An (opinionated) Guide to Microarray Data Analysis

Mark Reimers,
National Cancer Institute, and
Karolinska Instutitute, 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. The focus is on practical biological issues. There is no attempt to cover all the fancy data mining algorithms that have been developed. However I hope that statisticians and programmers will also find interesting material here.

This paper is organized as follows:

**Experimental design**
This section discusses how many replicates are needed, whether to pool samples, and designs for two color arrays.

**Pre-processing of Spotted Arrays**
This section discusses steps along the way from quantification of the image, through background correction, quality control, and normalization, to obtain reliable estimates of relative gene abundance in the samples.

**Pre-processing of Affymetrix Chips**
The Affymetrix multiple-probe system offers a unique set of statistical challenges before one even gets to interpreting biological meaning. This section discusses image quantification and background, MAS5.0, normalization, multi-chip algorithms for estimation, and model-based quality control.

**Exploratory Analysis**
A preliminary examination of data confirms that groups are homogeneous. Many studies aim to find unknown co-regulated genes. This section discusses how to achieve these goals using multi-dimensional scaling, and clustering.

**Statistical Tests for Identifying Differentially Expressed Genes**
This section discusses methods and problems in determining which genes are differentially expressed between groups of samples, covering t-tests, multiple-testing corrections, false discovery rate, empirical Bayes methods, and analysis of variance. For descriptions of the technology see the Nature Genetics Chipping Report supplements.
Design of Microarray Experiments

How many replicates is enough? Should you pool samples? What is good design for a cDNA experiment?

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 depending on resources and long-range plans. That said, some common sense principles apply across the board (but are ignored more often than they should be).

If your goal is to make a series of experiments and to compare the results, ensure that the designs and hybridisation conditions are similar. Conditions such as RNA preservation medium, the protocols of hybridisation, and even regional ozone levels, 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 hybridisations, you want to prepare enough common reference to serve for all experiments. Chip failures are common, and it’s wise to prepare more labelled cDNA than you expect to use, if that is possible. Some of the more efficient designs will lose much information if a single hybridisation 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.

Replicates

How many microarrays is enough?

How many microarrays is enough? Statisticians don’t like simple answers to this question; it depends on the goals of the study, the resources, and the reliability of the technology – specifically how accurate the chips are, and how often a hybridization fails.

However the following guidelines apply to most situations. If an exploratory study aims to find large (more than two-fold) differences between two conditions, then a design with three samples per condition is usually adequate. If the aim is to find smaller differences, or almost all of the large differences, then five samples per group are necessary to obtain sufficiently reliable enough estimates of variation among samples within conditions, in order to distinguish true differences between conditions. This applies to both treatment and control conditions.

Six samples per condition allows meaningful permutation tests, which can give more accurate, and less conservative, estimates of p-values and false discovery rates. If there are more than two conditions, and the treatments do not drastically alter the cell physiology, then the number of samples within any one condition can be somewhat less; with four or more conditions, one can obtain reasonable estimates of within-condition variation with only four samples per condition.

All of these suggestions assume that there are no outlying samples, which should be discarded; it’s wise to do one or two more per condition in clinical situations, where outliers occur commonly, and it’s safer to do one more for animal experiments, where sometimes one animal in a condition appears very different than all the others.

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 the only way to estimate the noise is to do three or four replicate hybridizations. For Affymetrix systems with the best analysis (see ), we find 3 to 5 chips per group gives useful information. Usually many more cDNA chips are needed for comparable levels of accuracy. To estimate replicability of a two-color chip, hybridize three pairs of replicate dye-swaps (6 chips) using the same two (different) RNA samples.

To do meaningful clustering requires at least 20 samples, and generally more many. The key issue for clustering genes is how many different types of samples there are, because the different conditions expose the correlations in gene regulation. It’s not useful to try to cluster genes from only two groups, as is sometimes done, and rarely useful to cluster genes from a study of fewer than five groups.

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 a gene among different individuals is approximately normally distributed, then pooling n independent samples would result in reduction of variance (s2) given by the formula:

$$\sigma^2_{\text{Pool}} = \frac{s^2}{n}$$

where $s^2$ is the variance of the expression estimates of any one gene across samples. In principle we could then reduce further the variation by making replicates of the pool, and hybridising to replicate arrays. Since technical variation is usually less than (roughly half of) individual variation, this strategy would in theory give us more accurate estimates of the group means for each gene.

Figure 11 Pooling

In practice the expression levels of many genes among individuals are not roughly normal; often there are more very high values (outliers) than the normal distribution. Some individual 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 long periods before preservation, which allows cells to respond to stress (Prichard et al ‘Project Normal’, PNAS (2002)).

It is easier to detect this, if individual samples are hybridized. In some studies (Terry Speed; unpublished data), where the same samples were analysed by pooled and unpooled designs, the majority of genes that were identified as differently expressed between two groups, turn out extreme in only one individual. Also, if one pools samples, there is no way to estimate variation between individuals, which is sometimes important and often interesting.

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 if the study aims to compare each of several samples against all others, there is no natural control. A reliable alternative is a common reference obtained by pooling all samples.

This enables samples to be compared with each other indirectly. A pooled reference sample reduces the number of extreme gene ratios (which have large errors) on each chip. Some labs take this further and create a ‘universal reference’: a pool of mRNA derived from several standard cell lines, which they use most often in their experiments. Using a universal reference enables them to compare results for all their experiments.
Designs for Two-Color Arrays

The most common design for two color (competitively hybridised spotted) arrays is the ‘reference design’: each experimental sample is hybridised against a common reference sample. Although this effectively means that only one sample of interest is hybridised per chip, the reference design has several practical advantages over more efficient designs:

- it extends easily to other experiments, if the common reference is preserved;
- it is robust to multiple chip failures; and
- reduces incidence of laboratory mistakes, because each sample is handled the same way.

We may represent designs by pictures where circles represent samples, and arrows represent chips; the red and green ends of the arrows represent the dyes used for the samples at either end.

![Figure 1.2](image1.png)
*Figure 1.2. A reference design: the red and green arrows represent chips.*

One complication in two-color arrays is that the two dyes 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-swapped hybridizations 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.

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, from a theoretical perspective. 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. This problem is so serious in practice that many microarray statisticians don’t recommend the loop design.

![Figure 1.3](image2.png)
*Figure 1.3. A loop design: arrows represent chips with samples labelled as indicated.*

There are many efficient direct designs, which are also robust to failure, based on ‘round-robin’ style contrasts where each sample is hybridised to a specific subset of all the others, in a balanced fashion.

These designs are appropriate where differences between any pair of samples are all equally important, and the experimenter does not plan to compare the expression values directly with other experiments in a longer series.

The simplest of these designs for a small number of samples, is a ‘saturated’ design: to hybridise every contrast exactly once. It is fairly easy to balance the dyes with three or five samples; with four or six samples, it is not possible to exactly balance the number of times each sample is labelled red and green.

![Figure 1.4](image3.png)
*Figure 1.4. A saturated design.*

A more common situation is that some contrasts are more important than others. For example to investigate the role of a receptor, one prepares wild-type and mutant (eg. knockout) animals, and then administers a treatment (eg. a ligand) to half of each group, while giving a non-effective vehicle to the other half. Then there are four groups, and the contrasts of most interest are the effects of the ligand on the two groups (WT and KO); the contrast between WT and KO animals in each treatment group is less important, and the contrasts between WT treated against KO control, and vice versa, are uninteresting.

A good design for this is to hybridise several dye-swap pairs between the treatment and control within each group, and perhaps to hybridise one or two slides between WT treated and KO treated, and between WT control and KO control.

This design gives fairly accurate estimates of both effects of treatment vs. control (in WT and Mutant), which enables accurate comparisons between the effects; there is less accurate information about the direct comparison between WT and mutant, although in effect there is more than one slide’s worth of information, because there are several indirect paths to make the comparisons between WT treated and mutant treated, for example.

![Figure 1.5](image4.png)
*Figure 1.5. A design for a comparative study of the effect of a treatment on two biological strains.*

A good general framework for estimating the abundances, and contrasts, is provided by using a linear statistical model. For example, the log ratios in figure 1.5 are each the difference of the log abundance in two of the four samples; we may construct a design matrix which specifies how the log ratios in all ten experiments are derived from the log abundances in the four chips; then the best estimate of the log abundances is obtained by solving the least squares problem for this design matrix.

Since the equations in the are all differences, adding a constant to any one solution will give another solution; therefore a set of equations specifying that the sum of the log abundances is 0 (or some other constant) is needed to obtain a unique answer.

The design matrix for figure 5 are shown in figure 6.

![Figure 1.6](image5.png)
*Figure 1.6. Design Matrix.*
Preprocessing of Two Color Competitively Hybridized Arrays – From Image to Estimates

This section covers the low-level preprocessing steps from the point at which the chip is scanned, to obtaining reliable estimates for the relative gene abundances of each gene in all of the samples. Broadly these steps may be classified as image analysis, quality control, background correction, and normalization, although all of these procedures are inter-dependent, and not always done in this order.

Image Quantification
Although the lab technician usually handles this step, using the default settings on an image quantification program, the program and the settings can have a noticeable impact on the noise level of the subsequent estimates. There are several steps in image quantification:

- i) laying the grid – finding where the printed spots ought to be in the image
- ii) identifying the extent of each spot, and separating foreground from background
- iii) summarizing the varying brightnesses of the pixels in the foreground of each spot
- iv) dealing with scanner saturation
- v) dealing with variable backgrounds

The technician does the first step interactively. The quantification program must deal with several problems in the second step. It is rare that the probes are uniform in size and shape; most programs try to adapt to the sizes, and sometimes shapes, of individual spots.

Sometimes (poorly stabilized) neighboring probes bleed for some region around which may invert the relation between foreground and background (see figure).

The different programs each try different approaches to these problems, and they make a difference in the reproducibility of gene expression measures. A study done in California showed considerable differences among results from different quantification programs applied to 8 arrays that compared the same two samples. Several different settings were used on most programs. Since ratios should be identical, the standard deviation is an (inverse) measure of quality.

![Figure 2.1. Some problems that quantification must deal with.](image)

Figure 2.1. Some problems that quantification must deal with.

Background Correction
The previous figure shows that background subtraction added to the noise in those measures, and many researchers have had similar experience. On the other hand it seems in principle wrong to ignore background, and some substrates (eg. poly-L lysine) show substantial additive background; ignoring this leads to substantial bias in the estimates of most gene ratios.

At the moment there is no consensus, but two suggestions may help for the time being. It may be pragmatic to ignore the background correction, if the goal is to detect a few differentially expressed genes among the multitude of noisy similarly expressed genes. For this purpose noise control is most important. On the other hand, when it comes time to estimate the fold-change, subtracting background, or some other form of background correction gives more accurate estimates. Perhaps the resolution of this paradox will come with more sophisticated forms of background correction.

The Spot program and the Agilent software both do something different than local background estimation, and both give less noisy estimates than subtraction. In principle the raw intensity of a spot is made up of fluorescence from labeled transcripts, plus reflection or emission from the substrate.

The label fluorescence comes from the target transcript and also a mixture of other transcripts that have bound non-specifically to the spot. The local background is made up of the reflection from the substrate. and stray bits of labeled transcript that have bound to the surface; sometimes if a nearby probe has spread, it includes labeled transcript from that other gene.

The make-up of the local background differs from the background we want to correct on the spot. On the other hand the negative controls show just non-specific hybridization, which is the bulk of what we want to correct on each gene probe. Perhaps a better way would be to subtract a weighted average of the local background and the values of negative controls nearby.

Quality Control
A lot of the messy business of statistics is cleaning up data. Although this is less exciting than higher-level analysis, it makes as much of a difference to the results as 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. Researchers from Agilent have recently confirmed that even moderate levels (5ppb) of ozone can degrade Cy5, while not affecting Cy3. 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 sample RNA and the labelling pass the wetlab quality checks, then the best information about the process of hybridisation comes from the controls. There’s no excuse for chips without a well-designed set of negative and positive control probes. Negative controls are probes designed for DNA sequences that should never occur in your sample. Positive controls are replicate 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) to the samples before labelling.

The negative controls 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 will typically have different means). The signal from negative controls gives an idea of the background in all signals due to non-specific hybridization. You won’t be able to estimate reliably those gene abundances, whose signals are comparable to the signals from negative controls, even if they are above the local background.

Positive controls give some idea of the spatial variation in hybridization. Probes for the same gene should show fairly uniform intensities across the chip. If the positive controls are very different from their average in some region, it is worth taking a closer look, and perhaps discarding all signals from that region. It is common to see spatial gradients in intensity, and sometimes in ratio. 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. Then one end of the chip is brighter than the other; this is not a serious problem unless the log ratios show a similar gradient.

In data from poorly functioning hybridization stations one often observes uneven signal 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.

Spike-in controls (such as Amersham’s) give some idea of the accuracy and linearity of the intensities. Some genes are added in ratios of 3:1 or 10:1 to the two samples. Typically one sees that the ratios as reported are squashed, and sometimes that the low intensity genes show up stronger than the majority. These can be very useful additional information during normalization, but they shouldn’t be taken too literally, as they rarely work exactly as expected.

Quality Control of Individual Probes
If the samples and the hybridisation pass the previous tests, the next step is QC for individual spots or probes. Spot-level QC detects mostly printing problems rather than hybridization anomalies. 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 and Quantarray give many). Some reported measures are often: spot area, uniformity (standard deviation of foreground), and background uniformity. It is not practical to examine thousands of spots individually; an automated filtering procedure is what’s needed. However the filtering criteria that are useful for one experiment, are too slack or too strict for the next; there aren’t rules about spot size, or background that apply across the board to all chips under all circumstances.

A sensible thing to do before filtering is to examine the distribution of the various measures across the chips for each new experiment. Then identify the ‘normal’ ranges for each of these variables, and what are unacceptable. Then discard (or down-weight) all those spots that fall outside the ‘normal’ range. This is best done in collaboration with a good core facility.

The printer often drops small amounts of probe, elsewhere than intended. This becomes a problem if a spatter of probe for a particular gene lands on a probe for a faint gene; then the signal from both channels reflects the abundant gene, rather than the gene that is annotated at that position. Another type of problem is spot formation – printers aim to deliver fairly round, high quality spots, which can be very useful additional information during normalization, but they shouldn’t be taken too literally, as they rarely work exactly as expected.

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, \( q_1, q_2, q_3, q_4 \) 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.

Reference:
Effects of Atmospheric Ozone on Microarray Data Quality
Thomas L. Fare, Ernest M. Coffey, Hongyue Dai, Yudong D. He, Deborah A. Kosellar, Kristopher A. Kilian,* John E. Koch, Eric Lefkowitz, Matthew J. Martin, Michael R. Meyer, Roland B. Stoughton, George Y. Tokiwa, and Yanqun Wang

Anal. Chem., ASAP Article 10.1021
Normalization of Competitively Hybridized (Two-Color) Microarrays

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 r-PCR. In practice methods that have worked well for these types of measures do not perform as well for microarray data, which shows many dimensioned systematic differences.

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.

Systematic non-biological differences between chips are evident in several ways:

- Total brightness differs between chips
- One dye seems stronger than the other (in 2-color systems) on one chip, but not on another
- Typical background is higher in one chip than on another.

There are also many non-observable systematic differences between chips in an experiment, and even between the two channels on a single array. 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 in one array than the other
- Hybridisation conditions may vary across an array
- Scanner settings are often different, and of course
- Murphy’s Law predicts even more variation than can be simply explained.

In order to identify the real biological differences, we attempt to correct for the systematic technical differences in measurement. Although the aims of normalization for all arrays are similar, the issues and techniques used in normalization of two-color arrays differ from those useful for normalization for Affymetrix arrays.

Housekeeping Genes

One early approach was to find a standard gene invariant across all chips or samples. This the commonsense approach, used routinely in r-PCR. The standard genes tried were ‘housekeeping’ genes – genes, required in all cell types – on the theory that they occur at near equal levels in all cells. However, these housekeeping genes seem to vary by 30% or more across interesting samples <ref>. The model is sufficiently accurate for rPCR, where one cycle corresponds to an increase of a factor of two. However in a microarray study a 30% difference over the whole genome is enormous.

Quantitative Approaches

Most approaches to normalizing expression levels assume that the overall distribution of RNA numbers doesn’t change much between samples, and that most individual genes change very little across the conditions. This seems reasonable for most laboratory treatments, although treatments affecting transcription apparatus have large systemic effects, and malignant tumours often have dramatically different expression profiles. If most genes are unchanged, then the mean transcript levels should be the same for each condition. An even stronger version of this idea is that the distributions of gene abundances must be similar.

Statisticians call systematic errors, which affect a large number of genes, ‘bias’. Keep in mind that normalization, like any form of data ‘fiddling’ adds noise (random error) to the expression measures. You never really identify the true source or nature of a systemic bias; rather you identify some feature, which correlates with the systematic error. When you ‘correct’ for this feature, you are adding some error into the measure for those samples where the feature you have observed doesn’t correspond well with the true underlying source of bias. 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 compensate fully.

Scaling by Brightness

The simplest approach posts that total abundance of all genes is equal in the two samples on any one chip. Scaling a chip means multiplying the signals (intensity measures) for all genes by a common scale factor. This makes sense if equal weights of RNA from the two samples are hybridised on the array. The sizes of the RNA molecules are equivalent, and the number of RNA molecules should also be the roughly the same in each sample. Consequently, the approximately the same number of labeled molecules from each sample should hybridise to the arrays and, therefore, the total of the measured intensities summed over all elements in the arrays should be the same. For a single chip compute scale factors $C_{\text{red}}$ and $C_{\text{green}}$ by:

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

where $f_i^{\text{red}}$ represents the measured intensity of array element $i$ in the red channel, and $N$ is the total number of elements represented in the microarray. Individual ratios are then scaled by their sum:

$$
\frac{f_i^{\text{red}}}{C_{\text{red}}} = \frac{f_i^{\text{green}}}{C_{\text{green}}}
$$

After this operation, the intensity of genes in each color is equal to one, while individual intensities are inconveniently small. Often researchers choose one channel (eg. Green) to be the standard, and multiply the other by a scaling constant (eg. $C_{\text{green}}/C_{\text{red}}$). The result is that the mean of gene expression values is the same in both channels, and that the mean difference (mean of all substracted intensities) is 0.

Sometimes this operation is done on a logarithmic scale, which has a somewhat different result: a mean log-ratio equal to zero; this means the (geometric) mean of the individual gene ratios is equal to 1. Done on a logarithmic scale, this operation is equivalent to subtracting the average log-ratio from all the individual log-ratios.

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 on one chip, but not on the other. More consistent results are obtained by using a robust estimator, such as median or one-third trimmed mean. To do the latter: compute the mean of the middle two-thirds of all probes in the red, and the green channels, and scale all probes to make those means 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-ratios within one chip, it is common to find also that the variance of the log-ratios differs between arrays. One approach to dealing with this problem is to scale the log2(ratio) measures (after scale normalization within chips) so that the spread (measured by the variance) of the log-ratios of genes is the same for all chips. This is an example of over-correcting a bias. This procedure usually works in reducing overall variance between log-ratios between chips, but sometimes the variability of many genes is actually increased. This approach is not widely used.

Intensity Dependent Normalization with Lowess

The scale normalization adjusts for overall dye bias. Terry Speed’s lab identified an intensity-dependent dye bias, and introduced a popular method for adjusting it. One commonly observes that the log2(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 – so this must be an artefact. It appears that the explanation is chemical: the two dyes don’t give off equal light per molecule at different concentrations.

This is due to ‘quenching’: a phenomenon where dye molecules in close proximity, re-absorb light from each other, thus diminishing the signal. The amount of re-absorption changes with concentration differently for the two dyes.
The easiest way to visualize intensity-dependent effects is to plot the measured log2(Ri/Gi) for each element on the array as a function of the log2(RiGi) product intensities. This ‘R-I’ (for ratio-intensity) plot can reveal intensity-specific artifacts in the log2(ratio) measurements.

Note that Terry Speed’s group calls these variables ‘M’ and ‘A’, (for ‘minus’ and ‘add’ – on the log scale) and they call the plot an ‘MA plot’.

![Figure 2.5](image)

**Figure 2.5.** 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 log2(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 \((x_1,y_1),...,(x_N,y_N)\), we calculate at each point \(x_i\), 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 \(x_i\). We can think of the calculated value as a kind of local mean. For each observation \(i\) on a two-color chip, set

\[ x_i = \log2(R_iG_i) \] 
\[ y_i = \log2(R_i/G_i) \]

The lowess approach first estimates \(y^*(x)\), the value of the regression line through points having similar intensities, then subtracts this from the experimentally observed ratio for each data point.

The normalized ratios \(r^*\) are given by:

\[ \log2(r^*) = \log2(R_i/G_i) – y^*(\log2(R_iG_i)) \]

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

![Figure 2.6](image)

**Figure 2.6.** As in Figure 2.5, 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 ‘sub grid’).

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 local differences in hybridization conditions across the array. However such a procedure may over fit the data, reducing accuracy, especially if the genes are not randomly spotted on the array. Terry Speed’s group makes the argument that a cell grid should have average expression ratios of 1, and that several hundred probes are in each cell. 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.

There is still not a consensus about the best way to do local normalization.

**Quantile Normalization**

A good design will place all contrasts of interest directly on chips, but sometimes that is impossible, or the afterwards the experimenter wants a contrast that wasn’t planned before. This requires comparing ‘parallel’ measures in a single channel between arrays. So many sources of systematic variation make such comparisons very difficult: 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 compensating the intensity-dependent dye bias, as well as does lowess normalization.
**Pre-processing of Affymetrix Chips**

**Image Quantification and Background**

The image scanning algorithm lays down a grid over the image, to identify squares in which the probes should be, and then selects the brightest subsquare (usually 4x4, 4x5, or 5x5 pixels) as the probe. The scanning software reports the average pixel intensity of this subsquare, and its standard deviation, which is usually 5% - 20% of the mean.

MAS 5.0 computes local background in each of 16 squares, and then subtracts a weighted combination of these background estimates from each probe intensity.

**Description of Affymetrix Probe Sets**

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?

![Image 3.1](image)

Figure 3.1. 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.

**Normalization of Affymetrix Chips**

**Why Normalize?**

Biologists have long experience coping with systematic variation between experimental conditions (technical variation) that is unrelated to the biological differences they seek. 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, have similar effects on any two chips.

The performance of expression arrays can vary in more ways than measures such as r-l-PCR. Normalization methods that have worked well for these types of measures do not perform as well for microarray data.

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. This is much better than using a single housekeeping gene, and probably adequate for about 80% of chips in practice.

Most approaches to normalizing expression levels assume that the overall distribution of RNA numbers doesn’t change much between samples, and that most individual genes change very little across the conditions. This seems reasonable for most laboratory treatments, although treatments affecting transcription apparatus have large systemic effects, and malignant tumours often have dramatically different expression profiles.

If most genes are unchanged, then the mean transcript levels should be the same for each condition. An even stronger version of this idea is that the distributions of gene abundances must be similar. Statisticians use the term ‘bias’ to describe systematic errors, which affect a large number of genes.

Keep in mind that normalization, like any form of data ‘fiddling’ adds noise (random error) to the expression measures. You never really identify the true source or nature of a systemic bias; rather you identify some feature, which correlates with the systematic error. When you ‘correct’ for that feature, you are adding some error to those samples where the feature you have observed doesn’t correspond well with the true underlying source of bias. 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 compensate fully.

**Normalization by Scaling and its Limitations**

The simplest approach to normalizing Affymetrix data is to re-scale each chip in an experiment to equalize the average (or total) signal intensity across all chips. The reasoning behind this is that there should be with equal weights of RNA for all the samples; 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, nearly the same number of labeled molecules from each sample should hybridise to the arrays and, if all other conditions were equal, the total hybridisation intensities summed over all elements in the arrays should be the same for each sample. Of course, other conditions aren’t equal; hence normalization restores this equality. In practice, for a series of chips, define normalization constants \( C_1, C_2, \ldots \), by:

---image---, and so on, where the numbers \( P_{x,y} \) are the fluorescent intensities measured for each probe on chip \( x \). Select a common total intensity \( K \) (e.g. the average of the \( C_i \)'s). Then to normalize all the chips to the common total intensity \( K \), divide all fluorescent intensity readings from chip \( i \) by \( C_i \), and multiply by \( K \). Variants of this approach, scaling all probes on a chip by the chip’s trimmed mean intensity, or by median intensity, are widely available in commercial software.

To do better, we examine in detail the relationships among replicate chips (chips hybridized to the same sample). Figure 1 shows a scatter plot 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 on a log scale; these distributions have very different shapes; on a log scale, applying a scaling transform to a chip, shifts its distribution curve to the right or left, but doesn’t change its shape. Figure 5A shows Ratio-Intensity plots of pairs of Affymetrix chips hybridized with replicate samples (so all log-ratios should be 0); a scaling transform will shift the R-I plots up or down, without changing their configuration. For perhaps 80% of chips, the relationship between probe distributions is close enough to linear that a scaling transform will bring the chips into line. For the other 20% of cases something different is needed.

![Image 3.2](image)

Figure 3.2. 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.
Normalization by MAS 5.0

MASS does slightly better than scaling 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%), and fit a regression line to the middle 98% of probes (i.e., estimate slope and intercept; two parameters). Transform the values in each probe, by subtracting the intercept, and dividing by the slope, so that the regression line becomes the identity \( y = x \) line.

A better 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: \( p_1, \ldots, p_K \) (\( K \approx 10000 \)), 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

\[ x_{\text{norm}} = F_i^{-1}(F_{\text{ref}}(x)) \]

where \( F_i \) is the distribution function of chip \( i \), and \( F_{\text{ref}} \) is the distribution function of the reference chip.

Distribution of Chip Intensities
Reference Distribution

Density function
Cumulative Distribution Function

\( F_1(x) \)
\( F_2(x) \)
\( \alpha \)

Figure 3.3. Den sity of PM probe signals on 23 different chips from GeneLogic spike-in experiment (Courtesy of Terry Speed)

Figure 3.4. Schematic representation of quantile normalizat on: the value \( x \), which is the \( \alpha \)-th quantile of all probes on chip 1, is mapped to the value \( y \), which is the \( \alpha \) quantile of the reference distribution \( F_2 \).

If \( F_1 \) and \( F_{\text{ref}} \) are fairly similar in shape, then in practice this transform is not too different from a straight line, which is what a scaling transform looks like; see Figure 4. However the transform is strong enough to cope with the non-linear ratio-intensity relationships revealed in figure 5A; see figure 5B after quantile transformation.

Figure 3.5. The effects of quantile normalization on raw probe values in three chips. Raw values are on x-axis, normalized values on y-axis. Often this transform looks very much like a scaling transform (nearly linear), but sometimes it is quite non-linear.

Figure 3.6. Ratio Intensity Plot of all probes for four pairs of chips from GeneLogic spike-in experiment

Figure 3.7. As in A, after normalization by matching quantiles. Both figures courtesy of Terry Speed

This form of normalisation also reduces noise among replicate measures of the same samples, compared to normalization by scaling, as shown below in figure 6.

Figure 3.8. Each dot represents one probe on an Affymetrix U95A chip. On the y-axis is the ratio of variance across a set of replicates after quantile normalisation, divided by the variance of the scale-normalized values. On the x-axis are the mean levels. Both axes on log scale.

The main drawback of this approach to normalization is the strong assumption that the distributions of probe intensities are identical (even if individual probes differ in their positions in the distribution). This is true for low abundance genes, and to a fairly good approximation for genes of moderate abundance, but certainly not true for the few high-abundance genes, whose typical levels vary noticeably from sample to sample.
Combining Probe-Level Signals into Gene Abundance Estimates

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 <LINK>.

Estimation by Affymetrix MicroArray Suite

Affymetrix has upgraded their MicroArray Suite (MAS) software several times over the short history of their product. MAS 4.0 was the standard until January 2002 and is still cited in published papers. MAS 4 calculates a weighted average of the probe-pair differences (PM – MM) for each probe pair representing a gene.

MAS 5.0 improves in two important ways. First the intensities are transformed to a logarithmic scale before the average is taken; this equalizes the contribution of different probes. Secondly an estimate of background based on MM itself in the difference PM-MM; this estimate is itself a weighted average of log probe pair differences: log(PM/MM). The “Statistical Algorithm Description Document” from Affymetrix, has more details.

The idea of averaging different probe intensities for the same gene is seems quite wrong. The individual probes in a probe set have very different hybridization kinetics. Taking an average of their intensities, is like averaging the readings from scans taken at very different settings. A good algorithm should consider information about probe characteristics, based on the performance of each probe across chips, and use this to make a better estimate. These principles are the basis of the multi-chip models. Affymetrix has seen the evidence, and they are planning to release their own multi-chip model in 2004. MAS 4.0 and MAS 5.0 will soon be of historic interest only.

Multi-Chip 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 that the signals from individual probes move in parallel across a set of chips; the signals have roughly the same pattern across the different samples, as shown in Figure 3.9. The animations of probe sets in dChip show this quite compellingly.

In practice, the discrepancies between real data and the ideal model are frequent outliers, far beyond the usual random fluctuations in signal intensities. These may be due to scratches, or uneven heating, or other artefacts. See Figure 8 in Quality Control. Typically up to 15% of probes in an acceptable Affymetrix chip are outliers. Most standard methods to fit data flounder badly on data with this many outliers. The robust approach is to try to identify the outliers, and exclude or down-weight them.

Constant Variance – the Li and Wong Model and Critique

Li and Wong originally suggested the model

$$PM_{ij} = MM_{ij} = \phi_i a_j + \epsilon_{ij},$$

by analogy with MAS 4.0.

They have found better fits with the model

$$PM_{ij} = \phi_i a_j + c_p, \text{ (PM-only)}.$$

Li-Wong assumes that the noise in all the probe measures is roughly same size. In practice all biological measures exhibit intensity-dependent noise. 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 as an outlier.

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 (1) and find

$$PM = a + \epsilon, \text{ (PM-only)}.$$

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:

$$y = a + \epsilon,$$

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.

At the moment RMA appears to be the best method available. Figure 6 compares the performance of several algorithms on replicate arrays; the smaller the variability the better the algorithm. Four strata of genes are shown: from lowest to highest expression. MAS5 apparently does a decent job on high abundance genes, but the multi-chip models do better on low-abundance genes, which include many genes of interest, such as transcription factors, and signalling proteins.

Figure 3.10. Ideal linear model relationship among intensity (height of green bars), abundance of transcript (a), and probe affinity (\(\phi\)).

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 MAS on the low-abundance genes; this category includes most transcription factors and signalling proteins.
Model-based 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. The outliers are surprisingly good at showing problems in distinct areas of some chips. For example, the following figure 7 was made from dChip, and shows outliers on one chip. These outliers mostly occur in two rings.

Figure 7. Outliers are show in pink

Software Available

Li and Wong’s method is available through their program dChip, at [http://www.dchip.org/](http://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](http://www.bioconductor.org). A commercial software vendor, lobion, has incorporated RMA into their GeneTraffic product.
Exploratory Analysis

Pattern-Finding
Exploratory analysis aims to find patterns in the data that aren’t predicted by the experimenter’s current knowledge or preconceptions. Some typical goals are to identify groups of genes expression patterns across samples are closely related; or to find unknown subgroups among samples. A useful first step in all analyses is to identify outliers among samples – those that appear suspiciously far from others in their group. To address these questions, researchers have turned to methods such as cluster analysis, and principal components analysis, although these have often been used inappropriately.

The first widely publicized microarray studies aimed to find uncharacterised genes, which act at specific points during the cell cycle. 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; the confusion has been perpetuated, since many software packages have catered to this impression. The proper way to analyze data is the way that addresses the goal at which the study was aimed. Clustering is a useful exploratory technique for suggesting resemblances among groups of genes, but it’s not a way of identifying the differentially regulated genes in an experimental study.

Clustering
After that disclaimer, suppose that we want to find groups of similar genes or similar samples, how do we go about it? Clustering depends on the idea that differences between gene expression profiles are like distances; however the user must make (somewhat arbitrary) choices to compute a single measure of distance from many individual differences. Different procedures emphasize different types of similarities, and give different resulting clusters. Four choices you have to make are:

i) what scale to use: original scale, log scale, or another transform,
ii) whether to use all genes or to make a selection of genes,
iii) what metric (distance measure) to use to combine the scaled values of the selected genes, and
iv) what clustering algorithm to use.

Scale
It is not clear what is the most appropriate scale for multivariate exploratory techniques, such as clustering and PCA (see below). Differences measured on the linear scale will be strongly influenced by the one hundred or so highly expressed genes, and only moderately affected by the hundreds of moderate abundance genes; the thousands of low abundance genes will contribute little. Often the high-abundance genes are ‘housekeeping’ genes; these may or may not be diagnostic for the kinds of differences being sought. On the other hand, the log scale will amplify the noise among genes with low expression levels. If low-abundance genes are included (see below) then they should be down-weighted. In the author's opinion, the most useful measure of a single gene difference is the difference between two samples, relative to that gene’s variability within experimental groups: this is like a t-score for difference between two individuals.

Gene Selection
It would be wise not to place much emphasis on genes whose values are uncertain. These are usually those with low signals in relation to noise, or which fail spot-level quality control. If the estimation software provides a measure of confidence in each gene estimate, this can be used to weight the contribution to distance of that gene overall. It’s not wise to simply omit (that is, set to 0) distances which are not known accurately, but it is wise to down-weight relative distances if several are probably in error. A simple general rule is that genes whose signal falls within the background noise range are probably contributing just noise to your clustering (and any other global procedure); discard them.

Metrics
Most cluster programs give you a menu of distance measures: Euclidean, Manhattan distances, and some relational measures: correlation, and sometimes relative distance, and mutual information. The names describe how differences are combined: Euclidean is straight-line distance: (root of sum of squares, as in geometry). Manhattan is sum of linear distances (like navigating in Manhattan). The correlation distance measure is actually 1- r, where r is the correlation coefficient. Probably a more useful version is 1 - |r|; negative correlation is as informative as positive correlation. The mutual information (MI) is defined in terms of entropy: H = \text{Sh}x\log2[p(x)] for a discrete distribution (p). Then MI(g1,g2) = H(g1) + H(g2) – H(g1,g2) 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. The author prefers to use ‘Manhattan’ metrics for clustering samples by similarity of gene expression levels, and to use a correlation measure to cluster genes. 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.

![Figure 4.1](image)

Figure 4.1. 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|.

Clustering Algorithms
Most biologists find hierarchical clustering more familiar, and other algorithms somewhat magical. Statisticians object to hierarchical clustering because it seems (falseely) 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; 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 the user to specify in advance the number of clusters to be constructed.

Of course you don’t know ahead of time, most people end up trying out many values. A criterion that some people use to decide how many clusters to use is to track how much the intra-group variance drops at each addition of another cluster; then going back to the point where the rate of decrease really slows down. More advanced methods allow clusters to overlap, as often happens in biology, or to omit some outlying genes.
Statistical significance of clusters by bootstrapping

An important question, but rarely asked, is whether the clusters obtained from a procedure depend on the overall pattern of gene expression, or on a few samples; they could be very different if one or two samples are omitted. One approach to address this is the Bootstrap: 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.

A variant is known as Jack-knife analysis. Branches in a hierarchical cluster that are supported by a large fraction of the re-sampled data sets are considered fairly reliable. A reasonable figure is 70%, but this is arbitrary, like the often-quoted 5% level of significance.

With any exploratory technique, one should think about what technical variable may underlie the groups discovered this way, before going to the lab to confirm findings. The author finds that clustering most often identifies systematic differences in collection procedures or experimental protocol. These are important but not biologically significant. Even when the difference is biological, it may not be a discovery. Most sets of breast cancer data segregate into ER+ and ER- in clustering, which is re-assuring but hardly news.

Principal Components and Multi-dimensional scaling

Several other good multivariate techniques can help with exploratory analysis. Many authors suggest principal components analysis (PCA) or singular value decomposition to find coherent patterns of genes, or ‘metagenes’, that discriminate groups.

These techniques with a long history in the statistical arsenal rely on the idea that most variation in a data set can be explained by a smaller number of transformed variables; they each form linear combinations of the data, which represent most of the variation, and in principle these approaches, are well-suited for this purpose. However this author believes these approaches are delicate enough that they are not very useful for deep exploratory work; often the largest coherent component, such as the first principal component (PC), reflects mostly systematic error in the data. In fact some researchers have seriously suggested normalizing microarray data, by removing the first PC. PCA is not terribly robust to outliers, which are common. Like cluster analysis, the results of PCA are sensitive to transforms, which are somewhat arbitrary.

These multivariate approaches are more useful for exploring relations among samples, and particularly for a diagnostic look at samples before formal statistical tests. Multi-dimensional scaling (MDS) makes the most useful graphical displays. Classical MDS is identical to PCA for most data sets; however if you fix a dimension, a modern iterative algorithm can place the points in an arrangement that is more representative of true distances, than are the same number of principle components. It’s worth getting the ‘strain’ parameter for the MDS fit; this parameter measures the discrepancy between the distances computed by the metric, and the distances represented in the picture. When this parameter is much over 15%, the picture can be misleading. Often if you omit outlying samples, or take a major subgroup, the picture is a more accurate representation of the computed distances.

When the data conform to expectation the MDS plot shows well-defined groups. Figure ** shows an MDS plot of seven groups in a comparative study; the control group is in black; three of the groups are quite similar to controls and each other, while three others are quite distinct from controls and each other. This experiment showed many genes under very clear regulation.

![Figure 4.2](image1.png)

**Figure 4.2.** Multidimensional Scale Plot of diabetic mouse-data.

Things aren’t always so happy. The following figure ** shows a group of experiments which might have been misinterpreted. Replicate experiments comparing three treatments against three controls were done on two different dates. Results were not consistent between experiments, but the MDS plot shows that we can work with the day 2 data quite confidently, and separately from day 1. The cluster analysis on the left doesn’t show this nearly as clearly.

![Figure 4.3](image2.png)

**Figure 4.3.** Cluster diagram versus MDS plot.

Representing two batches of 3 treatment (T) and 3 controls (C), done on two different dates: 1T3 represents 3rd treated sample on day one. 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. The MDS plot shows that the C samples on day 2 are in fact quite close, whereas the 3 Ts are more disparate but all quite different from the C’s.
Statistical Tests

The Purposes of Statistical Tests

Microarray studies often aim to identify genes that are differentially regulated across different classes of samples; examples are: finding the genes affected by a treatment, or finding marker genes that discriminate diseased from healthy subjects. Statistical tests, rather than cluster analysis, are the right tool for this purpose.

Statisticians have developed many new procedures that seem more useful than conventional tests for microarray data. Some studies also aim to identify groups of genes that act together, or to uncover molecular similarities among subsets of samples. The section on Exploratory Analysis describes appropriate techniques for these studies. Some investigators confirm results found by exploratory analysis by using formal statistical methods, or do cluster analysis on differentially expressed genes. Such procedure can be circular in logic; better procedure is to identify differences between groups defined on biological grounds.

Microarray data is often used as a guide to further, more precise studies of gene expression by qPCR or other methods. Then the goal of the statistical analysis is heuristic: to provide the experimenter with an ordered list of good candidate genes to follow up. Sometimes the experimenter plans to publish microarray results as evidence for changes in gene abundance; in this case it is important to state the correct degree of evidence: the 'p-value'. Many microarray papers still present p-values, as if each gene had been tested in isolation, even though many genes were actually tested in parallel; these p-values are wrong in the context of testing thousands of genes. Perhaps a better way to specify the confidence of microarray results is the 'false discovery rate'.

Transforms

Often the first step is transforming the values to log scale, and doing all subsequent steps on the log-transformed values. Although taking logarithms is common practice, and helpful in several ways, there are other options. The main justification for transforms in statistics is to better detect differences between groups whose within-group variances are very different. Most commonly the within-variance groups are higher in those groups where the mean is also higher. A different kind of variation, the measurement error in expression level estimates, grows with the mean level. If the measurement error is proportional to the mean, then the log-transformed values will have consistent variance for all genes. For both reasons many researchers argue that gene expression measures should be analyzed on a logarithmic scale.

However the gene abundances on the log scale usually show greater variation near the bottom end, giving rise to the characteristic 'funnel' shape of many microarray plots. The log transform is too 'strong', and partly reverses the inequality of measurement variances. A weaker transform, such as a cube root, would bring the variances closer to equal. Some researchers have devised 'variance-stabilizing' transformations, which attempt to equalize measurement errors for all genes. However this is not in principle comparing different genes, but rather the same genes across different groups, and in most experiments, few genes change more than three-fold in mean levels.

For studies where gene levels are fairly constant, and changes are expected to be modest, such as neuroscience studies, there is no need to transform data. For studies of cancerous tissue, where often some genes are elevated ten-fold or more, and these increases are highly variable between individuals, the log transform is very useful.

For studies with at most moderate fold changes -- such as most experimental treatments on healthy animals -- it would be better to use a weaker transform, such as a cube-root transform or a 'variance-stabilizing' transform.

However most people find these less common transforms confusing; thus as a practical alternative, it is worth doing the analysis in parallel in both log and true scale, and discarding genes whose significance disappears entirely under one or the other. However there are other reasons to equalize variances for all genes; roughly equal variances are necessary for SAM and for the empirical Bayes methods described below.

These issues are discussed more fully in the section on Procedures for Microarray Intensities.

Comparison of Two Groups of Samples

The simplest and most common experimental set-up is to compare two groups; for example, Treatment vs. Control, or Mutant vs. Wild type. The issues arising in simple comparisons arise also in more complex settings; it is easier to explain these in the simpler context.

The long-time standard test statistic for comparing two groups is the t-statistic:

$$ t = \frac{(x_i,1 - x_i,2)}{s_i / \sqrt{2}} $$

where $x_i,1$ is the mean value of gene i in group 1, $x_i,2$ is the mean in group 2, and $s_i$ is the (non-pooled) within-groups standard error (SE) for gene i.

Usually the majority of genes are not consistently changed between groups. For the unchanged genes, the values of their t-statistic would follow the t-distribution indexed by $n1+n2 – 2$ degrees of freedom, in repeated trials, there are no outliers in the samples.

As a matter of practice, in almost all cases, the values of the t-statistics for all of the unchanged genes in a single experiment also follow a t-distribution. If the researcher expects only a small number of genes are greatly changed, then a convenient way of assessing the most likely true positives is to plot the t-scores obtained, against the t-distribution. Usually the t-scores of the unchanged genes follow the t-distribution; if the unchanged genes are the overwhelming majority, then this plot will be a straight line at 45o. The t-scores for the few really significant genes are more extreme, and stand out from the straight line, where they can be easily seen. A small R program that does this is here <link>.

A drawback of the t-statistic for microarray datasets is that most experiments have only a few samples in each group ($n1$ and $n2$ are small), and so the standard error $s_i$ is not very reliable. In a modest fraction of cases, $s_i$ is (by normal sampling variation) greatly under-estimated, and genes that are little changed give rise to extreme t-values, and therefore false positives. The theory compensates for some of these false positives, in that the tails of the t-distribution are much further out than those of a Normal curve. This causes a reverse problem: a truly changed gene has to change a great deal to give convincing evidence; therefore many of the moderately changed genes won't show convincing evidence of change.

Tusher and Tibshirani suggested (a form of) the following test statistic $t^*$ rather than the usual t statistic $t^1$. The SE in the denominator is adjusted toward a common value for all genes; this prevents gross underestimation of SE's, and so eliminates from the gene list those genes that change only a little.

$$ t^* = \frac{(x_i,1 - x_i,2)}{(s_{i0} + s_{i0})/2} $$

where $s_{i0}$ is the median of the distribution of standard errors for all genes. The range of values of the revised denominator is narrower than the range of sample SEs; it is as if all the SE's are 'shrunk' toward the common value $s_{i0}$. Unfortunately no theoretical distribution is known for this statistic $t^*$, and hence the significance levels of this statistic must be computed by a permutation method (see below). A similar statistic also crops up in empirical Bayes analysis (see below).

Figure 5.1. A quantile plot.

A drawback of the t-statistic for microarray datasets is that most experiments have only a few samples in each group ($n1$ and $n2$ are small), and so the standard error $s_i$ is not very reliable. In a modest fraction of cases, $s_i$ is (by normal sampling variation) greatly under-estimated, and genes that are little changed give rise to extreme t-values, and therefore false positives. The theory compensates for some of these false positives, in that the tails of the t-distribution are much further out than those of a Normal curve. This causes a reverse problem: a truly changed gene has to change a great deal to give convincing evidence; therefore many of the moderately changed genes won't show convincing evidence of change.

Tusher and Tibshirani suggested (a form of) the following test statistic $t^*$ rather than the usual t statistic $t^1$. The SE in the denominator is adjusted toward a common value for all genes; this prevents gross underestimation of SE’s, and so eliminates from the gene list those genes that change only a little.

$$ t^* = \frac{(x_i,1 - x_i,2)}{(s_{i0} + s_{i0})/2} $$

where $s_{i0}$ is the median of the distribution of standard errors for all genes. The range of values of the revised denominator is narrower than the range of sample SEs; it is as if all the SE’s are ‘shrunk’ toward the common value $s_{i0}$. Unfortunately no theoretical distribution is known for this statistic $t^*$, and hence the significance levels of this statistic must be computed by a permutation method (see below). A similar statistic also crops up in empirical Bayes analysis (see below).

Figure 5.1. A quantile plot.

A drawback of the t-statistic for microarray datasets is that most experiments have only a few samples in each group ($n1$ and $n2$ are small), and so the standard error $s_i$ is not very reliable. In a modest fraction of cases, $s_i$ is (by normal sampling variation) greatly under-estimated, and genes that are little changed give rise to extreme t-values, and therefore false positives. The theory compensates for some of these false positives, in that the tails of the t-distribution are much further out than those of a Normal curve. This causes a reverse problem: a truly changed gene has to change a great deal to give convincing evidence; therefore many of the moderately changed genes won’t show convincing evidence of change.

Tusher and Tibshirani suggested (a form of) the following test statistic $t^*$ rather than the usual t statistic $t^1$. The SE in the denominator is adjusted toward a common value for all genes; this prevents gross underestimation of SE’s, and so eliminates from the gene list those genes that change only a little.

$$ t^* = \frac{(x_i,1 - x_i,2)}{(s_{i0} + s_{i0})/2} $$

where $s_{i0}$ is the median of the distribution of standard errors for all genes. The range of values of the revised denominator is narrower than the range of sample SEs; it is as if all the SE’s are ‘shrunk’ toward the common value $s_{i0}$. Unfortunately no theoretical distribution is known for this statistic $t^*$, and hence the significance levels of this statistic must be computed by a permutation method (see below). A similar statistic also crops up in empirical Bayes analysis (see below).
A drawback of the t-statistic for microarray datasets is that most experiments have only a few samples in each group (n1 and n2 are small), and so a standard error s is not very reliable. In a modest fraction of cases, s is (by normal sampling variation) greatly underestimated, and genes that are little changed give rise to extreme t-values, and therefore false positives. The theory compensates for some of these false positives, in that the tails of the t-distribution are much further out than those of a Normal curve. This causes a reverse problem: a truly changed gene has to change a great deal to give convincing evidence; therefore many of the moderately changed genes won't show convincing evidence of change.

Thus, with some work, using a model-based p-value. The model-based p-value is the probability of the test statistic, assuming that the gene has to change a great deal to give convincing evidence; therefore many of the moderately changed genes won't show convincing evidence of change.

The steps in a permutation-based computation of the significance levels of this statistic tend to be as follows: the permutated sample s gives a test statistic for that gene, at least as extreme as the one that occurs in the properly labelled samples. The idea is that if the gene is distributed similarly in both treatment and control groups, then the difference statistic (a t-statistic or any other) will appear about as big in the permuted arrangement, as in the true arrangement. If the gene levels in the treatment group are higher than any levels in the control group, then no value of the permutated statistic will be as great as the true value.

A permutation test needs at least two groups of six samples, in order to have enough different permutations. For two groups of six, there are C(12,6) = 924 permutations that give different groups; although half of these permutations are mirror images of the other half, so the true number of distinct pseudo-scores is 462.

Some statisticians use balanced permutations: where each pseudo-group has roughly equal representation from both the true treatment and the true control group. The true test statistics typically stand out better from this group of permutations, giving more extreme p-values, but at the cost of requiring larger numbers of samples; for example for two groups of six there are only C(6,3)^2 / 2 = 200 distinct balanced pseudo-groupings.

Permutation Tests

(\[ t^* = \frac{(t_1 - t_2)}{(s + s_0)/2} \])

where s0 is the median of the distribution of standard errors for all genes. The range of values of the revised denominator is narrower than the range of sample SEs; it is as if all the SE's are 'shrunk' toward the common value s0. Unfortunately no theoretical distribution is known for this statistic t*, and hence the significance levels of this statistic must be computed by a permutation method (see below). A similar statistic also crops up in empirical Bayes analysis (see below).

Permutation Tests

Permutation testing is an approach that is widely applicable and copes with distributions that are far from Normal; this approach is particularly useful for microarray studies because it can be easily adapted to estimate significance levels for many genes in parallel. The major drawback for experimentalists is that these tests usually require some programming. Some recent software packages, notably SAM, implement permutation testing in a menu-driven interface.

The meaning of a p-value from a permutation procedure differs from the meaning of a model-based p-value. The model-based p-value is the probability of the test statistic, assuming that the gene levels in both the treatment and control groups follow the model (e.g., a Normal distribution). A permutation-based p-value tells how rare that test statistic is, among all the random partitions of the actual samples into pseudo-treatment and pseudo-control groups. The steps in a permutation-based computation of the significance level of a test statistic are as follows:

1. Choose a test statistic, e.g., a t-score for a comparison of two groups,
2. Compute the test statistic for the gene of interest,
3. Permute the labels on samples at random, and re-compute the test statistic for the rearranged labels; repeat for a large number (perhaps 1,000) permutations, and finally,
4. Compute the fraction of cases in which the test statistics (from iii) exceed the real test statistic from ii).

The p-value for the gene is the fraction of cases in which the randomly permuted samples give a test statistic for that gene, at least as extreme as the one that occurs in the properly labelled samples. The idea is that if the gene is distributed similarly in both treatment and control groups, then the difference statistic (a t-statistic or any other) will appear about as big in the permuted arrangement, as in the true arrangement. If the gene levels in the treatment group are higher than any levels in the control group, then no value of the permutated statistic will be as great as the true value.

A permutation test needs at least two groups of six samples, in order to have enough different permutations. For two groups of six, there are C(12,6) = 924 permutations that give different groups; although half of these permutations are mirror images of the other half, so the true number of distinct pseudo-scores is 462.

Some statisticians use balanced permutations: where each pseudo-group has roughly equal representation from both the true treatment and the true control group. The true test statistics typically stand out better from this group of permutations, giving more extreme p-values, but at the cost of requiring larger numbers of samples; for example for two groups of six there are only C(6,3)^2 / 2 = 200 distinct balanced pseudo-groupings.

Volcano Plot

However one chooses to compute the significance values (p-values) of the genes, it is interesting to contrast the size of the fold change to the statistical significance level. The ‘volcano plot’ arranges genes along dimensions of biological and statistical significance.

The first (horizontal) dimension is the fold change between the two groups (on a log scale, so that up and down regulation appear symmetric), and the second (vertical) axis represents the p-value for a t-test of differences between samples (most conveniently on a negative log scale – so smaller p-values appear higher up).

The first axis indicates biological impact of the change; the second indicates the statistical evidence, or reliability of the 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.

Genes significantly lower in LOH

![Volcano Plot](https://example.com/volcano_plot.png)

Figure 5.2. A volcano plot

Genome-Wide Comparisons, Corrected P-Values, and False Discovery Rates

P-Values and False Discovery Rates

Most scientific papers quote p-values, however few papers discuss their meaning. In order to understand what the problem is with quoting p-values for massively parallel comparisons, we need to be precise. Let’s consider, for example, a t-test of differences between two samples. If there is no systematic (real, reproducible) difference between groups, nevertheless the t-score for differences between groups is never exactly 0.

Common sense cannot decide whether a particular value provides strong evidence for a real difference. The natural question to ask is: how often a random sampling of a single group would produce a t value as far from 0 as the t we observed. When you declare an effect is significant at 5%, you say you are willing to let one false positive sneak in, roughly every twenty tests. We don’t accept this for critical decisions; we won’t long continue to cross the street, if we do so on a 95% confidence that there is a break in traffic. We may call this the false positive rate (FPR); the FPR of a procedure is the fraction of truly unchanged genes which appear as (false) positives.

If the aim of the microarray study is to select a few genes for more precise study, then the goal is an ordered list of genes, most of which are really different (true positives). Another way to say this is that the expected number of false positives is some reasonable fraction (for example less than .5) of the genes selected. This goal leads naturally to specifying the false discovery rate (FDR) for a list, rather than significance level (FPR). The FDR is the expected fraction of false positives in a list of genes selected following a particular statistical procedure.
Multiple Testing P-Values and False Positives

Suppose you compare two groups of samples drawn from the same larger group, using a chip with 10,000 genes on it. On average 500 genes will appear ‘significantly different’ at a 5% threshold. For these genes, the variation between samples will be large relative to the variation within groups due to random, but uneven allocation of the expression values to the treatment and control groups. Therefore the p-value appropriate to a single test situation is inappropriate to presenting evidence for a set of changed genes.

The quantile plot can single out a modest number of genes that are really different, but it is not a rigorous or systematic procedure. For example, it can’t really be applied to large numbers of differentially regulated 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:

\[
p_i^* = \frac{p_i}{N},
\]

where \( p_i \) is the p-value for a single test of gene \( i \), and \( N \) is the number of genes being tested (which may be less than the number of genes on the array). This is correct, but too conservative. In practice, few genes meet this strict criterion, including many, which are known differentially expressed from other work. Some researchers use a related ‘single-step’ adjustment procedure: they find the smallest p-value, \( p_0 \), which is corrected to \( Np_0 \); then the next smallest p-value \( p_1 \) is corrected to \( (N-1)p_0 \), unless this is smaller than \( Np_0 \), in which case it is corrected to the identical value \( Np_0 \); successively larger p-values are corrected in similar fashion. This procedure is still too conservative; in fact since \( N \) is usually over 10,000, and the number of genes is a few hundred, this procedure gives corrected p-values that are almost the same as the Bonferroni. The reason that both these procedures are too conservative is that test statistics are correlated.

To give some idea of why correlation makes a difference, imagine an extreme case: suppose all genes are perfectly correlated, and not changed between groups. In that case the tests for all the genes give identical results: the p-values for one gene are the p-values for all other genes under multiple-comparisons is needed, because all the t-scores either exceed or fall short of the threshold together. For example, if the researcher sets a threshold at the 5% level, then no genes will appear as (false) positives in 95% of all experiments testing for differential expression.

Of course, when false positives occur, a great number occur: in 5% of all experiments, all genes will appear as (false) positives; the false positives are ‘clumped’. In practice real gene expression levels are highly correlated, because genes are co-regulated; hence the probability of false positives is much less than calculated by the Bonferroni procedure; on the other hand, when false positives do occur, they tend to occur in abundance.

For example, doing a t-test on all genes, and selecting those whose t-score is above a threshold of the .05-point of the t-distribution would be such a procedure. If following this procedure in many experiments would give gene lists including twenty percent false positives, on average, then the procedure’s FDR is 20%. The FDR is distinct from the false positive rate (FFPR), which is the rate at which truly unchanged genes appear as false positives. If following a particular statistical procedure on samples with no (really) changed genes gives one positive (false) in every single experiment, then the procedure’s FFPR is 100%.

Related to the FDR is the q-value, estimated by Storey [2]. The q-value is the smallest FDR at which a particular gene would just stay on the list of positives. This is not identical to the p-value, which is the smallest false positive rate (FFPR) at which the gene appears positive. The p-value is much stricter than the q-value. Most of the researchers, who compute significance of genes by permutations, are actually computing the q-value, rather than the p-value.

As of yet no conventions have been established for false discovery rate in published work. An FDR of 5 or 10% should be acceptable for journal publication of gene lists, in keeping with the practice of accepting 5% of erroneous single results. For individual follow-up experiments, the investigator must decide what’s an acceptable waste of time. If the investigator is selecting only half of the genes for follow-up based on a priori biological plausibility, then perhaps a less stringent criterion is acceptable.

Multiple Testing P-Values and False Positives

Calculate Permutation-Based Corrected P-values

To calculate p-values, first calculate single-step p-values for all genes: \( p_1, \ldots, p_N \). Order the p-values: \( p_1, \ldots, p_N \), from least to greatest. Then permute the sample labels at random, and compute the test statistics for all genes between the two (randomized) groups. For each position \( k \), keep track of how often you get at least one p-value more significant than \( p_k \), from gene \( k \), or from any of the genes further down on the list: \( k+1, k+2, \ldots, N \).

After all permutations, compute the fraction of permutations with at least one apparently more significant p-value than \( p_k \). This is the corrected p-value for gene \( k \). Although this procedure is complicated, it is much more powerful than the other corrections: that is, the procedure gives a much smaller corrected p-value for each gene than the Bonferroni procedure, and therefore a bigger list of significant genes at any corrected significance level (specified risk of false positives). This is known as the Westfall–Young correction <refs>.

The number of extreme test statistics, and therefore apparently significant p-values, for correlated genes will be more variable than for independent genes, although it will have the same long-run average. Therefore for realistically correlated data, the multiple testing correction of p-values should be weaker than the correction for independent genes.

If the random experiments are done with the two groups drawn repeatedly from the same cell line; suppose that 2% of the (unchanged) genes appear significant at the .05 threshold, in 90% of the experiments, and 40% of the genes appear significant at the .05 threshold, in 10% of experiments. Suppose we adopt a procedure of selecting ‘differentially expressed’ genes, only if more than 2% of the genes individually appear significant at the .05 threshold; this procedure will select false positives only in 10% of the experiments. That is, the corrected p-value for more than 2% genes at the .05 level, should be 0.10 (10%).

This gives us an approach to correcting for multiple testing: for a group of genes, which appear to differ between sample types, we ask how often would a group this size exceed the threshold that these genes exceed? To be precise: for a specific number \( k \) and a threshold \( \alpha \), how often will random sampling from the same group give at least \( k \) single test p-values will fall under the threshold for significance level \( \alpha \)? In practice we don’t have the luxury of repeating experiments that could give us these estimates. However we can calculate the results of an experiment, where we resample from our existing samples, and calculate how often large groups of genes appear significant.

Empirical Bayes Methods

Empirical Bayes approaches make assumptions about the parameters to be estimated (such as the differences between gene levels in treatment and control groups); used intelligently these assumptions can make use of prior experience with microarray data. A pure Bayes approach assumes specific distributions (prior distributions) for the mean differences of gene levels, and their standard deviations. The empirical Bayes approach assumes less, usually that the form of these distributions is known; the parameters of the prior distribution are estimated from data.

As we get more detailed knowledge of the variability of individual genes, it should become possible to make detailed useful prior estimates based on past experience. A simple empirical Bayes approach to identifying changed genes runs as follows, in two stages:

We believe that all unchanged genes have a range of variances that follow a known prior distribution. We estimate the mean of that distribution as the mean of the experimental variances for all the genes. We estimate the variance of that distribution as the variance of the gene variances. Both of these estimates assume that a small minority of genes are actually changed. Then we consider each gene individually. We would like an estimate of the variability of that gene that is more reliable than its sample standard deviation.
One way to derive that is to work backward from the assumption that the gene variances follow the known distribution. We can compute the probability of any value for sample variance, given a supposed value for the (unknown) true variance of that gene. If we combine that with the prior assignment of probability for each true variance of a gene, we can work out the probabilities that various possible true values underlie the one observed value of the variance.

Not surprisingly it is more probable that a very high sample variance comes about by an over-estimate of a moderately high variance, than a true estimate of a very high variance, simply because there are many more moderate variances than high variances, and over-estimates are not very uncommon compared to precise estimates. In the Bayesian tradition we construct a density function for the probability function for the true value of the gene variance (called a posterior distribution). To come up with a single value we might estimate the expected value of the posterior distribution.

By considerable algebra, this expected value turns out to be a weighted average of the sample variance for the gene, and the mean of the prior distribution, which was set to the mean of all the sample variances. Thus we can obtain a theoretically more reliable estimate of variation, by pooling information about variation, derived from all genes simultaneously. This more reliable estimate of variation should then translate into more reliable t-scores, with more power to pick up moderate differences in gene abundance between samples.

**Several Groups – Analysis of Variance**

Many current microarray studies compare more than two groups. Sometimes the question is to determine differences among three or more cell lines, or strains of experimental animal. Another common design compares the effect of a particular treatment (often a ligand for a receptor), on cell lines (or animals) with wild-type and mutant versions of the receptor. Usually the experimenter wants to know which genes are actively regulated during treatment in both cell lines, or wants some criterion for selecting those that are differentially regulated among groups. These questions belong in the tradition of analysis of variance (ANOVA).

Generally, all of the procedures that were discussed above in the context of two-sample comparisons, carry over to analogues in ANOVA. However the ANOVA analogues are more complex, because many hypotheses are being tested, and some nested within others. Especially in the factorial design case, there is no obvious way to do permutation testing to obtain genome-wide p-values for the interaction (2nd order) effects. Several researchers have suggested permuting the residuals from a fit of the main effects to obtain a permutation distribution.

Analysis of variance is complex and many researchers don't bother; they do t-tests on the contrasts that interest them. This procedure is less effective than analysis of variance for two reasons. The power of a t-test to pick up a difference increases with greater confidence in the denominator — the estimate of variability. ANOVA computes a consensus estimate of variability within groups, based on all the groups. This estimate, based on more information, has more confidence (greater degrees of freedom) than the variability estimated between two samples alone. Thus for the same degree of difference, and the same variability estimate, the ANOVA will pick up more differences than the t-test. It's possible to use the consensus estimate of variability in the t-score denominators for all tests.

Sometimes when researchers compare treatment effects on WT and mutant, they look for genes that are significant on one list and not on the other. This procedure is very fallible; if genes are changed by treatment to the same extent in both samples, according to statistical theory, the significance levels should fluctuate considerably. Many should appear on one list but not on the other. The ANOVA for a factorial design is the most efficient way of identifying the true changes in regulation under treatment among the many noisy genes.

**References**