value of all observations after
removing the values in the lowest 2% of observations and removing those values in the upper 2% of observations. If the algorithm settings indicate user defined scaling, then \( sf = \) user defined value.

The reported value of probe set \( i \) is:

\[
\text{ReportedValue}(i) = nf \times sf \times 2^{(\text{signalLogValue},)}
\]

where \( nf = 1 \) for absolute analysis and is computed as follows for a comparison analysis.

If the algorithm settings indicate user defined normalization, then \( nf = \) user specified normalization.

Otherwise, the algorithm settings either indicate normalizing all or selected probe sets:

\[
nf = \frac{\text{TrimMean}(SPV_{b}, 0.02, 0.98)}{\text{TrimMean}(SPV_{e}, 0.02, 0.98)}
\]

where \( SPV_{b}[i] \) is the baseline signal, and \( SPV_{e[i]} \) is the experiment signal (scaled-only) and \( i \) defines the probe sets selected by the user.

This is reported as Signal.

Since comparison analyses are done on matched probe pairs, the individual probe pair values are also modified by this scaling factor. The scaled probe value \( SPV \) is

\[
SPV_{i,j} = PV_{i,j} + \log_2 (nf \times sf)
\]

These values are used in computing the log-ratio in comparison analysis.

I

Affymetrix report validating their procedure on several real and simulated datasets, some good some bad.

**Supplementary reading**