Quality Control and Early Diagnostics for cDNA Microarrays

by Günther Sawitzki

We present a case study of a simple application of R to analyze microarray data. As is often the case, most of the tools are nearly available in R, with only minor adjustments needed. Some more sophisticated steps are needed in the finer issues, but these details can only be mentioned in passing.

Quality control for microarrays

Microarrays are a recent technology in biological and biochemical research. The basic idea is: you want to assess the abundance of some component of a *sample*. You prepare a more or less well defined *probe* which binds chemically to this specific component. You bring probe and sample into contact, measure the amount of bound component and infer the amount in the original sample.

So far, this is a classical technique in science. But technology has advanced and allows high throughput. Instead of applying samples to probes one by one, we can spot many probes on a chip (a microscope slide or some other carrier) and apply the sample to all of them simultaneously and under identical conditions. Depending on the mechanics and adhesion conditions, today we can spot 10,000–50,000 probes on a chip in well-separated positions. So one sample (or a mixture of some samples) can give information about the response for many probes. These are not i.i.d. observations, but the probes are selected and placed on the chip by our choice, and response as well as errors usually are correlated. The data have to be considered as high dimensional response vectors.

Various approaches how to analyze this kind of data are discussed in the literature. We will concentrate on base level data analysis. Of course any serious analysis does specify diagnostic checks for validating its underlying assumptions. Or, to put it the other way, an approach without diagnostics for its underlying assumptions may be hardly considered serious. The relevant assumptions to be checked depend on the statistical methods applied, and we face the usual problem that any residual diagnostic is influenced by the choice of methods which is used to define fit and hence residuals. However there are some elementary assumptions to check which are relevant to all methods. If you prefer, you can name it quality control.

We use R for quality control of microarrays. We simplify and specialize the problem to keep the presentation short. So for now the aim is to implement an early diagnostic to support quality control for microarrays in a routine laboratory environment - something like a scatterplot matrix, but adapted to the problem at hand.

Some background on DNA hybridization

Proteins are chemical components of interest in biochemical research. But these are very delicate to handle, and specific probes have to be developed individually protein by protein. As a substitute, the genes controlling the protein production can be studied. The genes are encoded in DNA, and in the synthesis process these are transcribed to RNA which then enters to the protein production. DNA and transcribed RNA come in complementary pairs, and the complementary pairs bind specifically (*hybridization*). So for each specific DNA segment the complementary chain gives a convenient probe.

In old Lackmus paper, the probe on the paper provides colour as visible indicator. In hybridization experiments, we have to add an indicator. In DNA/RNA hybridization experiments the DNA is used as a probe. The indicator now is associated with the sample: the RNA sample is transcribed to cDNA (which is more stable) and marked, using dyes or a radioactive marker. For simplicity, we will concentrate on colour markers. After probe and sample have been hybridized in contact and unbound material has been washed off, the amount of observable marker will reflect the amount of bound sample.

Data is collected by scanning. The raw data are intensities by scan pixel, either for an isolated scan channel (frequency band) or for a series of channels. Conventionally these data are pre-processed by image analysis software. Spots are identified by segmentation, and an intensity level is reported for each spot. Additionally, estimates for the local background intensities are reported.

In our specific case, our partner is the Molecular Genome Analysis Department of the German cancer research center (DKFZ). Cancer research has various aspects. We simplify strongly in this presentation and consider the restricted question as to which genes are significantly more (or significantly less) active in cancer cell, in comparison to non-cancerous "normal" cells. Which genes are relevant in cancer? Thinking in terms of classical experiments leads to taking normal and tumor cell samples from each individual and applying a paired comparison. Detecting *differentially expressed* genes is a very different

 ISSN 1609-3631

challenge from classical comparison. Gene expression analysis is a search problem in a high dimensional structured data set, while classical paired comparison deals with independent pairs of samples in a univariate context. But in the core gene expression analysis benefits from scores which may be motivated by a classical paired comparison.

Statistical routine immediately asks for factors and variance components. Of course there will be a major variation between persons. But there will be also a major contribution from the chip and hybridizations: the amount of sample provided, the sample preparations, hybridization conditions (e.g., temperature) and the peculiarities of the scanning process which is needed to determine the marker intensity by spot, among others are prone to vary from chip to chip. To control these factors or variance components, sometimes a paired design may be used by applying tumor and normal tissue samples on the same chip, using different markers (e.g., green and red dye). This is the basic design in the fundamental experiments.

The challenge is to find the genes that are relevant. The typical experiment has a large number of probes with a high variation in response between samples from different individuals. But only a small number of genes is expected to be differentially expressed ("*many genes few differentials*").

To get an idea of how to evaluate the data, we can think of the structure we would use in a linear gaussian model to check for differences in the foreground (fg). We would implement the background (bg) as a nuisance covariable and in principle use a statistics based upon something like

$$\Delta = (Y_{tumor} - Y_{normal})$$

where e.g.,

$$\begin{aligned} Y_{tumor} &= \ln(Y_{fg\ tumor} - Y_{bg\ tumor}) \\ Y_{normal} &= \ln(Y_{fg\ normal} - Y_{bg\ normal}) \end{aligned}$$

given or taken some transformation and scaling. Using log intensities is a crude first approach. Finding the appropriate transformations and scaling as a topic of its own, see e.g. Huber et al. (2002). The next non-trivial step is combining spot by spot information to gene information, taking into account background and unknown transformations. Details give an even more complex picture: background correction and transformation may be partly integrated in image processing, or may use information from other spots. But there is an idea of the general structure, and together with statistical folklore it suggests how to do an early diagnostics.

We hope that up to choice of scale Δ covers the essential information of the experiment. We hope we can ignore all other possible covariates and for each spot we can concentrate on $Y_{fg\ tumor}$, $Y_{bg\ tumor}$, $Y_{fg\ normal}$, $Y_{bg\ normal}$ as source of information. Up to choice of scale, $(Y_{fg\ tumor} - Y_{fg\ normal})$ represents the raw "effect" and a scatter plot of $Y_{fg\ tumor}$ vs $Y_{fg\ normal}$ is the obvious raw graphic tool.

Unfortunately this tool is stale. Since tumor and normal sample may have different markers (or may be placed on different slides in single dye experiments) there may be unknown transformations involved going from our target, the amount of bound probe specific sample, to the reported intensities Y. Since it is the most direct raw plot of the effect, the scatterplot is worth being inspected. But more considerations may be necessary.

Diagnostics may be sharper if we know what we are looking for. We want diagnostics that highlight the effect, and we want diagnostics that warn against covariates or violation of critical assumptions. For now, we pick out one example: spatial variation. As with all data collected from a physical carrier, position on the carrier may be an important covariate. We do hope that this is not the case, so that we can omit it from our model. But before running into artifacts, we should check for spatial homogeneity.

Diagnostic for spatial effects

For illustration, we use a paired comparison between tumor and normal samples. After segmentation and image pre-processing we have some intensity information $Y_{fg\ tumor}$, $Y_{bg\ tumor}$, $Y_{fg\ normal}$, $Y_{bg\ normal}$ per spot. A cheap first step is to visualize these for each component by position (row and column) on the chip. With R, the immediate idea is to organize each of these vectors as a matrix with dimensions corresponding to the physical chip layout and to use `image()`.

As usual, some fine tuning is needed. As has been discussed many times, in S and R different concepts of coordinate orientation are used for matrices and plots and hence the image appears rotated. Second, we want an aspect ratio corresponding to the geometric while image follows the layout of R's graphics regions. A wrapper `imagem()` around `image()` is used as a convenient solution for these details (and offers some additional general services which may be helpful when representing a matrix graphically).

Using red and green color palettes for a paired comparison experiment gives a graphical representation which separates the four information channels and is directly comparable to the raw scanned image. An example of these images is in `http://www.statlab.uni-hd.de/projects/genex/`.

The next step is to enhance accessibility of the information. The measured marker intensity is only an indirect indicator of the gene activity. The relation is influenced by many factors, such as sample amount, variation between chips, scanner settings. Internal scanner calibration in the scan machine (which may automatically adjust between scans) and settings of

the image processing software are notorious sources of (possibly nonlinear) distortions. So we want to make a paired comparison, but each of both sides undergoes an unknown transformation. Here specific details of the experiment come to help. We cannot assume that both samples in a pair undergo the same transformation. But while we do not know the details of the transformations, we hope that at least each is guaranteed to be monotonous.

At least for experiments with "many genes - few differentials", this gives one of the rare situations where we see a bless of high dimensions, not a curse. The many spots scanned under comparable conditions provide ranks for the measured intensity which are a stable indicator for the rank of the original activity of each. So we apply `imagem()` to the ranks, where the ranks are taken separately for the four components and within each scan run.

The rest is psychology. Green and red are commonly used as dyes in these experiments or in presentations, but these are not the best choice for visualization. If it comes to judging quantitative differences, both colour scales are full of pitfalls. Instead we use a colour palette going from blue to yellow, with more lightness in the middle value range.

To add some sugar, background is compared between the channels representing tumor and normal. If we want a paired comparison, background may be ignorable if it is of the same order for both because it balances in differences. But if we have spots for which the background values differ drastically, background correction may be critical. If these points come in spatial clusters, a lot more work needs to be done in the analysis. To draw attention to this, spots with extremely high values compared to their counterpart are highlighted. This highlighting is implemented as an overlay. Since `image()` has the facility to leave undefined values as background, it is enough to apply `imagem()` again with the uncritical points marked as `NA`, using a fixed colour.

Wrapping it up

Rank transformation and image generation are wrapped up in a single procedure `showchip()`. Since the ranks cover all order information, but lose the original scale, marginal scatterplots are provided as well, on a fixed common logarithmic scale.

Misadjustment in the scanning is a known notorious problem. Estimated densities over all spots within one scan run are provided for the four information items, together with the gamma correction exponent which would be needed to align the medians.

If all conditions were fixed or only one data set were used, this would be sufficient. The target environment however is the laboratory front end where the chip scanning is done as the chips come in. Experimental setup (including chip layout) and sampling protocols are prone to vary. Passing the details as single parameters is error prone, and passing the data repeatedly is forbidding for efficiency reasons due to the size of the data per case.

The relevant information is bundled instead. Borrowing ideas from relational data bases, for each series of experiments one master list is kept which keeps references to tables or lists which describe details of the experiment. These details go from description of the geometry, over a representation of the actual experimental design to various lists which describe the association between spots and corresponding probes. S always had facilities for a limited form of object oriented programming, since functions are first class members in S. Besides data slots, the master list has list elements which are functions. Using enclosure techniques as described in Gentleman and Ihaka (2000), it can extract and cache information from the details. The proper data are kept in separate tables, and methods of the master list can invoke `showchip()` and other procedures with a minimum of parameter passing, while guaranteeing consistency which would endangered if global variables were used.

From the user perspective, a typical session may introduce a new line of experiments. The structure of the session is

```
curex <- NewGenex("<new project name>")
## create a new descriptor object from
## default.  if an additional parameter is given,
## it is a model to be cloned.

curex$nsetspotdesc("<some descriptor name>")
## we access other lists by name.
## This is risky, but for the present purpose
## it does the job ...
## Possibly many more experimental details

curex$save()
## The current experiment descriptor saves
## itself as <new project name>.RData
```

Once an experimental layout has been specified, it is attached to some specific data set as

```
curex$nsetdata("<some data name>")
## associates the data with the
## current experimental layout
```

Of course it would be preferable to set up a reference from the data to the experiment descriptor. But we have to take into account that the data may be used in other evaluation software; so we should not add elements to the low level data if we can avoid it.

When the association has been set up, application is done as

```
curex$showchip(<some chip identification>)
## names or indices of chip/chips to show.
```

This is a poor man's version of object oriented programming in R. For a full grown model of object oriented programming in R, see Chambers and Lang (2001).

Looking at the output

We cannot expect a spatially uniform distribution of the signals over the spots. The probes do not fall at random on the chip, but they are placed, and the placement may reflect some strategy or some tradition. In the sample output (Figure 1), we see a gradient from top to bottom. If we look closer, there may be a separation between upper and lower part. In fact these mirror the source of the genes spotted here. This particular chip is designed for investigations on kidney cancer. About half of the genes come from a "clone library" of kidney related genes, the other from genes generally expected to be cancer related. This feature is apparent in all chips from this special data series. The immediate warning to note is: it might be appropriate to use a model which takes into account this as a factor. On some lines, the spots did not carry an active probe. These unused spots provide an excellent possibility to study the null distribution. The vertical band structure corresponds to the plates in which the probes are provides; the square blocks come from characteristics of the print head. These design factors are well known, and need to taken into account in a formal analysis.

Of course these known features can be taken into account in an adapted version of showchip, and all this information is accessible in the lists mentioned above. In a true application this would be used to specialize `showchip()`. For now let us have another look at the unadapted plot shown in Figure 1.

Background is more diffuse than foreground—good, since we expect some smooth spatial variation of the background while we expect a grainy structure in the foreground reflecting the spot probes. High background spots have a left-right antisymmetry. This is a possible problem in this sample pair if it were unnoticed (it is is an isolated problem on this one chip specimen).

As in the foreground, there is some top/bottom gradient. This had good reason for the foreground signal. But for the background, there is no special asymmetry. As this feature runs through all data of this series of experiments, it seems to indicate a systematic effect (possibly a definition of "background" in the image processing software which picks up too much foreground signal).

Conclusion

`showchip()` is used for the first steps in data processing. It works on minimal assumptions and is used to give some first pilot information. For any evaluation strategy, it can be adapted easily if the critical statistics can be identified on spot level. For a fixed evaluation strategy, it then can be fine tuned to include information on the spatial distribution of the fit and residual contributions of each spot. This is where the real application lies, once proper models are fixed, or while discussing model alternatives for array data analysis.

An outline of how `showchip()` is integrated in a general rank based analysis strategy is given in Sawitzki (2001).

P.S. And of course there is one spot out.

Acknowledgements

This work has been done in cooperation with Molecular Genome Analysis Group of the German cancer research center (DKFZ) http://www.dkfz.de/abt0840/.

Thanks to Wolfgang Huber and Holger Sültmann for background information and fruitful discussions.

A video showing the experiments step by step is accessible as http://www.uni-hd.de/media/mathematik/microarrays.rm and available as DVD on CD from the DKFZ address.

Bibliography

J. M. Chambers and D. T. Lang. Objectoriented programming in ᵹ *R News*, 3(1):17–19, September 2001. 9

R. Gentleman and R. Ihaka. Lexical scope and statistical computing. *Journal of Computational and Graphical Statistics*, 3(9):491–508, September 2000. 8

W. Huber, A. von Heydebreck, H. Sültmann, A. Poustka, and M. Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. (Submitted to Bioinformatics), January 2002. 7

G. Sawitzki. Microarrays: Two short reviews and some perspectives. Technical report, StatLab Heidelberg, 2001. URL http://www.statlab.uni-hd.de/projects/genex/. (Göttingen Lectures) Göttingen/Heidelberg 2001. 9

Günther Sawitzki
StatLab Heidelberg
gs@statlab.uni-heidelberg.de

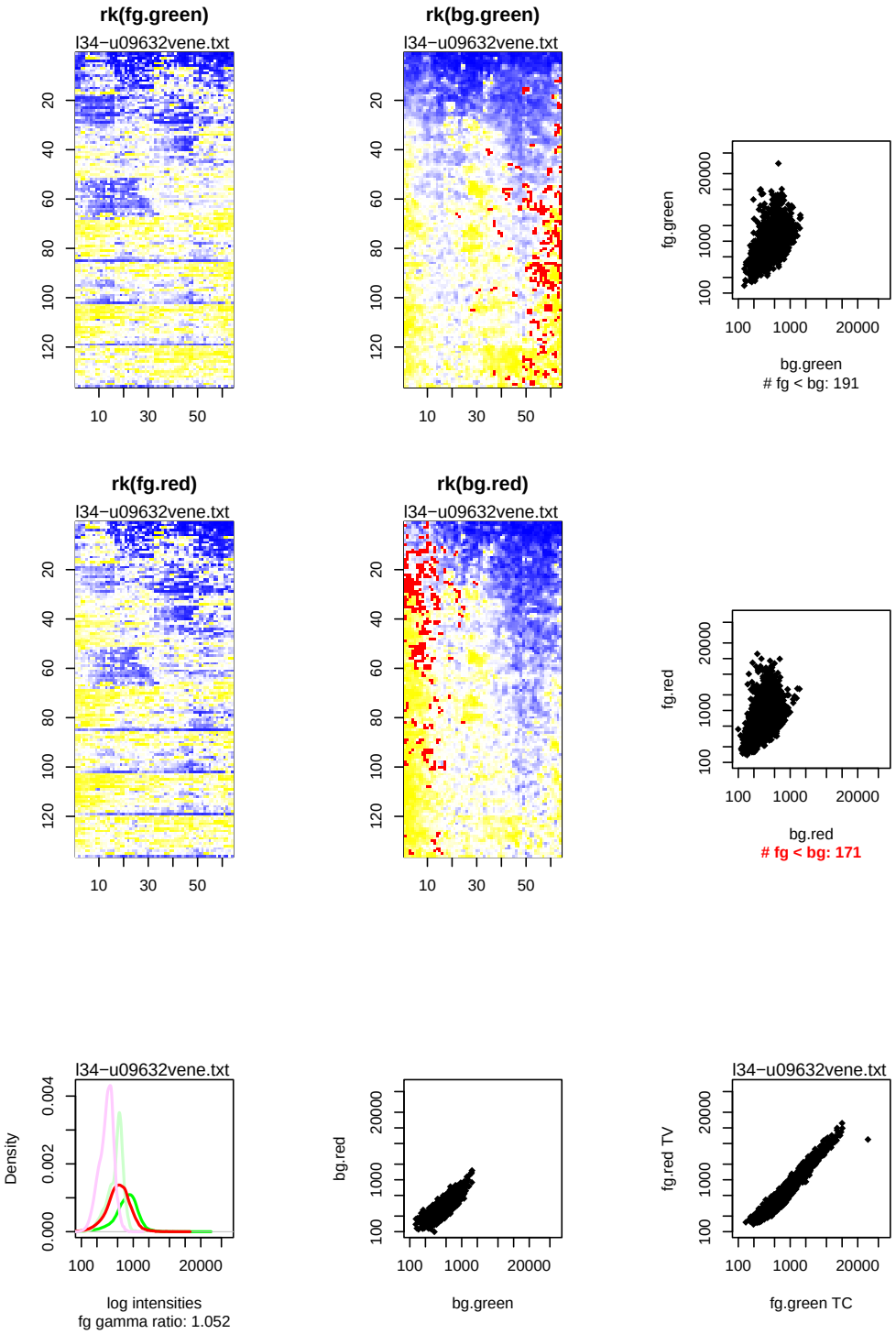

Figure 1: Output of `qcshowhip()`. This example compares tumor center material with tumor progression front.