Normalizing Expression Arrays

Key ideas

Normalization is the attempt to compensate for systematic technical differences between chips, to see more clearly the systematic biological differences between samples. Biologists have long experience coping with technical variation between experimental conditions that is unrelated to the biological differences they seek. However expression arrays may vary in even more ways than measures such as rt-PCR. In practice methods that have worked well for rt-PCR and similar measures do not perform as well for microarray data,which shows many dimensions (here 'many dimensions' means simultaneous inferences)of systematic differences. No lab technician can reproduce exactly the same technical conditions in each assay of a long series. Researchers do not have detailed data on the procedures and therefore cannot compare all the relevant technical condition of individual hybridizations.

The key assumption that enables us to normalize microarrays is that only a few genes are expressed at really different levels between samples (here 'a few’ means up to a few hundred out of tens of thousands). Hence the expression levels of the majority of genes are essentially identical across samples. This assumption may not always be held, for example when comparing highly transformed cancer cells with normal cells. However,we need to assume something of this sort if we are to make comparisons at all, since the fluorescent intensity measures can easily be manipulated by twiddling settings on the photo-multiplier tube and these measures are routinely affected by different amounts one sample than of another and different efficiencies of label incorporation. A more specific assumption is that microarray measures should not be correlated with technical characteristics of probes, such as: base content,dye bias, intensity effects,print-tip effects, T_m,position in gene, etc. We would hope that biological changes would be independent of technical characteristics of expression probes.

Normalization of Expression Microarrays

Why Normalize?

Heraclitus said "You cannot step into the same river twice": no technician does things in exactly the same way twice. In many cases differences in processing of two samples, particularly during the processes of making cDNA, labelling and hybridization systematically affect the signals on the arrays, on which those samples' gene expressions are measured. Some systematic processing differences between chips that frequently affect measures are:

- Different amounts of RNA are used;
- One dye is more readily incorporated than the other (in 2-color systems);
- Different amounts of labelling may occur (in one-color systems)
- The hybridization reaction may proceed more fully to equilibrium in one array than the other;
- Hybridization conditions may vary across the physical extent of one array;
- Scanner settings are often different;

and of course, Murphy's Law predicts even more variation than any expert may expect.

In order to identify the real biological differences among samples, we attempt to compensate 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 expression arrays differ from those useful for other kinds of array-based assays. Traditionally normalization has been done differently for one-color (e.g. Affymetrix and Illumina) arrays, compared to two-color arrays (e.g. most Agilent and Nimblegen arrays). With two-color arrays there is usually a 'within-array' normalization between the colors, before an 'across-array' normalization. This guide will introduce across array methods first before discussing 'within-array' methods.

Housekeeping and Standard Genes

One early approach was to select one gene, or a small subset of genes, which were not expected to be differentially expressed across any samples; these were usually chosen to be ‘housekeeping ’genes: genes which are required for basic cell processes, and which were once believed to have very stable expression levels in all cells. This commonsense approach is used routinely in rt-PCR. However, these housekeeping genes seem to vary by 30% or more across healthy samples, and even more in cancer samples (Lee et al, 2002). A 30% error in normalization is not a big deal for rt-PCR, where the accuracy is at best a half-cycle, and one cycle corresponds to an increase of a factor of two. However, in a microarray study a 30% difference over the whole genome can give many false positives.

Rank-invariant normalization method

If there are no really stable genes across all samples, can we identify a set of genes stable across a particular set of samples? This question leads to an approach called 'invariant set normalization': the idea is to find empirically a set of genes which seem like the best candidates because their expression values maintain the same relation to each other across all samples. If such a group could be found, then the rest of the genes could be 'pinned' to their levels of expression. The implementation in dChip (Li & Wong, 2001) selects a reference chip, and then uses pairwise comparison with that reference chip, because such an invariant subset cannot be identified across the full range of signal values, but only the most abundant genes. This may not be a fault of the idea, but rather due to the wide range of signals obtained on Affymetrix chips for genes which are absent (up to signal intensities of 2000), in turn a a reflection of the high levels of cross-hybridization of 25-mers. It may be that this idea can be useful for the other technologies, based on longer oligomers, on which non-expressed genes have uniformly low signals.

Statistical Approaches

Most approaches to normalizing expression levels assume that the overall distribution of RNA levels doesn't change much between samples or 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 that feature, you are adding some error to those samples where the feature you have observed does not correspond well with the true underlying source of bias. Statisticians try to balance bias and noise; their rule of thumb is that it is better to slightly under-correct for systemic biases than to compensate fully.

A key decision researchers must make, with consequences for normalization, is on what scale to analyse their data. It is common practice to transform to a logarithmic (usually base 2) scale. The principal motivation for this transformation is to make variation roughly comparable among measures which span several orders of magnitude. This often works as intended however such a transformation may actually increase variation of the low intensity probes relative to the rest. In particular when a measure can be reported as zero, the logarithm isn't defined. A simple remedy is to add a small constant to the measures before taking the logarithm. A more sophisticated approach is to use a non-linear variance stabilizing transform; a simple such transform is $f(x) = \ln((x + (x^2 + c^2)^{1/2} / 2)$, where c is the ratio of the constant portion of the variance to the rate of increase of variance with intensity.

Scaling by overall brightness

The simplest approach posits 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 hybridized on the array. The sizes of the RNA molecules are comparable, so the number of RNA molecules should also be roughly the same in each sample. Consequently, approximately the same number of labeled molecules from each sample should hybridize 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_{red} and C_{green} , by: where f_i^{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:

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

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 subtracted 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 outliers at the top end, for example 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 because they are less influent by the outliers. To do the latter, you 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 that the variance of the log-ratios differs between arrays. One approach to dealing with this problem is to scale the $\log_2$(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 $\log_2$(ratio) values have a systematic dependence on intensity – most commonly a deviation from zero for low-intensity spots. Under–expressed genes appear up-regulated in the red channel. Moderately expressed genes appear up regulated in the green channel. No known biological process would regulate genes that way – so this must be an artefact. It appears that the explanation is chemical: the two dyes do not 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 $\log_2(\mathrm{R_i}/\mathrm{G_i})$ for each element on the array as a function of the $\log_2(R_iG_i)$ product intensities. This 'R–I' (for ratio–intensity) plot can reveal intensity-specific artifacts in the $\log_2$(ratio) measurements. Note that Terry Speed's group calls these variables 'M' and 'A', (for ‘minus’ and ‘add’ – on the log scale) and they call the plot an ‘M-A plot'.

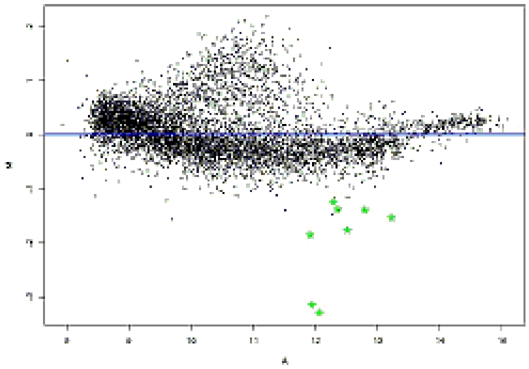

Figure 1. Ratio–Intensity plot showing characteristic 'banana' shape of cDNA ratios; log scale on both axes. (courtesy Terry Speed)

We would like a normalization method that can remove such intensity-dependent effects in the $\log_2$(ratio) values. The functional form of this dependence is unknown, and must depend on many variables we don't measure. An ad-hoc statistical approach widely used in such situations, is to fit some smooth curve through the points. One example of such a smooth curve is a locally weighted linear regression (lowess) curve. Terry Speed's group at Berkeley used this approach. To calculate a lowess curve fit to a group of points $(x_1, y_1), \ldots (x_N, y_N)$, 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 $x_i = \log2(R_i G_i)$ and $y_i = \log_2(R_i/G_i)$. 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 $\log_2(r_i^*) = \log2(R_i/G_i) - y^*(\log_2(R_i G_i))$ The result is that the mean ratio for probes with any mean intensity is 0, as seen below.

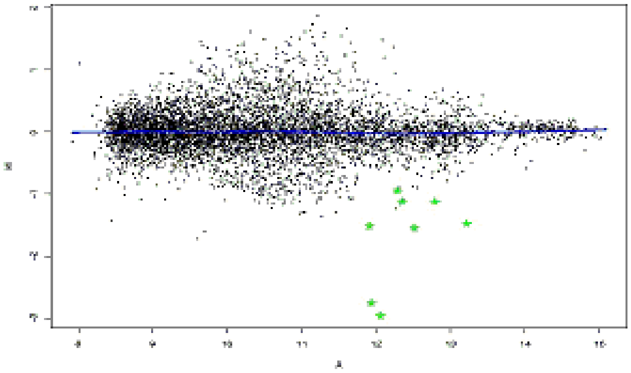

Figure 2. As in Figure 1, but corrected by lowess normalization.

Local normalization

Most normalization algorithms, including lowess, can be applied either globally (to the entire data set) or locally (to specific subsets 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', 'print-tip 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; the approach assumes that genes in any sub grid should have average expression ratios of 1, and that several hundred probes are in each group. Another approach is to look for a smooth correction to uneven hybridization. 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

By 2003 statisticians were developing more complex normalizations. Some statisticians noticed that there were pronounced differences in the loess curves fit to log-ratios in different regions of the same chip; they tried to fit separate loess curves to each set of probes produced by a common print tip of a robotically printed cDNA array. Others tried to fit two-dimensional loess surfaces over chips. Further complications included estimating a clone order effect, and re-scaling variation within each print-tip group. In 2003, Benjamin Bolstad, one of Terry Speed’s students, proposed cutting through all the complexity by a simple non-parametric normalization procedure, at least for one-color arrays. He proposed to shoe-horn the intensities of all probes on each chip into one standard distribution shape, which he determined by pooling all the individual chip distributions. In practice, the distribution of intensities from any high-quality chip will do. The algorithm mapped every value on any one chip to the corresponding quantile of the standard distribution; hence the method is called quantile normalization. This simple 'between-chip' procedure worked as well as most of the more complex procedures then current, and certainly better than the regression method, which was then the manufacturer's default for Affymetrix chips. This method was also made available as the default in the affy package of Bioconductor, which has become the most widely used suite of freeware tools for microarrays (see www.bioconductor.org). For all these reasons quantile normalization has become the normalization procedure which I see most often in papers.

In a formula, the transform is

$$x_{norm} = F_i^{-1}(F_{ref}(x)) ,$$

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

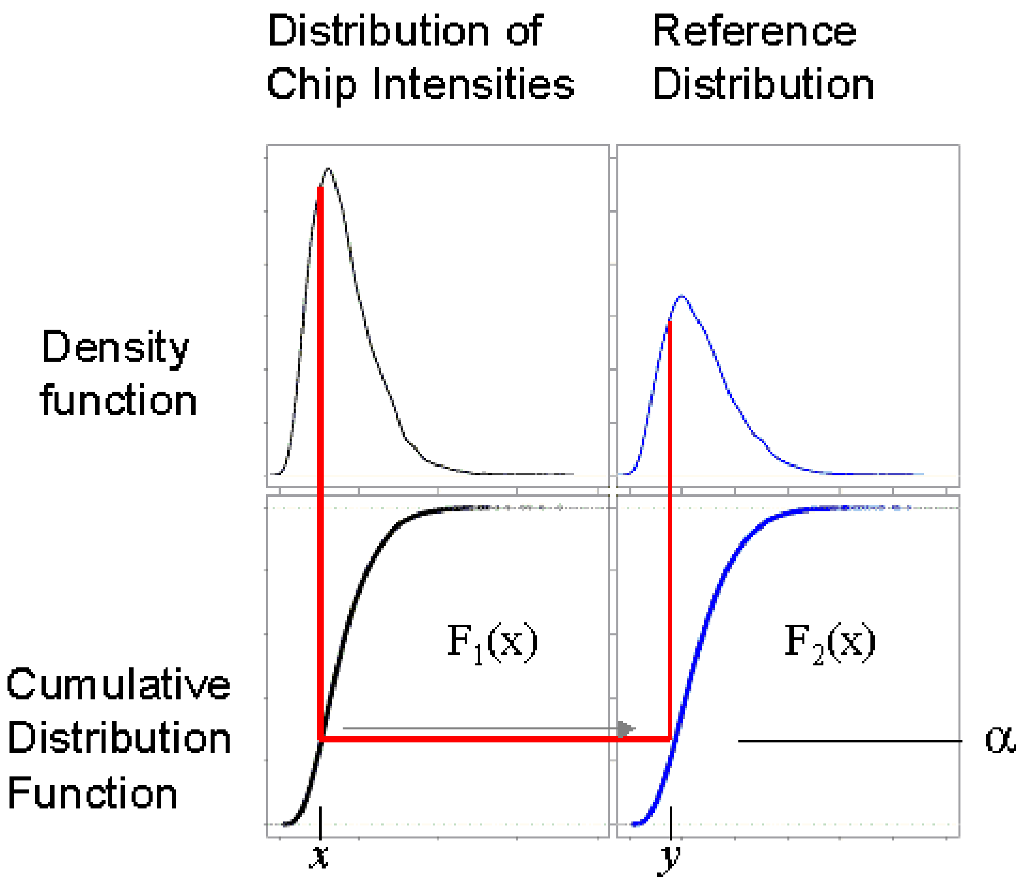

Figure 5. Schematic representation of quantile normalization: the value *x*, which is the α-th quantile of all probes on chip 1, is mapped to the value y, which is the α quantile of the reference distribution F_2.

If F_i and F_{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 6. However the transform is strong enough to cope with the non-linear ratio-intensity relationships revealed in figure 7A; see figure 7B after quantile transformation.

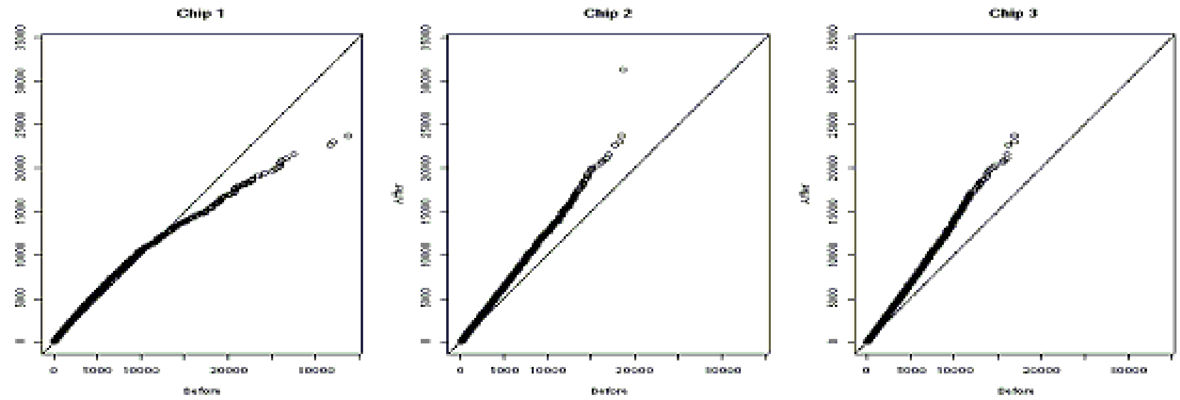

Figure 6. The effects of quantile normalization on raw probe values in three chips. Raw values are on x-axis, normalized values on y-axis. Often this transform looks very much like a scaling transform (nearly linear), but sometimes it is quite non-linear.

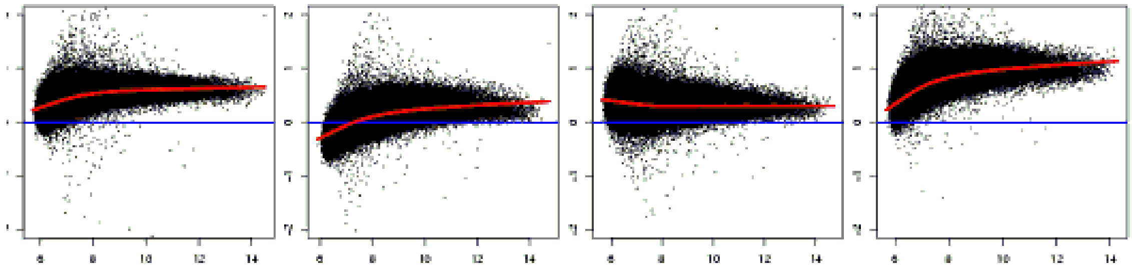

Figure 7A. Ratio Intensity Plot of all probes for four pairs of chips from GeneLogic spike-in experiment

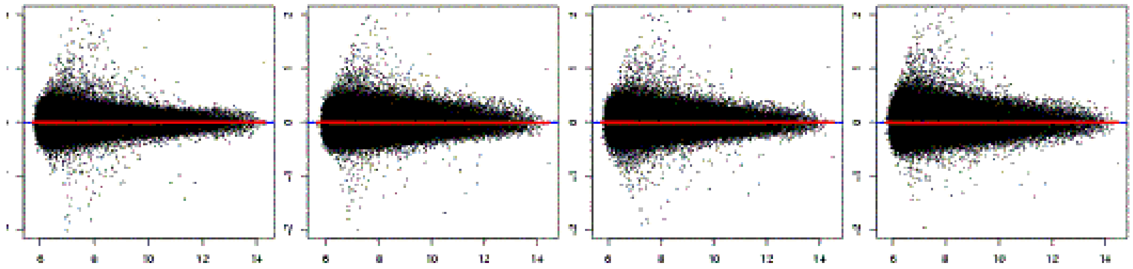

Figure 7B. As in A, after normalization by matching quantiles. Both figures courtesy of Terry Speed

This form of normalization also reduces noise among replicate measures of the same samples, compared to normalization by scaling, as shown below in figure 8.

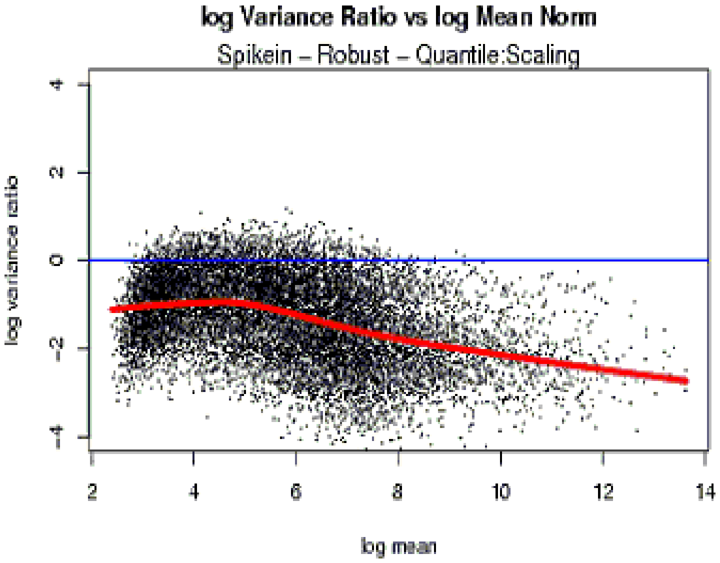

Figure 8. Each dot represents one probe on an Affymetrix U95A chip. On the y-axis is the ratio of variance across a set of replicates after quantile normalization, divided by the variance of the scale-normalized values. On the x-axis are the mean levels. Both axes on log scale.

While quantile normalization is a simple fast one-size-fits-all solution, it engenders some problems of its own. For example the genes in the upper range of intensity are forced into the same distribution shape; such shoe-horning reduces biological differences as well as technical differences. A recent adjustment to the quantile procedure in the latest versions of the affy package fixes that problem. A second issue is more subtle. For reasons that are still not entirely clear, the errors in different sets of probes are highly correlated. For probes for genes that are in fact not expressed in the samples under study, these correlated errors comprise most of the variation among chips. When quantile normalization acts on these probes, the procedure preserves this apparent but entirely spurious correlation among low-intensity probes and sometimes seems to amplify that correlation. Hence sophisticated data mining methods that depend on subtle analysis of correlations may pick up spurious relationships. Finally quantile normalization explicitly depends on the idea that the distribution of gene expression measures does not change across the samples. This assumption is unlikely to be true when testing treatments with severe effects on the transcription apparatus or studying cancer samples with severe genomic aberrations.

Normalization by regression on technical variables

Several recent approaches to microarray normalization have attempted to estimate the biases on individual arrays as non-parametric functions of a moderate number

(5-10) of technical variables describing the probes on the array [gcRMA, Carvalho, Liu]. However these attempts have not employed anything like a consistent methodology.

Normalization by inferring covariates

Researchers have observed that changes in the sample preparation environment, such as a different technician, or a new batch of arrays, or a new hybe station, can make a significant difference in the measures. However these kinds of data are often not available to the data analyst, and surely there are other factors, not tracked, which could make a difference. What if the analyst could infer these covariates from the data itself? This is the basis of two proposals, with very different algorithms.

Mike West proposed selecting control genes, which should not change among samples, and then doing a multivariate analysis of these controls, to identify covariates that influence many gene expression measurues.

(Leek & Storey, 2007) proposed using multivariate structural analysis of residuals to infer some sample covariates with significant effect on many gene expression measures. They first fit the design model and then perform a singular value decomposition (SVD) of the residual matrix.

Normalization by projection away from inferred technical variation

A related proposal by (Reimers, 2010) also works from an SVD of residuals. However (Reimers, 2010) recovers the subspace corresponding to characteristic technical variation.