An (Opinionated) Guide To Microarray Data Analysis

by

Mark Reimers

Karolinska Institute
Dept. of Biosciences

The aim of these pages is to set out the author's opinions about the best ways to deal with the most common issues in microarray data analysis. The opinions are based on experience with dozens of collaborating lab scientists, and discussions with microarray statisticians. Contributions, disputes, and opinions are welcome (and may be posted!). These pages are intended primarily for people working in a microarray facility. However I hope that statisticians and programmers will also find interesting material here.

This guide is organized as follows

General Issues

Experimental design

Distributions and transforms

Approaches to Normalization

Spotted Array Issues (two-color competitively hybridized microarrays)

Image analysis for spotted arrays (not done)

Quality control of spotted arrays

Normalization of spotted arrays

Affymetrix Issues

Quality control

Normalization

Estimates of Abundance (methods for combining data from multiple probes to get single estimates)

Downstream Analysis

Graphics

Clustering

Statistical Significance

For descriptions of the technology see TECH LINKS

Design of Microarray Experiments

How many replicates is enough?

Should you pool samples?

What is a good designs for a cDNA experiment with many samples?

The design of scientific experiments is an art of balancing considerations: skill, cost, equipment, and accuracy. For a given question, there won't be one 'right' design: you may choose different designs for the same scientific question in different contexts. Some important practical issues for microarray experiments are:
1.how your experiment fits into a larger plan,
2.how often your hybridizations fail, and
3.how noisy your measures are.
If your goal is to make a series of experiments and be able to make comparisons, ensure that the designs and conditions are similar. Conditions such as RNA preservation medium, the protocols of hybridization, and even environmental conditions, can introduce systematic biases comparable in size to the biological differences you wish to detect. Taking a great deal of care to standardize conditions will pay off in much higher discovery rates. To do a series of two-color hybridizations, you want to prepare enough common reference to serve for all experiments.
Chip failures are frequent, some of the more efficient designs will lose much information if a single hybridization fails; you won't want to use those designs if you can't set aside samples to be re-hybridized quickly to chips from the same batch.
Although replicates are costly, you can only estimate the variability by replicates. To be confident in your results, you should find out the variability of measures based on the chips and protocol you are using, or better, estimate this yourself.

Designs for Two-Color Arrays

The Reference Design is when each experimental sample is hybridized against a common reference sample. The Reference Design
- extends easily to other experiments, if the common reference is preserved;
- is robust to multiple chip failures, and makes it easier to replace failed chips;
- reduces incidence of laboratory mistakes, because each sample is handled the same way.

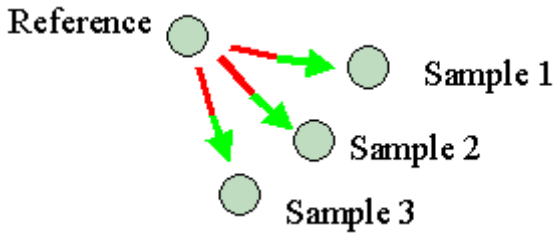


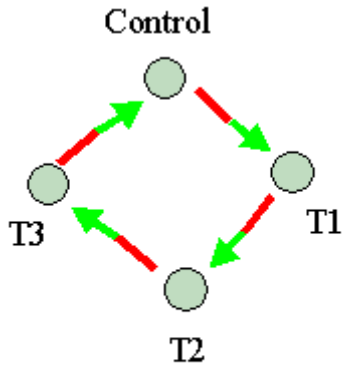
Figure 1. A reference design: the red and green arrows represent chips.

The reference sample is used in many chips, therefore the reference mRNA needs to be abundant. When comparing treatment versus control samples the most natural reference is the wild type or the biological controls, which are often the most abundant. However design becomes more perplexing if the study involves several samples, and the aim is to compare each against all others; to do a separate chip for all possible contrasts will take too many chips. An alternative is a common reference obtained by mixing all samples. This enables samples to be compared with each other, at the cost of

making indirect comparisons, which are less reliable. A mixed reference sample reduces the number of extreme gene ratios on each chip, which gives more accurate estimates since, extreme ratios have typically large errors. Some labs take this further and use a 'universal reference': a pool of mRNA derived from several standard different cell lines. Using a universal reference enables them to compare results for all their experiments.

One complication in two-color arrays is that the two don't get taken up equally well, so that the amount of label per amount of RNA differs (dye bias). An early proposal to compensate for dye bias was to make duplicate hybridizations with the same samples using the opposite labeling scheme. For example, to compare two samples: A & B, make two arrays (or an even number), and hybridize them as follows:
Array 1: **A** vs **B** ; Array 2: **B** .vs. **A**
The intent was to compensate dye bias by averaging ratios from dye-swapped hybridizations. However dye bias is not consistent, and in practice the ratios in dye-swap experiments don't precisely compensate each other. Normalization methods such as lowess give more consistent results, although dye-swapping makes it easier to compensate for dye-bias. However the dye-swap is the basis for most other efficient designs: the general principles of a good two-color design are that
i) it should be balanced: every sample appears equally often in red and green;;
ii) the samples whose ratios are most interesting should appear on the same chips most often.

From a theoretical perspective, for comparing a number of samples of equal interest and high quality, a design that utilizes a large number of direct sample-to-sample comparisons is most accurate for the cost. The simplest of these is a 'loop' design: each sample is hybridized to each of two different samples in two different dye orientations. This design results in half the variance per estimate, because each sample occurs twice, rather than once; at the cost of only one more chip. The drawback is that if one chip fails, or is of poor quality, then the error variance for all estimates is doubled.



'Loop' Design

Figure 2. A loop design: arrows represent chips with samples labelled as indicated.
There are many efficient and robust designs based on 'round-robin' style contrasts where each sample is hybridized to a specific subset of all the others, in a balanced fashion. These designs are really most appropriate where all samples are equally important, and the experiment is not part of a longer series.

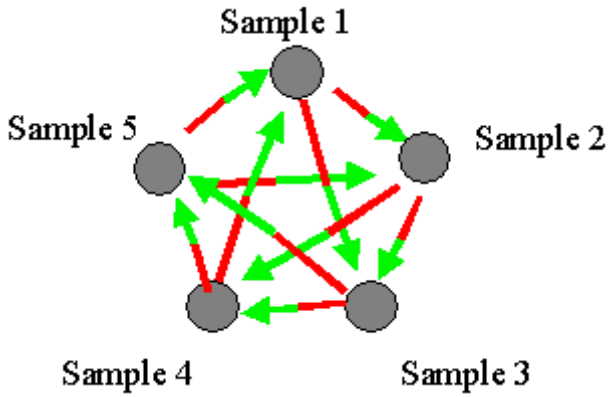


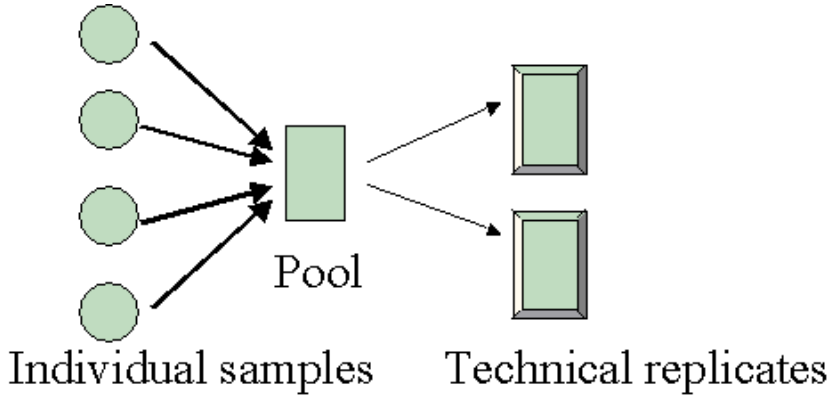
Figure 3. A 'round-robin' style of design: this is easiest with an odd number of chips, but similar designs exist for even numbers also.

A more common situation is parallel contrasts. For example to investigate the role of a receptor, one prepares wild-type and knockout animals, and then administers a ligand to half of each group, while giving a non-effective vehicle to the other half. Then there are four groups, and the contrast of interest is the difference between the effect of the ligand on WT and KO. A good design for this is:

Pooling

There is considerable disagreement about whether to pool individual samples, among practitioners and also among statisticians. Sometimes the amount of sample from any one individual sample is insufficient for hybridization and in that case, pooling is a practical necessity. In theory, if the variation of all genes were independent and approximately normally distributed, then pooling n independent samples would result in reduction of variance given by the formula:

$$\sigma^2_{Pool} = \sigma^2 / \#\ in\ pool$$



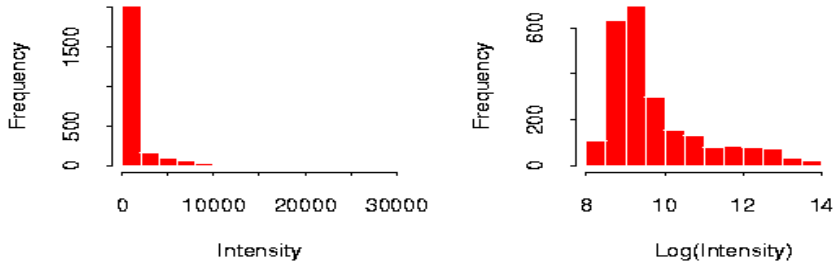
In practice the expression levels of many genes across samples have many more extreme values (outliers) than does the normal distribution. Some samples have levels of stress response proteins and immunoglobulins five to ten fold higher than typical. This can be due to many factors unrelated to the experimental treatment: for example, individual animals or subjects may be infected, or some tissue samples may be anoxic for longer periods than others. It is easier to detect this, and discard such a sample, if individual samples are hybridized. Finally if one pools samples, there is no way to estimate variation between individuals. For further information on excess variability between individuals, see Prichard et al "Project Normal", PNAS (2002)

Replicates

The question of how many replicates to do depends on how small the differences are that you want to detect, and the noise level in your system. Different systems have different noise levels, and a simple way to estimate the noise is to do three or four replicate hybridizations. For a cDNA system we get useful information about the variability in gene measures from three pairs of replicate dye-swap hybridizations (6 chips) using the same two (different) RNA samples.

The Distribution of Microarray Intensities

Before we start, a word about the distributions and transforms we use here. The distributions of gene expression measures are extremely skewed by statistical standards. Even after the log transform, which is the strongest skew-adjusting transform in regular use, the distribution of gene abundances remains visibly skewed. Two cautions should be drawn from this. First, it is generally a good idea to take a transform that makes variances of different variables equal. The log transform gets part-way there, but at the cost of introducing a very large variance at the lower end. Secondly some standard statistical procedures, such as linear models, depend sensitively on the assumption of normality. There does not appear to be any transform where this assumption holds very well. Hence statisticians have a preference for robust or non-parametric procedures.



The first thing to notice is that most genes are expressed at very low levels; few genes are expressed at high copy number. In statistical jargon we say that the distribution is skewed to the right. Statisticians often deal with highly skewed data on a logarithmic scale; this transform often corrects the skew for microarray data. Normally the distribution of intensities appears roughly bell-shaped; however depending on the choice of genes, and the estimation algorithm used (eg. how background is handled, how the low abundance genes are estimated) the distribution of intensities from the microarray may appear double-peaked or skewed on a log scale.

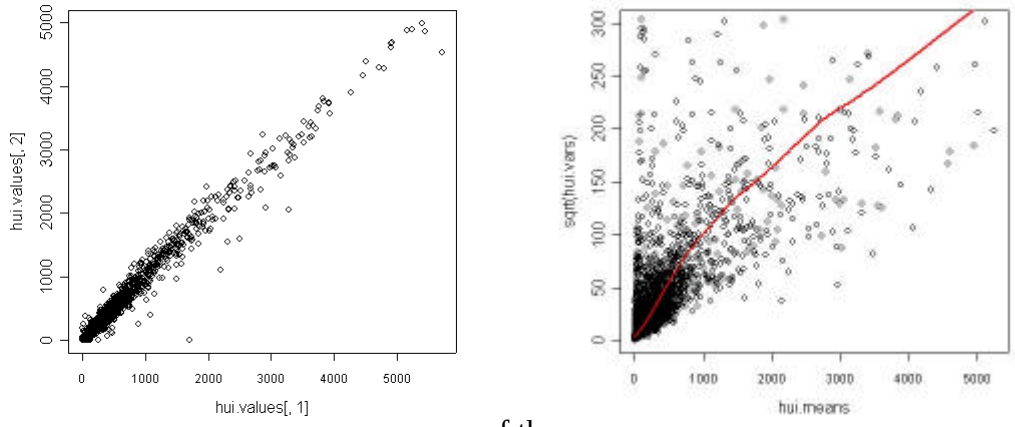
Note that the signals from the microarray are not direct measures of copy number, rather the signal from each gene probe is proportional to the copy number, but with a different proportion for each gene. Therefore we can't conclude from the graph that there are a small number of genes with a very low abundance, and quite a lot that are slightly more abundant.

This shape of distribution is pretty common in microarray data. However the rise on the left doesn't immediately make sense, because we expect a most genes to be expressed at very low copy number, and fewer genes expressed at higher levels. The current best explanation for the distribution shape is that the signal for each gene is due to a combination of the hybridization of that gene, plus some non-specific hybridization, from all the other similar sequences, or partial transcripts in the sample, plus noise: eg. dust particles, other labelled transcripts binding to streaks of other probe, etc. The amount of non-specific hybridization depends on the gene, but we think for most genes the amount of non-specific hybridization has some bell-shaped distribution (probably not normal). From the number of genes in the left hand side of the peak, we get some estimate of how reliable our estimates are.

Variation

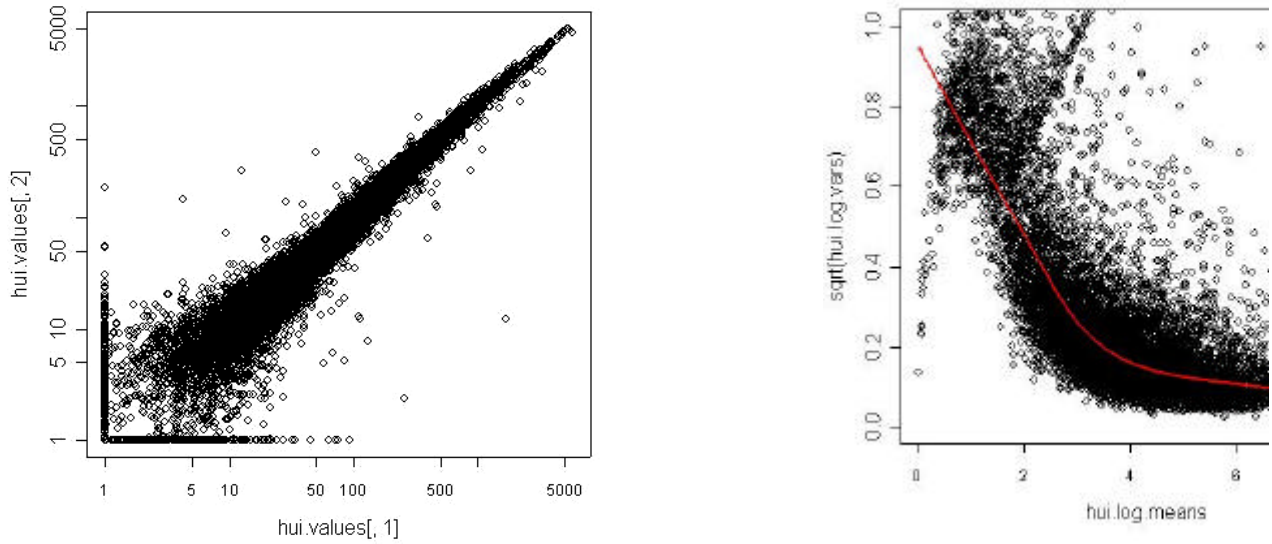
Let's distinguish technical variability, the typical differences between repeated measures on the same sample, from individual variation. Technical variation is due to differences in sample preparation, the course of hybridization, and other factors. This is usually what is called 'noise'. On top of that, different (healthy) individuals have consistently different patterns of gene expresssion. In experiments where several individuals, this may also be considered 'noise'.

A common observation in biology is that noise increases with level. So the technical variation in a measure of a housekeeping gene is higher than that of a transcription factor.



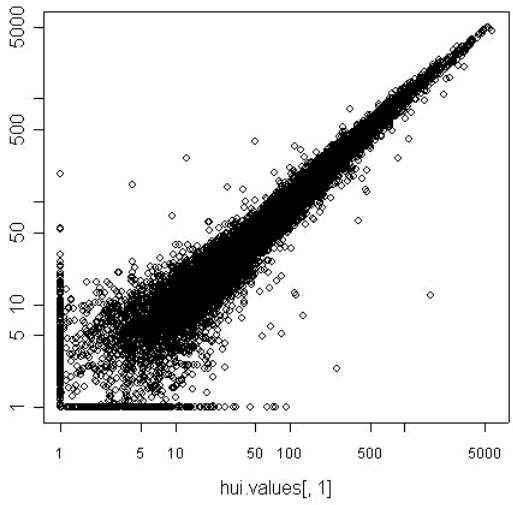
The plot on the left shows a scatter plot of the same genes on two chips. The plot on the right shows standard deviations across chips as a function of the mean over all chips.

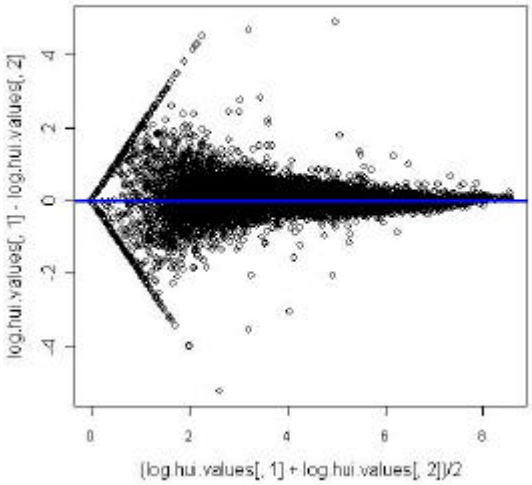
Most statistical procedures depend on all genes being compared have comparable noise levels, and will give erroneous p-values if there is great discrepancy. Statisticians treat this problem with transforms, and a common choice is fractional power (eg. cube root) or logarithm transforms. The most common transform in microarray is the logarithm transform. This appeals to some because the fold change appears the same for all genes. It also compensates for the intensity dependent noise, but actually over-compensates. Noise at the lower end is now higher than noise at the upper end.



The plot on the left shows a scatter plot of the log intensities of genes on two chips. The plot on the right shows standard deviations of the log intensities across chips as a function of the log mean over all chips.

At this point it is worth introducing a common device for displaying the comparison between two samples. The ratio-intensity plot (R-I plot). This is most convenient on a log scale because down-regulations (ratios lower than one are represented symmetrically with ratios higher.





We might ask if there is a simple transform that makes noise comparable at all levels. There are such transforms, but they are not simple. Several research groups presented variance-stabilizing transforms in 2002. Their proposals are based on a model where the noise for each gene has an additive component (perhaps reflecting background), and a multiplicative component (reflecting hybridization fluctuations). The simplest model is:

and this gives rise to a simple transform of this form:

$$g(y) = \log\left(y - \alpha + \sqrt{(y-\alpha)^2 + \sigma_\varepsilon^2/\sigma_\eta^2}\right)$$

Although in principle one should be able to estimate the parameters empirically, in practice you often get better results from other choices, and the groups have published calibration algorithms. One practical advantage of displaying data on scale is that straight lines on a scatterplot statistically significant differences.
A simpler approach is to try both a logarithm transform, and a cube-root transform; often one or the other will be almost as good as the variance-stabilizing transform.

Normalization Approaches

Why Normalize?

Biologists have long experience coping with systematic variation between experimental conditions that is unrelated to the biological differences they seek. However expression arrays have even more ways to vary systematically than measures such as rt-PCR. In practice methods that have worked well for these types of measures do not perform as well for microarray data, where there are many more dimensions of systematic differences observed.

Normalization is the attempt to compensate for systematic technical differences between chips, to see more clearly the systematic biological differences between samples. Differences in treatment of two samples, especially in labelling and in hybridization, bias the relative measures on any two chips.

To an astute observer of many microarray results, systematic non-biological differences between chips become apparent in several obvious ways

- Total brightness differs
- One dye seems systematically stronger than the other (in 2 –color systems)
- Background is different

Some causes of systematic measurement variation include:

- Different amounts of RNA
- One dye is more readily incorporated than the other (in 2 –color systems)
- The hybridisation reaction may proceed more fully to equilibrium in one array than the other
- Hybridisation conditions may vary across an array
- Scanner settings are often different, and of course
- Murphy's Law

In order to see real biological differences we attempt to compensate for these systematic differences in measurement. For convenience we will divide approaches into two types. Parametric approaches fit one (or sometimes two) parameter(s). Non-parametric approaches fit a curve (or a surface) usually the equivalent of 3 – 10 parameters.
Although the principles are similar, the details of normalization for cDNA arrays differ from normalization for Affymetrix arrays.

One early approach was to find a standard gene – preferably several genes – that are invariant across all chips or samples. This the commonsense approach, used routinely in rt-PCR. The standards people tried were 'housekeeping' genes – genes, required in all cell types – on the theory that they occur at nearly equal levels in all cells. This last assumption appears to be false:: housekeeping genes apparently vary substantially between cell lines, and certainly between cell types. See Novak et al, Genome Biology 2002. Perhaps several genes will be better indicators than one. Gene Logic has identified 100 genes that are the most constant in many cell types. They have not released the detailed results, but it appears that normalization by fitting these is still not as good as the best statistical techniques. The rest of this tutorial will describe these.

Most approaches to normalizing expression levels assume that the overall distribution of RNA numbers doesn't change much between samples, and that most genes change very little. The simplest approach posits that measures of most genes are proportional across any two different samples. This makes sense, since we are starting with equal quantities of RNA for the two samples we are going to compare, and, if the sizes of the RNA molecules are comparable, the number of RNA molecules should also be the roughly the same in each sample. Consequently, approximately the same number

of labeled molecules from each sample should hybridise to the arrays and, therefore, the total hybridisation intensities summed over all elements in the arrays should be the same for each sample. For a series of chips, define normalization constants C_1 , C_2 , …, by:

$$C_1 = \sum_{genes} f_1^{gene}; \quad C_2 = \sum_{genes} f_2^{gene}$$, and so on,

where the numbers f_i^{gene} are the fluorescent intensities measured for each gene on chip *i*. Then to normalize all the chips to a common total intensity *K* (e.g. the average or median total intensity among all the chips), for each chip *i*, divide all fluorescent intensity readings from chip *i* by C_i., and multiply by *K*.

There are many variations on this type of normalization, including scaling the individual intensities so that the mean or median intensities are the same within a single array or across all arrays, or using a selected subset of the arrayed genes rather than the entire collection.

Statistical approaches assume that most genes aren't really changed across all the conditions. For most laboratory treatments this seems reasonable, although treatments affecting transcription or translation apparatus have systemic effects; also malignant tumours often have dramatically different expression profiles. Following the assumption, then some overall characteristic of the expression distribution, such as the mean or median, should really be the same for all chips; the goal of normalisation is to make them equal. An extension of this idea is that all quantiles of the distributions must be equal.

We should keep in mind that normalization, like any form of data 'fiddling' adds noise (random error) to the expression measures. Statisticians try to balance bias and noise, and their rule of thumb is that it's better to under-correct for systemic biases than to exactly match. How do you tell how much correction is enough? Generally one stops correcting when the estimated remaining bias reaches the noise level.

Quality Control

A lot of the messy business of statistics is cleaning up data. Although this is less exciting, it is no less important, than normalization and other processes.

Wet Lab Quality Checks

The best place to check quality is in the wet lab, before the measures are taken. Two standard checks are RNA quality and dye incorporation.
Between the time that a sample is taken, and the time the RNA is extracted and purified, enzymes in the cell rapidly degrade mRNA by cutting it into shorter pieces. Most of these short pieces will hybridize more easily to several different probes, which distorts expression measures. One way to detect degraded RNA is to examine two abundant types of RNA – the 18S and 28S ribosomal RNAs. If the ribosomal RNAs are mostly intact they form two sharp peaks as the total RNA is washed through a gel. This may be done also with a commercial tool such as the Agilent BioAnalyzer.
Since the measures depend directly on the how much of the labelling dye is present on a probe, it makes sense to check how well the label is incorporated in the sample. In practice the amount of label in different samples varies, especially for the red Cy-5 dye. Microarray technicians have often observed that the Cy5 label is taken up poorly in hot humid summers. A commercial product to measure how much label is incorporated in the sample is the NanoDrop Probe.
Since it is much more trouble to detect and correct problems after the hybridization, it is worth the effort to check for hybridization problems in the lab. You may discard chips with very problematic hybridizations. You *should* do this before testing your favorite hypothesis; in the real world, you often do it interactively as you find faults with chips that don't fit your ideas so well.

Controls

If the wetlab measures are good, then the best further information about how even and specific the hybridization process was comes from the controls. There's no excuse any more for chips without a well-designed set of control probes. Negative controls are probes designed for DNA sequences that should never occur in your sample. Positive controls are multiple probes for sequences that should be abundant. Both positive and negative controls should be distributed over the chip. Spike-in controls are probes for transcripts not expected in your sample but added (in known amounts) in the hybridization mix.
After the wet-lab the next line of defence against corrupt data is the negative controls. They should all report low signal, and this low value should be fairly uniform (i.e. it should not show any pronounced spatial pattern, although control probes from different genes may pick up different amounts of non-specific signal). The signal from negative controls gives an estimate of the background in all signals due to non-specific hybridization and from the substrate. You won't be finding any gene signals reliably at the same levels as the signals from negative controls. You can also detect background anomalies using negative controls.

The second line of defence against corrupt data is the set of positive controls. Positive controls give some idea of the spatial variation in hybridization. Probes for the same gene should be uniform across the chip. The most common spatial patterns are gradients. Often during hybridization a two-color chip is placed on a surface, which isn't precisely level. More of the sample is present at one end of the chip, or along one side. In hybridization stations one often observes uneven hybridization, and high background, around the inlet ports – it seems the turbulent fluid affects the hybridization reaction. One should discard signals from the affected regions, and if this uneven pattern extends for a long way it's better to discard the chip.

The final line of defence is the spike-in controls. These give some idea of the accuracy and linearity

of the measures, in a well-done experiment that shows minimal background or spatial variation. Some very careful experimenters add some spike-ins to the sample before labeling, and some (previously labeled) after labeling.

Quality Control of Individual Probes

If the chip passes all the previous tests, the next step is QC for individual spots or probes. Most image quantification programs flag spots that fail their internal QC measures; it's rarely a good idea to keep spots that have been flagged. You may want to do further QC of individual spots based on several other measures reported by the image processing program (GenePix is especially good in this regard). It's not practical to examine thousands of spots individually. Some simple criteria that may be applied to all spots in batch mode use reported measures, for example the area of the spot, geometric measures of circularity, and uniformity.

The theory behind spot QC is to detect printing anomalies, rather than hybridization problems. The printer often drops small amounts of probe, elsewhere than intended. This becomes a problem if a spatter of probe for a highly expressed gene lands on a probe for a faint gene; then the signal from both channels looks more like the bright gene, rather than the gene which is annotated at that position. Another type of problem is spot formation – printers aim to deliver fairly round, even sized spots. When they fail, printed clones may flow into each other. So in practice it makes trouble to use data from extremely small, or extremely large spots, or those that are very irregular. Further measures you might use in batch filtering depend on the level of noise in the image, and the uniformity of the color ratios.

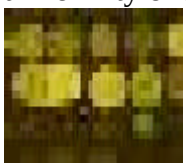


Figure. Section of cDNA image: some spots run into each other; these spots have excessively large areas.

The area criterion is the easiest to apply and understand. Spots whose size is only a few tens of pixels are much more likely to be scatterings of bright probe. Spots, which are much bigger than intended, are likely to be mingled with their neighbors.

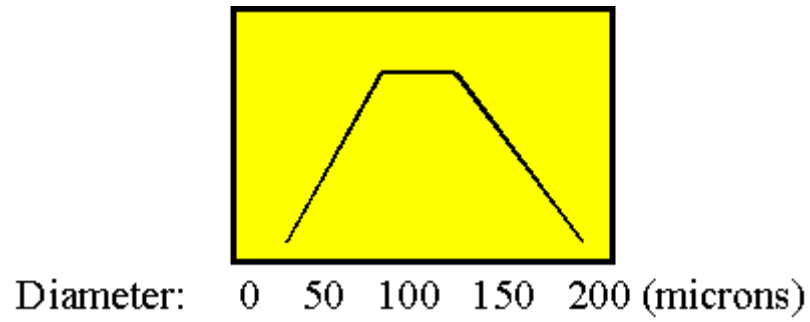


Figure. A plot of one quality score as a function of diameter, for a grid where the intended diameter is 100 microns, and the inter-spot distance is 200 microns.

There are further criteria. Most programs give both a mean and a median for a spot. If the spot has a reasonable distribution of pixels, the mean and median should be similar. We accept spots if the mean and median differ by at most 15%: $|m - m^*| < 0.15\ (m + m^*)/2$. If they are quite different, something strange is happening, such as a bright droplet. Finally many new chips feature duplicate (or more) probes for each gene. The signals from these duplicates should be similar. We use a 15% criterion there also: Accept if $|m_1 - m_2| < 0.15(m_1 + m_2)/2$.

It is simplest to set up criteria as filters, and to exclude spots that fail any quality criterion at a certain

threshold. However in practice few spots may pass all criteria, even with reasonable thresholds for each. Some groups use a composite score. (Wang et al 2002) construct quality measures q_1, q_2, q_3, and q_4, based on area, signal-to-noise, background level, and variability; they define a composite score $q^* = (q_1 q_2 q_3 q_4)^{1/4}$, and reject a spot if the composite $q^* < 0.8$. The threshold of 0.8 is somewhat arbitrary, although spots in their arrays with $q^* \sim 0.5$ have twice the random variation of those with $q^* > 0.8$.

In principle, most quality measures are continuous, and while there are obvious outliers, there is no clear-cut threshold. A better procedure than filtering would be to down weight probe signals, in further analysis, based on quality score. This poses a practical problem for most people, since it is difficult to use weight information in packaged software, although it is easy to adapt hand-coded R routines to weighted signals

Normalization of Competitively Hybridized (Two-Color) Microarrays

Normalization by Scaling

Scaling a chip means multiplying the signals (intensity measures) for all genes by a common scale factor. The reason to do this is that the total brightness is significantly different between the from the two channels. If the same total weight of RNA is hybridized in both channels, the differences between channels must be due to different uptake of label (dye bias) of RNA hybridized. In fact microarray technology can only measure relative levels of expression: per mg RNA. For a two-color chip, we have two measures for each gene, one from each channel. For each chip we compute scale factors C_{red} and C_{green} , by:

$$C_{\text{red}} = \sum_{i=1}^{N} f_i^{\text{red}} ; \quad C_{\text{green}} = \sum_{i=1}^{N} f_i^{\text{green}}$$

where G_i and R_i are the measured intensities for the i-th array element (for example, the green and red intensities in a two-color microarray assay) and N is the total number of elements represented in the microarray. To compare ratios both intensities are appropriately scaled, for example:

$$f_i^* = f_i^{\text{red}} / C_{\text{red}} ; f_i^* = f_i^{\text{green}} / C_{\text{green}}$$

This is equivalent to subtracting their average from the logarithms of all the expression ratios, which results in a mean $\log_2$(ratio) equal to zero, or the (geometric) mean ratio is equal to 1.

In order to make individual channels more comparable across chips, the same constant is used for all chips. In practice there are often anomalies at the top end, for examples a number of probes are saturated. One gets more consistent results by using a robust estimator, such as median or 1/3 – trimmed mean: take mean of middle 2/3 of probes, and scale all probes to make those equal. (John Quackenbush suggested this originally, but TIGR now uses lowess – see below.)

Two Parameter Normalization Methods

Whereas normalization adjusts the mean of the $\log_2$(ratio) measurements, it is common to find also that the variance of the measured log2(ratio) values to differ between arrays. One approach to dealing with this problem is to adjust the $\log_2$(ratio) measures so that the variance is the same. This often works, in reducing variance, but sometimes works too well, in that variance of individual measures is actually increased. Probably a partial adjustment is optimal, but it seems unprincipled. Another two-parameter approach is a linear regression of one channel on the other. This doesn't seem to do as well.

Intensity Dependent Normalization with Lowess

With a little experience it becomes clear to a researcher that these approaches do not compensate for all the systematic differences between chips that obscure and bias analysis of real biological differences. Several statisticians have tried to identify variables, which systematically bias expression ratios. For example one commonly observes that the $\log_2$(ratio) values have a systematic dependence on intensity – most commonly a deviation from zero for low-intensity spots. Under-expressed genes appear up-regulated in the red channel. Moderately expressed genes appear up-regulated in the green channel. No known biological process would regulate genes that way – this must be an artefact. It appears that the explanation is chemical: dyes don't fluoresce equally at

different levels, because of different levels of ‘quenching’ – a phenomenon where dye molecules in close proximity, re-absorb light from each other, thus diminishing the signal. Quenching acts at different levels for each dye.

The easiest way to visualize intensity-dependent effects is to plot the measured $\log_2(R_i/G_i)$ for each element on the array as a function of the $\log_2(R_i*G_i)$ product intensities. This 'R-I' (for ratio-intensity) plot can reveal intensity-specific artifacts in the $\log_2$(ratio) measurements. Note that Terry Speed’s group calls these variables ‘M’ and ‘A’, and the plot is an ‘MA plot’.

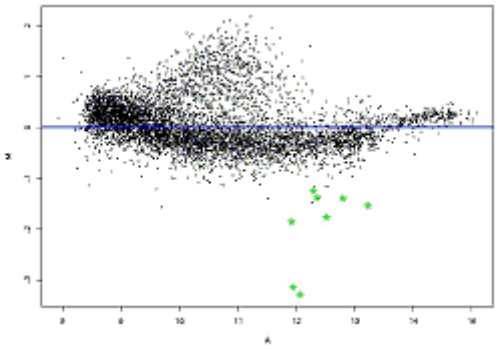


Figure 1. Ratio-Intensity plot showing characteristic ‘banana’ shape of cDNA ratios; log scale on both axes. (courtesy Terry Speed)

We would like a normalization method that can remove such intensity-dependent effects in the $\log_2$ (ratio) values. The functional form of this dependence is unknown, and must depend on many variables we don’t measure. An ad-hoc statistical approach widely used in such situations, is to fit some smooth curve through the points. One example of such a smooth curve is a locally weighted linear regression (lowess) curve. Terry Speed’s group at Berkeley used this approach.

To calculate a lowess curve fit to a group of points (x1,y1),…(xN,yN), we calculate at each point xi, the locally weighted regression of y on x, using a weight function that down-weights data points that are more than 30% of the range away from xi. We can think of the calculated value as a kind of local mean. For each observation i on a two-color chip, set xi = log2(Ri*Gi) and yi = log2(Ri/Gi). The lowess approach first estimates y(xk), the mean value of the log2(ratio) as a function of the log2 (intensity). Lowess normalization corrects systematic deviations in the R-I plot by carrying out a local weighted linear regression as a function of the log2(intensity) and subtracting the calculated best-fit average log2(ratio) from the experimentally observed ratio for each data point.

The normalized ratios r* are given by

$$\log(r_i^*) = \log(R_i / G_i) - \text{lowess}(R_i * G_i)$$

The result is that ratios at all intensities have a mean of 0, as seen below.

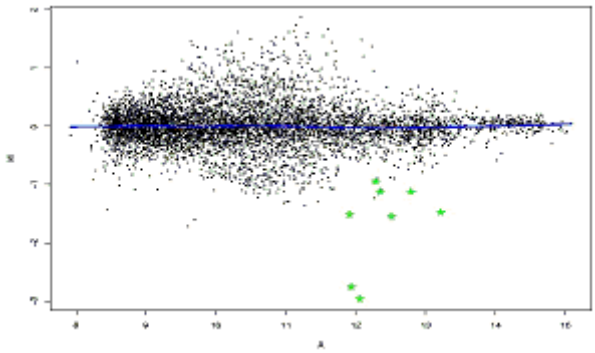


Figure 2. As in Figure 1, but corrected by lowess normalization.

Global versus local normalization.

Most normalization algorithms, including lowess, can be applied either globally (to the entire data set) or locally (to some physical subset of the data). For spotted arrays, local normalization is often applied to each group of array elements deposited by a single spotting pen (sometimes referred to as a 'pen group' or 'subgrid'). Local normalization has the advantage that it can help correct for systematic spatial variation in the array, including inconsistencies among the spotting pens used to make the array, variability in the slide surface, and slight local differences in hybridisation conditions across the array. There is some controversy among biotechnologists about how likely it is that a single print tip will cause a systematic variation.

Another approach is to look for a smooth correction to uneven hybridisation. The thinking behind this approach is that most spatial variation is caused by uneven fluid flow. Flow is continuous, and hence the correction should be continuous as well.

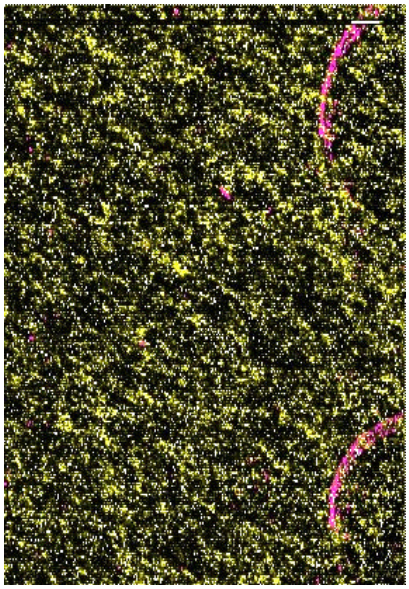
When a particular normalization algorithm is applied locally, all the conditions and assumptions that underlie the validity of the approach must be satisfied. For example, the elements in any pen group should not be preferentially selected to represent differentially expressed genes, and a sufficiently large number of elements should be included in each pen group or spatial area for the approach to be valid.

Quantile Normalization

A good design will place all contrasts of interest directly on chips, but sometimes that is impossible, or just not done. In that case we may want to compare parallel measures: , ie. measures that are not directly contrasted on an array. We observe that variance is very high between parallel measures. We need a kind of normalisation that works across arrays as well as within arrays. It turns out that quantile normalization works quite well at reducing variance between arrays, while not losing any of the properties of lowess normalization.

Quality Control of Affymetrix Chips

One of the particular values of a multi-probe system is that all probes effectively act like positive controls. Since the Affymetrix probes have such different response characteristics, you don’t want to reject large or small probes, but with a good multi-chip model, the hybridization problems show up as outliers from fitted multi-probe model.



Normalization of Affymetrix Chips

Normalization by Scaling and its Limitations

The simplest approach to normalizing Affymetrix data is to re-scale each chip in an experiment by its total intensity, as described in the Normalization Introduction. Variants of this approach, scaling by trimmed mean intensity, or by median intensity, are widely available in commercial software. Affymetrix introduced a new approach for their 133 series chips, using a set of 100 'housekeeping genes': the chips are re-scaled so the average values of these housekeeping genes are equal across all chips. The author believes these approaches are adequate for about 80% of chips in practice.
To do better, we examine in detail the relationships among replicate chips (chips hybridized to the same sample). Figure 1 shows a scatter plots of probes from one pair of chips; there is clearly a non-linear relation among probes. Figure 2 shows plots of probe distributions from a number of replicate chips; these distributions have very different shapes; any scaling transform applied on a log scale, will shift the distribution curve to the right or left, but not change its shape. Finally figure 5 shows R-I plots of pairs of Affymetrix replicate chips; a scaling transform will shift the R-I plots up or down, without changing their configuration. For perhaps 80% of chips, (perhaps 65% of pairs), the relationship is close enough to linear that a scaling transform will get results to within 20% of the best possible. The relationships among different chips are quite non-linear in perhaps 20% of cases. We want to correct that to get the best possible accuracy.

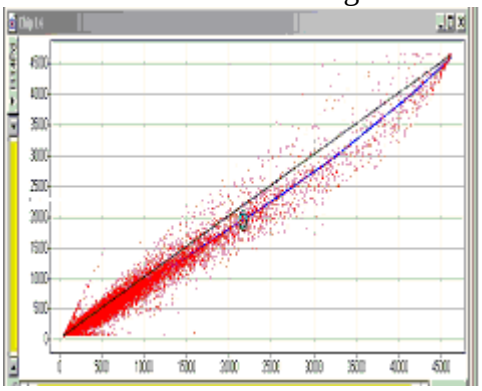


Figure 1. Plot of probe signals from two Affymetrix chips hybridized with identical mRNA samples. The black straight line represents equality, while the blue curve is a spline fit through the scatter plot.

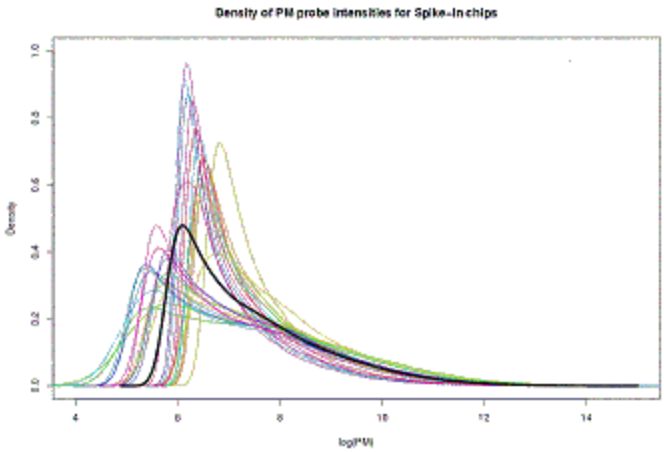


Figure 2. Density of PM probe signals on 23 different chips from GeneLogic spike-in experiment (Courtesy of Terry Speed)

Two-Parameter Methods

Two-parameter methods can do better, at the expense of greater complexity. MAS5 introduced a reference (baseline) chip method using linear regression. The procedure is to construct a plot of each

chip's probes against the corresponding probes on the baseline chip; eliminate the highest 1% of probes (and for symmetry the lowest 1%). Fit a regression line to the middle 98% of probes. Another two-parameter approach is to both re-scale and shift the origin, in order fit both the mean and the standard deviation of the probe distribution to the common mean and standard deviation of all data. This seems to do somewhat better than regression, in reducing noise (variation among replicate measures on the same sample), at the cost of (sometimes) introducing a few negative values.

Invariant Set Normalization

Li and Wong introduced a method, where a large number of genes are selected ad-hoc as references, rather than using a standard set of 'housekeeping genes'. Their method assumes that there is a subset of unchanged genes, between any two samples. Their method selects a subset of genes $g_1, \ldots, g_M$, whose probes: $p_1, \ldots, p_K$, (K ~ 10000), occur in the same rank order on each chip such that $p_1 < p_2 < \ldots < p_K$ in both chips (an invariant set); then fits a non-parametric curve (running median) through the points $\{ (p_1^{(1)}, p_1^{(2)}), \ldots, (p_K^{(1)}, p_K^{(2)}) \}$. Ideally one would like a common invariant set of reference genes across all chips, but in practice, only a very few probes are in common rank order, or even close to that, across all chips.

Quantile Normalization

Terry Speed's group introduced a non-parametric procedure normalizing to a synthetic chip. Their method assumes that the distribution of gene abundances is nearly the same in all samples. For convenience they take the pooled distribution of probes on all chips. Then to normalize each chip they compute for each value, the quantile of that value in the distribution of probe intensities; they then transform the original value to that quantile's value on the reference chip. In a formula, the transform is

$$x_{norm} = F_2^{-1}(F_1(x)),$$

where F_1 is the distribution function of the actual chip, and F_2 is the distribution function of the reference chip.

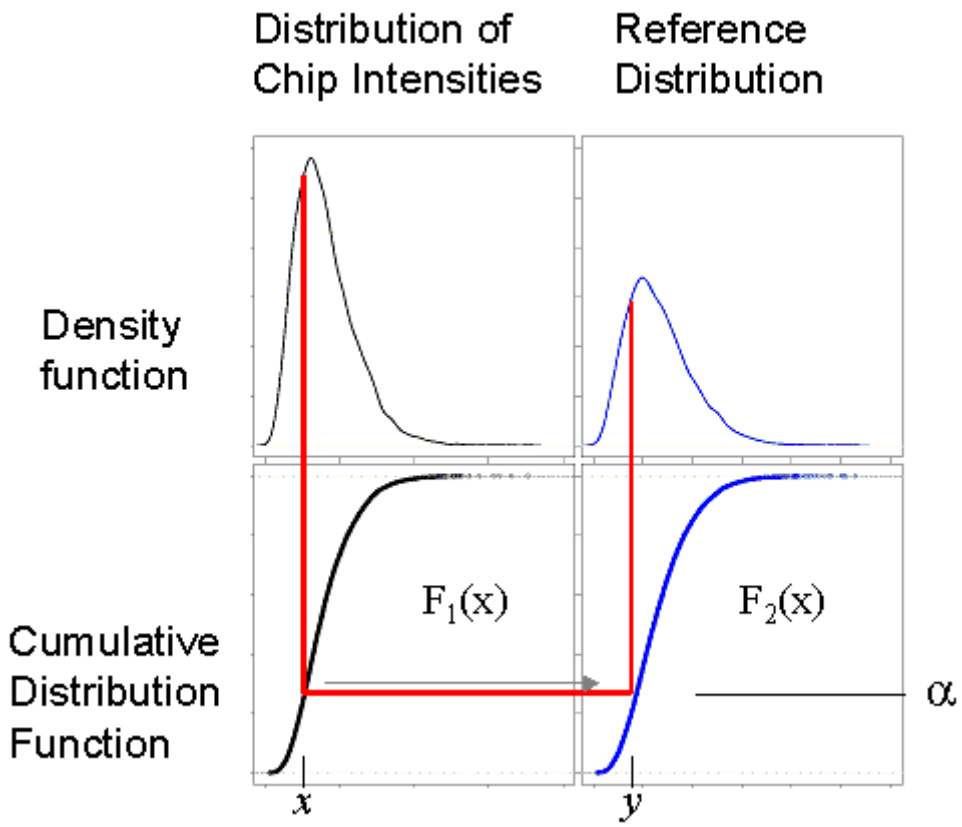


Figure 3. Schematic representation of quantile normalization: the value *x*, which is the α-th quantile on the chip, is mapped to the value y, which is the α quantile of the reference distribution.
In practice this transform is non-linear, but not usually too different from straight. See Figure 4. In practice this removes most of the apparent bias from the R-I plot. See figure 5. It also reduces variance among replicates, much more than normalization by scaling.

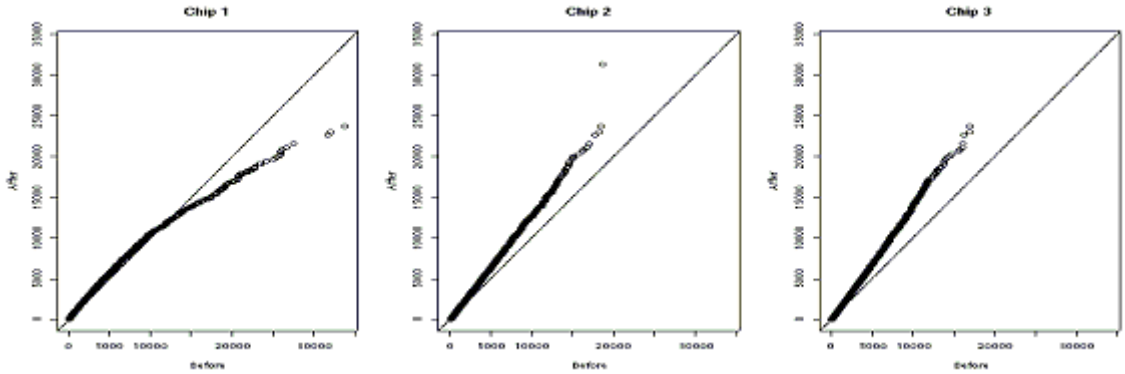


Figure 4. Some typical transforms by quantile normalization. Many are nearly linear, but some are quite non-linear.

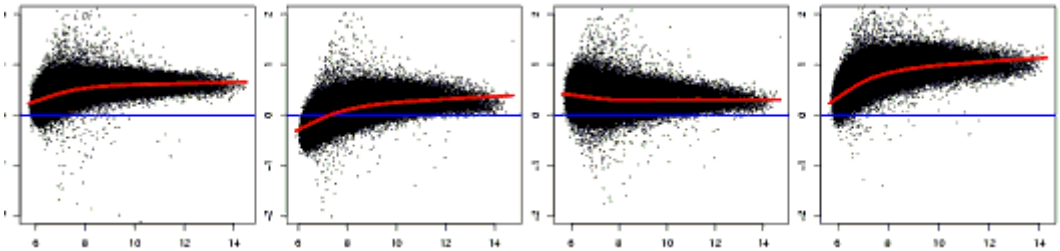


Figure A. Ratio Intensity Plot of all probes for four pairs of chips from GeneLogic spike-in experiment

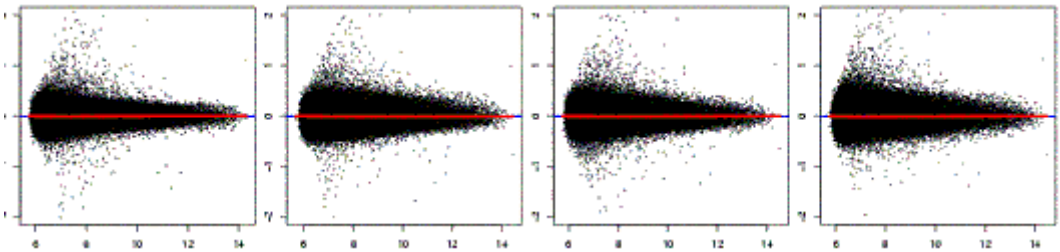


Figure B. As in A, after normalization by matching quantiles. Both figures courtesy of Terry Speed
Local Regression
We construct a synthetic reference chip by averaging the values of each probe across all chips.

Critical Assessment

Ideally we would like a method that is based on some understanding of the hybridization process, and uses simple statistical procedures, to bring all chips to a common reference. Scaling is simple, but seems to be inaccurate. Methods based on multiple house-keeping genes, such as the MAS method for the 133 chip, and the Li and Wong method, appear promising, however they would work better if the reference set of genes were similar across all chips. These methods use a single chip reference, so peculiarities in that chip are forced onto all the others. Quantile normalization uses a single standard for all chips, however it assumes that no serious change in distribution occurs. This appears to be a rather strong assumption about gene distributions; however, in practice genes move up and down roughly equally; it would need several hundred genes to be changed greatly and in one direction, to drive quantile normalization in error by more than 20%. This may well be true in studies of senescence, or interference with basal transcriptional apparatus, or selective comparisons of RNA's attached to ribosomes, and perhaps in extremely malignant tumors.

Low-Level Analysis of Affymetrix Chips

Description of Affymetrix Probes

The strength of the Affymetrix system is that multiple distinct oligonucleotide probes on each chip represent every gene. However the signals from the different probes for the same gene aren't the same; signals from individual probes for the same gene may differ, on the same chip, by as much as two orders of magnitude (a factor of 100). See Figures 1 and 2. The sequences are different, and the probes have different hybridization constants for their target: the most important factor in signal intensity is C:G content. How do we combine signals from the many probes for a gene, into a single estimate of the abundance of that gene?

Figure 1. Images of probes from human GAPDH probe set extracted from an Affymetrix U95A chip image. PM probes in top row; corresponding MM probes on bottom. Two probes are bright, three others are moderately bright, the rest are dim.

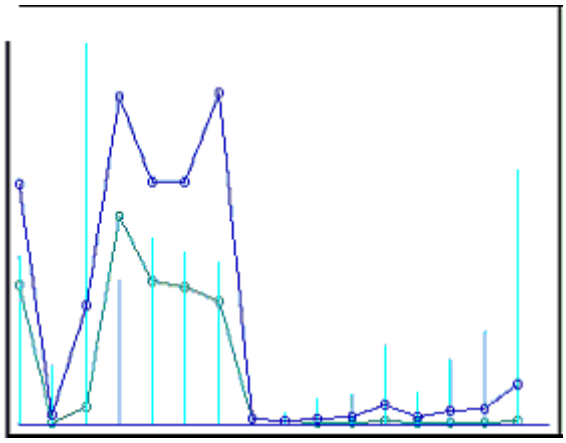


Figure 2. Line representation of intensities from a typical probe set in the mouse chip. PM values appear in blue; MM in green. Vertical axis height represents 30,000. Pale blue lines represent standard deviations of probes across chips. Image from dChip.

Estimates from Probe-Level Signals

There has been considerable discussion over the appropriate algorithm for constructing single expression estimates based on multiple-probe hybridization data. To date, over a dozen different methods have been published, which aim to synthesize the different readings from the various probes for a gene, into a single estimate of transcript abundance. Affymetrix recently sponsored a conference on the topic.

Affymetrix MicroArray Suite

Affymetrix has upgraded their MicroArray Suite (MAS) software several times over the short history of their product. MAS 4 was the standard until January 2002 and is still the most commonly cited measure in published papers. The simplest way to get one number from several numbers is to take an average. MAS 4 calculates a robust average of the probe-pair differences (PM – MM) for each probe pair representing a gene. The more recent MAS 5 improves in three ways: first the difference is

taken between PM and an estimate of background based on MM (rather than MM itself); secondly the intensities are transformed to a logarithmic scale before the average is taken; third the average is a more sophisticated robust mean (Tukey biweight).

Principles of MAS5

MAS 5.0 computes local background in each of 16 squares, and then subtracts a weighted combination of these background estimates at each probe. For each probe set, compute a robust average of log probe pair differences: $\log(PM_j/MM_j)$. Call this SB. Then adjust each PM probe as follows: if $MM_j < PM_j$, then $\log_2(PM_j/MM_j)$ is used; if $MM_j > PM_j$, then $\log(PM_j) - SB$ is used, unless SB is too small. See the "Statistical Algorithm Description Document" from Affymetrix, for more details.

Critique

The idea of averaging different probe intensities for the same gene is seems quite wrong. It is like averaging the angular height of a building seen from different vantage points; or measuring a person's height in inches, feet, cm, ells, furlongs, and meters, and taking the average; or averaging the readings from scans taken at very different settings. A second failing is that there is no 'learning' about probe characteristics, based on the performance of each probe across chips.

Multi-Chip (Linear) Models

A chemical motivation for multi-chip models comes from reasoning that the amount of signal from one probe in a gene's probe set, should depend both on the amount of that gene in the sample, and on the specific affinity of the probe for that gene's mRNA. The statistical motivation for multi-chip models is observing that the signals from individual probes move in parallel across a set of chips (this is clearer with the better normalizations). See Figure 3. Another way to see this is to watch the animations of probe sets in dChip.

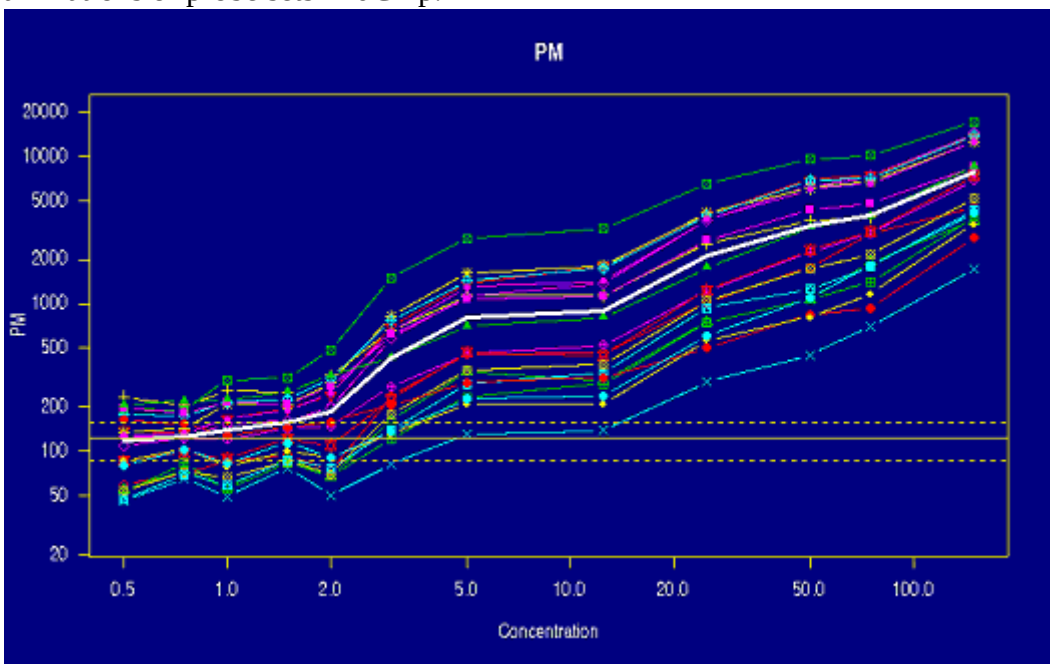


Figure 3. Probe signals from a spike-in experiment. The concentrations are plotted along the horizontal axis (log scale), and the probe signals are plotted on the vertical axis (log scale). Each probe is represented by one color. The different probe signals change in parallel. Image courtesy of Terry Speed

We want a statistical model that estimates both the factors probe affinity, and gene abundance. Statisticians like linear two-factor models: that means, the errors in each data point have similar variances, and the two factors combine in a simple way. If the signal from each probe is proportional

to both probe affinity, and gene abundance, then it must depend on the multiplicative product. Suppose for one target gene, the chip has a set of probes $p_1,...,p_k$; each probe p_j binds to the target with affinity f_k. Suppose in each sample i an the gene occurs in amount a_g. Then the intensity of probe j on chip i should be proportional to $f_k\, a_i$. See figure 4.

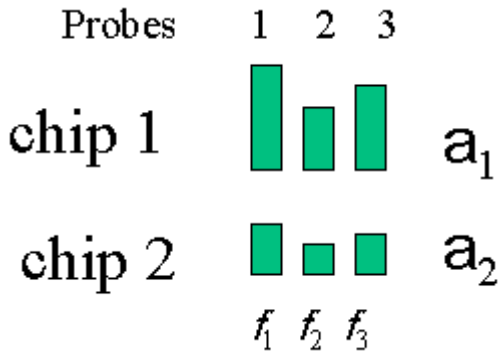


Figure 4. Ideal linear model relationship among intensity (height of green bars), abundance of transcript (a_i), and probe affinity (f_j).

In practice, the discrepancies between data and ideal model, include frequent outliers, besides the usual random fluctuations in signal intensities. Outliers are measures that lie far beyond the typical range of 'noise' (random variation). These may be due to scratches, or uneven heating, or other artefacts. See Figure 1 in Quality Control. Typically 10-15% of probes in an Affymetrix chip are outliers. Most methods to fit data flounder badly on data with this many outliers. One approach is to try to identify the outliers, and exclude them; this is the Li & Wong approach. Their method proceeds in this cycle: fit, identify outliers, throw out outliers, and fit again. Another approach is to use a robust fitting method. Robust methods try to fit the majority of data points quite well, but willing to fit a small fraction quite badly. Some such methods are median polish, or IRWLS (iteratively re-weighted least-squares), which are implemented in RMA. Another approach is least median squares, which is not implemented.

Constant Variance – the Li and Wong Model and Critique

Li and Wong originally suggested the model $PM_{ij} - MM_{ij} = f_k\, a_i + \varepsilon_{ij}$, following on from MAS4. Since then they have found better fits with the model $PM_{ij} = f_k\, a_i + \varepsilon_{ij}$, (PM-only). Li-Wong assumes that the noise in all the probe measures is roughly same size. In practice all biological measures exhibit intensity-dependent noise. (see Figure 4 in Distributions). The effect of their assumption is that probes with smaller variation are ignored, even though this variation may be measuring real differences. Fortunately the bright probes are often the most specific, and it does little harm to ignore the majority of probes, if the bright probes are good. They have tuned their fitting procedure to try to reduce the emphasis on the very bright probes, but this has resulted in often throwing out a good bright probe probe as an outlier.

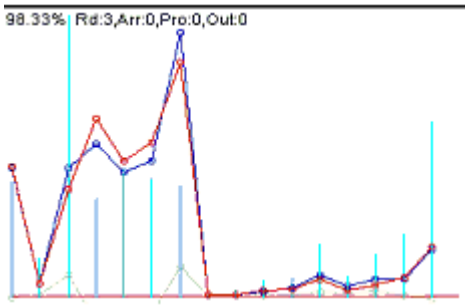


Figure 5. A probe set with values (represented by red lines) fitted to actual PM-MM values (in blue).

Proportional Variance – RMA

This is largely the work of Terry Speed's group at Berkeley, especially Ben Bolstad, and Rafael Irizarry. They work only with PM values, and ignore MM entirely. They take a log transform of equation () and find

$$\log(PM_{ij}) = \log(a_i) + \log(f_j)$$

With errors proportional to intensity in the original scale, the errors on the log scale have constant variance. After background subtraction and normalization they fit:

$$\text{nlog}(PM_{ij} - \text{bg}) = a_i + b_j + \varepsilon_{ij}$$

where nlog is their terminology for 'normalize and then take logarithm'. They fit this model by iteratively re-weighted least squares, or by median polish. Code is available in the affy package on BioConductor, together with quantile normalization.

Critical Assessment

This appears to be the best overall method available. See figure 6. Comparing the performance on replicate arrays – so criterion is noise should be small. four strata of genes – lowest to highest expression. Measures were computed for each and standard deviation. MAS5 apparently does a decent job on high abundance genes, but the multi-chip models do better on low-abundance genes, such as transcription factors, and signalling proteins. Affymetrix has seen the evidence, and they are planning their own multi-chip model. However details are not being revealed. Furthermore the marketing people at the company want to remove information from the public domain. This will hinder further improvements to the model, and prevent people from using the best analytic tools for their data.

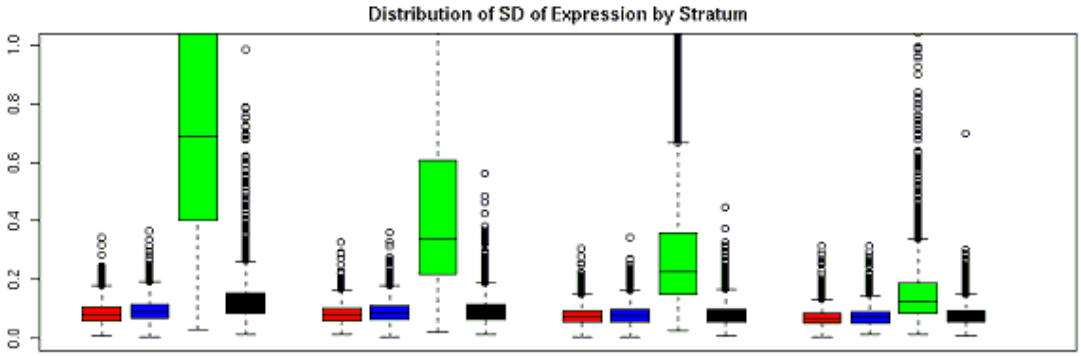


Figure 6. Comparison of MAS5 (green), dChip (black), RMA (blue), RMA (red): The genes have been divided into quarters based on average expression. Each boxplot represents the standard deviation of genes in one fraction. Note that the multi-chip models do almost ten times better than MAS5 on the low-abundance genes; this category includes most transcription factors and signalling proteins.

Software Available

Li and Wong's method is available through their program dChip, at www.dchip.org. Academic licenses are free.

The RMA method is available as part of the affy package in the Bioconductor tools suite: see www.bioconductor.org. There is also a windows standalone from Ben Bolstad's web site. A commercial software vendor, Iobion, has incorporated RMA into their GeneTrafic product.

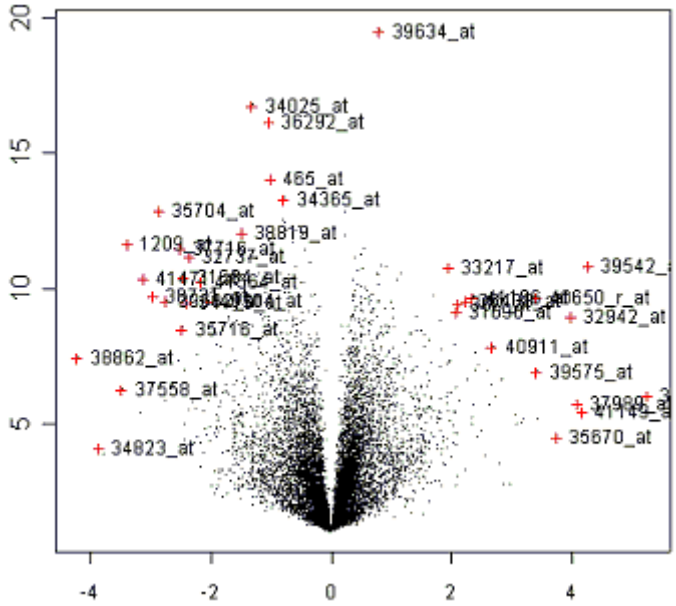
Exploratory Methods

Exploratory Graphics

The goal of exploratory graphics is to easy identify genes and groups. Several types of display make it easier to select differentially expressed genes (those whose expression levels change significantly with the experimental treatment or clinical condition). Some displays showing groups appear in Clustering.

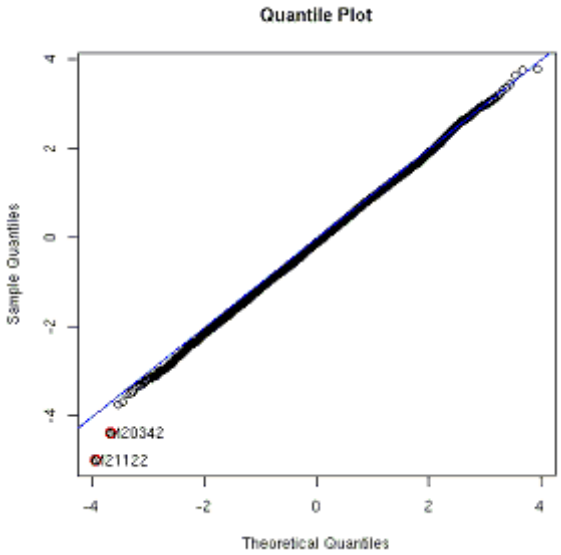
Volcano Plot

The 'volcano' plot is a heuristic device that arranges genes along dimensions of biological and statistical significance. The first (horizontal) dimension is the best p-value for a t-test of differences between samples (most conveniently on a negative log scale – so smaller p-values appear higher up); this indicates the statistical evidence for change. The researcher can then make judgements about the most promising candidates for follow-up studies, by trading off both these criteria by eye. With a good interactive program <link>, it is possible to attach names to genes that appear promising.



Quantile Plot

A major issue in microarray studies is separating out false positives from true positives. See Statistical Significance for a detailed treatment. A convenient way of assessing the most likely true positives is to plot the t-scores obtained by the test against the t-distribution. If the test scores follow the t-distribution this plot will be a straight line. The really significant genes stand out from the straight line.


Quantile Plot
Sample Quantiles
Theoretical Quantiles
20342
21122
4
2
0
-2
-4
-4
-2
0
2
4

Clustering

Pattern-Finding and Clustering

The goals are to identify groups of genes, and to find relations among samples, and to identify outliers among samples. The first widely publicized studies using microarrays aimed to find uncharacterized genes which act at specific patterns during the cell cycle; clustering is a natural way to select with similar expression profiles. Clustering is the natural first step in doing this. Unfortunately many people got the impression that clustering is the 'right' thing to do with microarray data, and many software packages have catered to this impression. The proper way to analyze data is the way that addresse the goal at which the study was aimed. If you have limited ideas about what groups you will find, clustering remains a valuable exploratory technique for suggesting resemblances among groups of genes. It's not a way of finding up- and down-regulated genes in an experimental study.

As with any exploratory technique, one should look to see what may underlie the groups, before going to the lab. In practice the author finds that clustering most often identifies systematic differences in collection procedures or experimental protocol. These are useful but not biologically significant. Most breast cancer profiles segregate into ER+ and ER-, which is re-assuring but hardly news

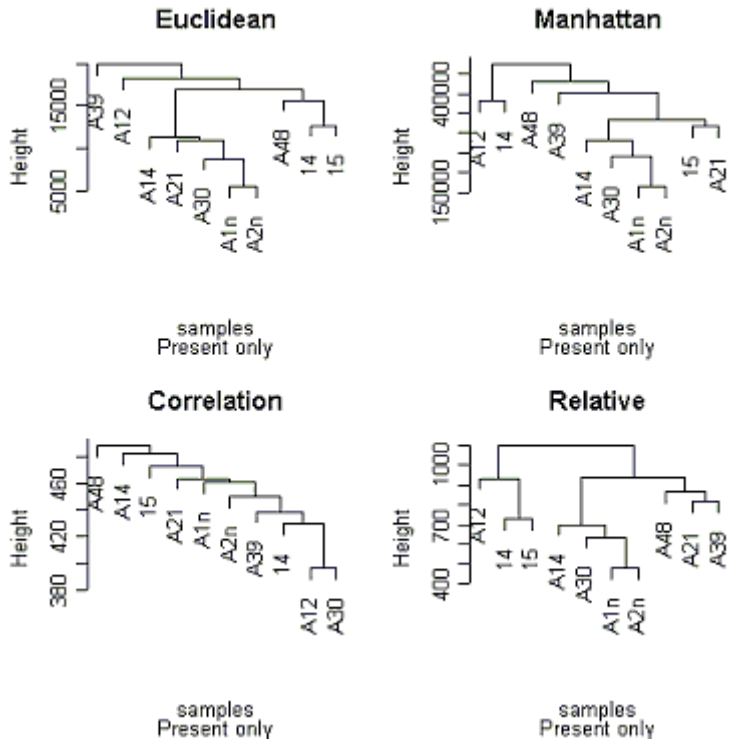
How to Do Clustering

After that disclaimer, suppose that we want to find groups of similar genes, how do we go about it? Almost all exploratory microarray methods use the idea that differences between gene expression profiles are like distances. How to get from many differences to a single measure of distance is somewhat arbitrary. Different methods give different results. Nevertheless it's useful to think about what similarities will be amplified by the data scale you use, and what assumptions are made by the assembly method you choose. Four choices you have are:
i) what scale,
ii) what selection of genes,
iii) what metric (distance measure), and
iv) what clustering assembly algorithm.

Scale and Selection

Differences measured on the linear scale will be strongly influenced by highly expressed genes. The log scale will amplify the noise among genes which have low expression levels. In the author's opinion, a variance stabilizing transformation (such as Durbin and Huber's) is a more appropriate scale for multivariate exploratory techniques, such as clustering and PCA (see below). Genes whose signal falls within the background noise range are probably contributing just noise to your clustering (and any other global procedure). It's probably wise to discard them.
Most cluster programs give you a menu of distance measures: Euclidean, Manhattan distances, and some relational measures: correlation, relative, and mutual information. The distance measures refer to how differences are combined: Euclidean is straight-line distance: (root of sum of squares), Manhattan is sum of linear distances. The correlation distance measure is actually 1-r, where r is the correlation coefficient. The mutual information (MI) is defined in terms of entropy: $H = \Sigma p(x) \log_2(p(x))$ for a discrete distribution {p}. Then $MI(g_1,g_2) = H(g_1) + H(g_2) – H(g_1,g_2)$ for genes g. This measure is robust – not affected by outliers. However it is tedious to program, because it requires adaptive binning to construct a meaningful discrete distribution.
By and large there are no theoretical reasons to pick one over the other, since we don't really know what we mean by 'distance' between expression profiles. Most of these measures are fairly sensitive to outliers, except mutual information. Robust versions of these measures can easily be constructed

by a competent statistician, but are not available in most software packages. However we do get different results depending on the algorithm we use, as shown below for a study with 10 samples: two normal samples and two groups of tumor samples.



Clustering of the same data set using four different distance measures. All genes were on a logarithmic scale, and only genes with an MAS 5 'Present' call in 8 out of 10 samples were used (Affymetrix data). The four measures are listed in the titles; 'relative' is |x-y|/|x+y|.

Algorithms for assembling clusters

Most biologists find hierarchical clustering more familiar, and other algorithms somewhat magical. Statisticians object to hierarchical clustering because it seems (falsely) to imply descent; however this is a quibble: all of the common clustering methods are based on models which don't really apply to microarray data. Broadly speaking, the differences between clustering methods show up in how ambiguous cases are assigned; if you have very many ambiguous cases you'll see great differences; however if so, then maybe clustering isn't appropriate anyway, because the data don't separate into groups naturally. The k-means and SOM methods require you to pick a number of clusters to target, but you don't know ahead of time, you'll be trying out lots of values. A criterion that some people use to assess how many clusters to use is to look at how much the intra-group variance drops at each stage.

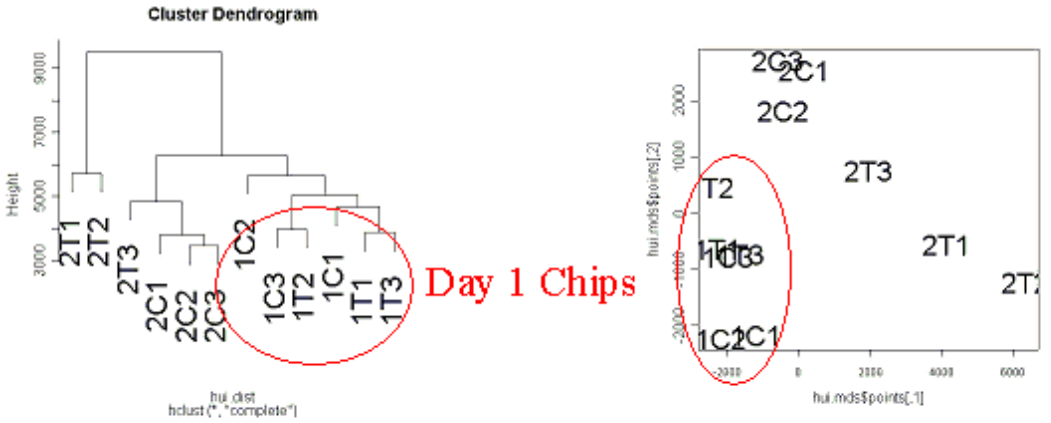
Statistical significance of clusters by bootstrapping

An important but rarely asked question, is how reliable are the clusters you obtain. Another approach is to use Bootstrap or Jack-knife analysis, where you re-cluster many times, each time re-sampling conditions or genes from your original data, and then derive new clusters of genes or conditions. Branches in a hierarchical cluster that are supported by a large fraction of the re-sampled data sets are considered moderately reliable. A common figure is 70%, but this is arbitrary, like the often-quoted 5% level of significance.

Principal Components and Multi-dimensional scaling

Several other good multivariate techniques can help with exploratory analysis. Many authors suggest principal components analysis (PCA). IMHO this is not very often useful, since the first PC will be

largely concerned with unresolved systematic error in your data. In fact some people have suggested seriously normalizing by removing the first PC. PCA is also not terribly robust to outliers, and is sensitive to the scale that is used.
A similar technique that I like for graphical display is multi-dimensional scaling. The classical form of MDS is identical to PCA for most data sets. However if you fix a dimension, a modern iterative algorithm can place the points in a fairly representative arrangement that is more representative of true distances, than the PCA for that number of dimensions.



Cluster diagram

Multi-dimensional scaling

A study comparing treated and control: two batches of 3 treatment (T) and 3 controls (C), done on two different dates: 1T3 represents 3rd treated sample on day one. Both represent genes on log scale. We can see that the day 1 chips cluster together, and are displayed together by MDS. However the day 2 genes seem to fall into two distinct clusters, which don't divide neatly along T and C. Should this experiment be abandoned? The MDS plot shows that the C samples on day 2 are in fact quite close, where as the 3 T's are more disparate but all quite different from the C's. We can work with the day 2 data quite confidently, and separately from day 1. Data courtesy of Hui Gao.

Statistical Tests

The Purpose of Formal Statistical Tests

Microarray data is used both as a guide to further, more precise studies of gene expression, and sometimes the microarray data itself is presented as evidence for changes in gene abundance. If the expression data is used as evidence, then the experimenter must present the degree of evidence – ie. the correct 'p-value' – for each gene selected out of the thousands in the array. Most microarray papers present p-values for individual genes more significant than they should be, for the reason that many genes are tested in parallel; the p-value means something different in the multiple-testing context.

The normal follow-on to a microarray study is to estimate abundance of single genes by qt-PCR or other methods. We may want simply to rank the genes in order of likely difference, for future qt-PCR studies. Then the goal of the analysis is to provide the experimenter with a list of good candidate genes to follow up, where a majority (say more than half), are really different (true positives). Another way to say this is that the expected number of false positives is some manageable fraction (say less than .5) of the genes selected. Here statistical significance is a guide to better screening. This question leads us to specify the false discovery rate (FDR), rather than significance level (p-value).

What Tests are Appropriate to Microarray Data?

Outliers and systematic biases are common in expression data. Many kinds of statistical tests work well with normally distributed data, and but give falsely extreme p-values when the distribution of values itself has frequent extreme values. The p-values from a t-test are within 5% most of the time under a skewed distribution, but give erroneously small or large p-values when there are more than 10% 2-fold outliers, which is not unusual with microarray data. Also errors in microarray data are correlated; very few statistical tests give accurate results in this case. What kinds of tests are reliable under these circumstances?

One approach is to use non-parametric tests throughout. Non-parametric tests deliver conservative p-values for all kinds of distributions, and their p-values are generally insensitive to outliers. However they are less likely to pick up regulated genes, than parametric tests; furthermore even non-parametric tests will give inaccurate p-values in the presence of correlated errors. If there is a single approach that is widely applicable and copes with all these problems it is permutation tests, which are simple and easy to program.

To do a permutation test, you

i) choose a test statistic
ii) compute the test statistic for the groups
iii) permute the labels on samples at random, and recomputed the test statistic for the rearranged labels. Repeat for at least 1,000 permutations
iv) compare the true test statistic to this distribution.

A common test statistic for comparing treatment vs. control is the t-statistic.

$$t_i = \frac{\bar{x}_{i,group\ 1} - \bar{x}_{i,group\ 2}}{SD_i},$$

where SD is the standard deviation for gene i.

Some authors use the following test statistic s rather than the usual t statistic, in order to reduce the number of statistically significant genes, which change only a little.

$$s_i = \frac{\bar{x}_{i,group1} - \bar{x}_{i,group2}}{SD_i + q_{\alpha,SD}},$$

where q_α is the a-th quantile of standard deviations.

What is a P-Value?

Most tests give p-values, however their meaning is not often discussed. A p-value refers to something in a possible world. The usual (single test) p-value is the probability of the observed test statistics, if there is no real difference in any gene among groups (this assumption is called the null (as in default) hypothesis). If one decides that a difference occurs, whenever a test statistic passes a certain threshold, then the p-value is the answer to the question: how often would the numbers deceive us? How often would random sampling from the null distribution give test statistics as extreme as observed? When you decide an effect is significant at 5%, you say you are willing to be wrong once in twenty decisions that there is a difference. (We don't often cross the street on a 95% confidence that there is a break in traffic.)

Multiple Testing Corrections and False Positives

Suppose you compare two groups of samples with no real differences, using a chip with 10,000 genes on it. For some genes, the variation between samples will be large relative to the variation within groups due to random, but uneven distribution of the genes; ie. 500 will appear 'significantly different' at a 5% threshold. Therefore the p-value appropriate to a single test situation is inappropriate to presenting evidence for a set of changed genes.

Statisticians have devised several procedures for adjusting p-values to correct for the multiple comparisons problem. The oldest is the Bonferroni correction; this is available as an option in many microarray software packages. The corrected p-value, p_i^* for gene i is set to:

$p_i^* = Np_i$, if $Np_i < 1$,

1, if $Np_i > 1$.

This is correct, but too conservative. In practice, few genes meet this strict criterion, including many known to be differentially expressed from other work.

Another procedure is the Sidak correction:

$p_i^* = 1 - (1 - p_i)^N$.

The Bonferroni correction is a conservative approximation to the Sidak: expanding $1 - (1 - p_i)^N = 1 - (1 - Np_i + \ldots)$ gives Bonferroni.

This is exactly correct if all genes are independent, however it is still too conservative if test statistics are correlated (ie. genes are co-regulated).

To give some idea of how to approach the problem in the realistic case when genes are correlated, imagine an extreme case: if all genes were perfectly correlated. In that case all tests are identical, and p-values for one are p-values for all; the multiple-comparisons correction changes nothing. In reality typical gene data is highly correlated: one factor may account for as much as half the variance. The multiple corrections correction for correlated data should be weaker than for independent data, while stronger than that for identical data. The number of extreme test statistics will be more variable than with independent data, although it will have the same long-run average. More sensitive tests are possible if we can generate an accurate joint null distribution of p-values.

This is illustrated below

Null distribution from independent genes		
.5	.3	.9
.7	.03	.1
.4	.9	.05
.6	.8	.4
.2	.2	.9

Null distribution from perfectly correlated genes		
.5	.3	.9
.5	.3	.9
.5	.3	.9
.5	.3	.9
.5	.3	.9

Null distribution from highly correlated genes		
.5	.3	.9
.45	.2	.95
.65	.25	.8
.4	.35	.75
.5	.4	.85

Figure. P-values from genes under null hypothesis, under various degrees of correlation

The average number of genes exceeding the .05 threshold in the long run is always 5%. If genes are independent, then (roughly) 5% of genes exceed the .05 threshold, 100% of the time. If genes are perfectly correlated, then 0% of genes exceed the .05 threshold, 95% of the time, and 100% of genes exceed the .05 threshold 5% of the time. In a realistically correlated situation, for example, 2% of genes exceed the .05 threshold, 90% of the time, and 40% of the genes exceed the .05 threshold, 10% of the time. In that case the corrected p-value when 2% of the genes exceed the .05 threshold should be 10%.

This gives us an approach to correcting for multiple testing: for a group of potentially significant genes, we ask how often would a group this size appear significant? To be exact: for a specific number k and a threshold α, how likely is it that at least *k* single test p-values will fall under the threshold for significance level α?

Calculating Permutation-Based P-values

To calculate corrected p-values, first calculate single-step p-values for all genes: $p_1, \ldots, p_N$. Then pick a set of m genes, which appear interesting to you; order the smallest m p-values: $p_{(1)}, \ldots, p_{(m)}$, from least to greatest. Now permute the labels at random relative to the samples and compute the test statistic between (randomized) groups. For each $k < m$ keep track of how often you get k p-values less than $p_{(k)}$. After all permutations, for each value of k compute the fraction of permutations with k p-values less than $p_{(k)}$. This is the corrected p-value. This procedure is more powerful than the other corrections, in that it gives a bigger list of significant genes at any specified risk of false positives. It is implemented in the Bioconductor package multtest. See Ge, et al, Test, 2003