GS01 0163
Analysis of Microarray Data

Keith Baggerly and Kevin Coombes
Section of Bioinformatics
Department of Biostatistics and Applied Mathematics
UT M. D. Anderson Cancer Center
kabagg@mdanderson.org
kcoombes@mdanderson.org

2 September 2004

Lecture 2: Structure of Microarrays

- Obtaining Extra R Packages
- Graphics in R
- The Structure of Glass Microarrays
- The Structure of Affymetrix Microarrays

Obtaining extra R packages

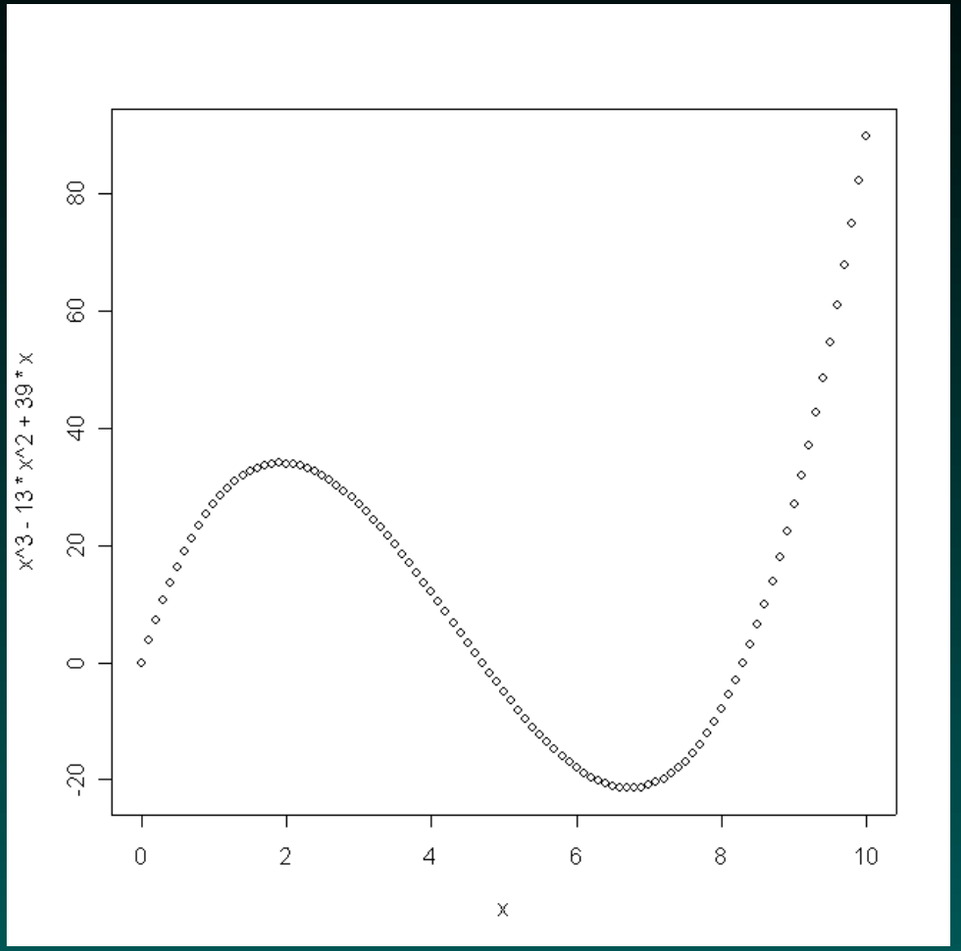
The R GUI makes it easy to get additional packages via the internet. From the “Packages” menu, you simply select either “Install package(s) from CRAN...” or “Install package(s) from Bioconductor”. Both menu items present you with a dialog box containing a list of the available packages. You then select one or more (by holding the control key while clicking with the mouse) and press the “OK” button. R then downloads the package, installs it, and updates the help files. It finishes by asking if you want to delete the downloaded files; unless you want to save them to install them on another computer without an internet connection, the usual answer is “yes”. We’ll come back to this point later when we start working with Bioconductor.

Graphics in R

R includes a fairly extensive suite of graphics tools. There are typically three steps to producing useful graphics.

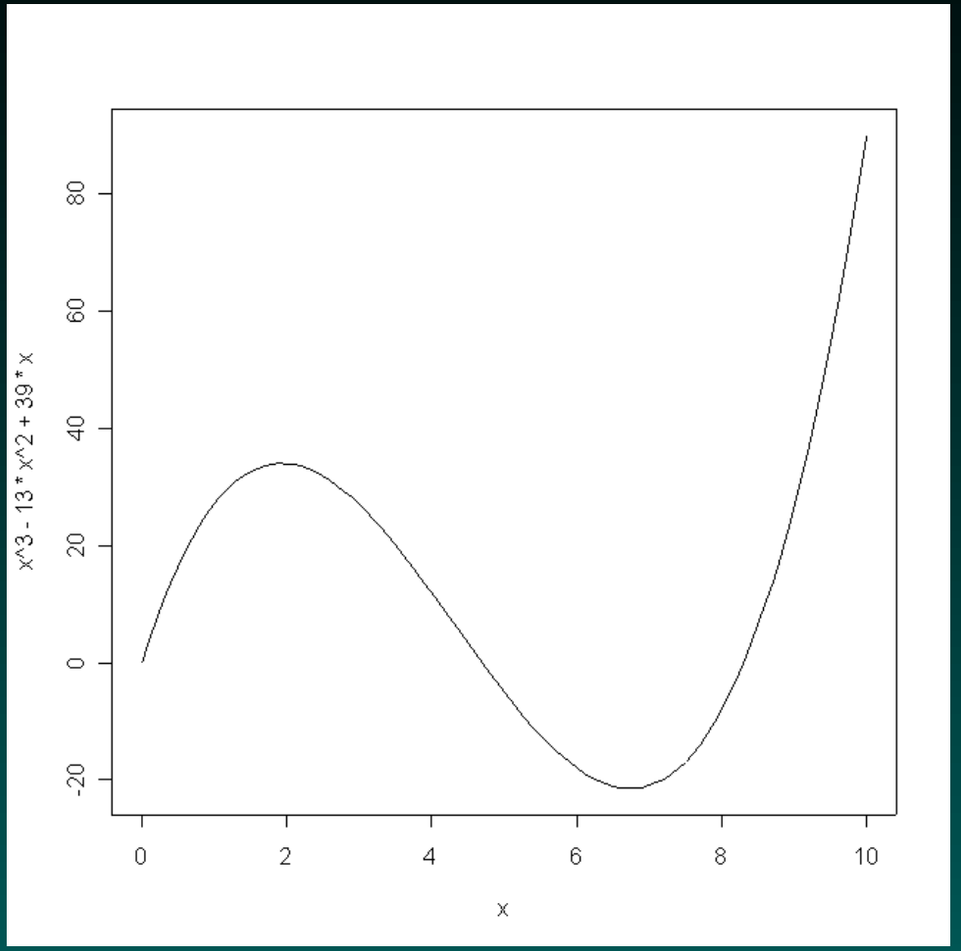
- Creating the basic plot
- Enhancing the plot with labels, legends, colors, etc.
- Exporting the plot from R for use elsewhere

Basic plot



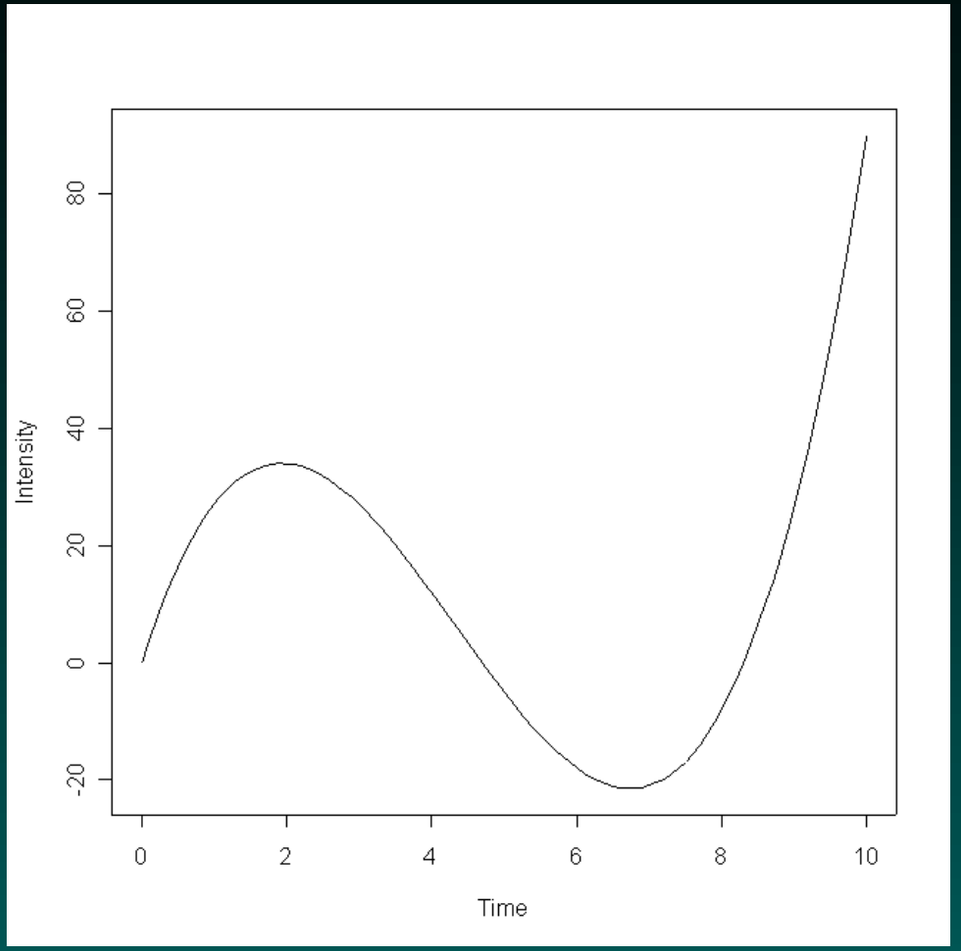

```
> x <- (0:100)/10  # from 0 to 10, increment of 0.1
> plot(x, x^3-13*x^2+39*x)
```

Plotting curves instead of points




```
> plot(x, x^3-13*x^2+39*x, type='l')
```

Labeling axes




```
> plot(x, x^3-13*x^2+39*x, type='l',
+      xlab='Time', ylab='Intensity')
```

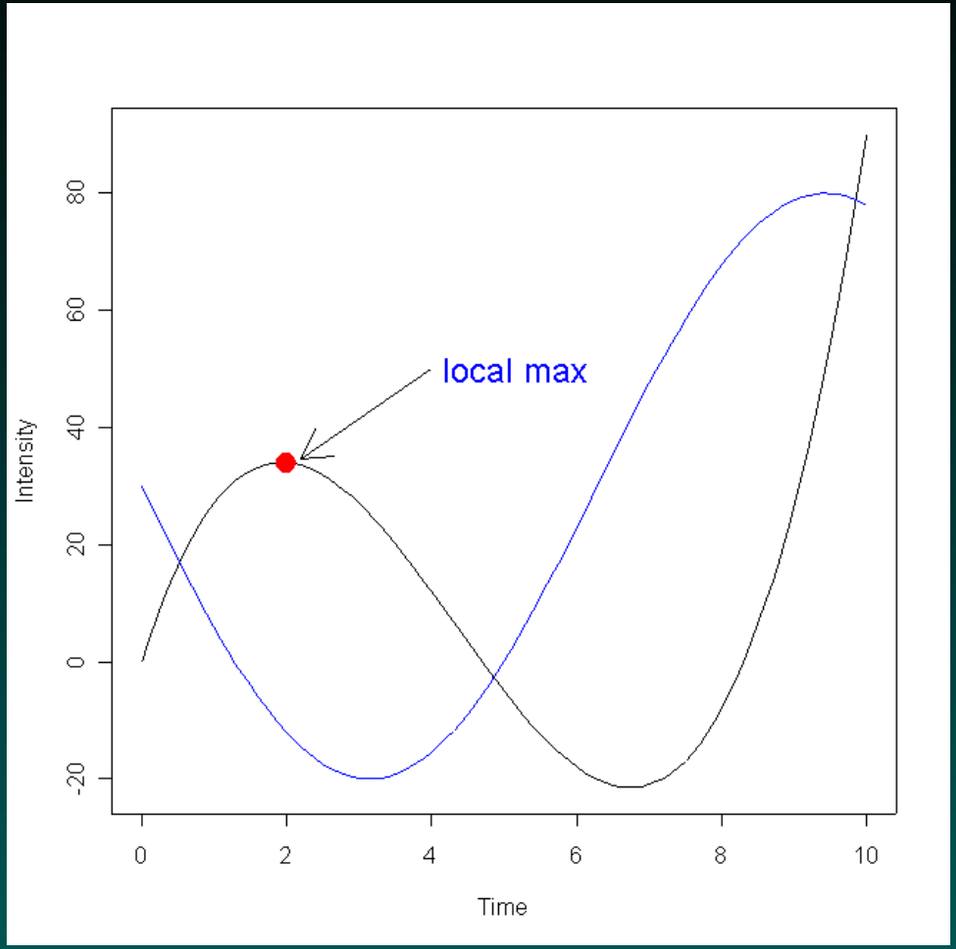
Repeating yourself ...

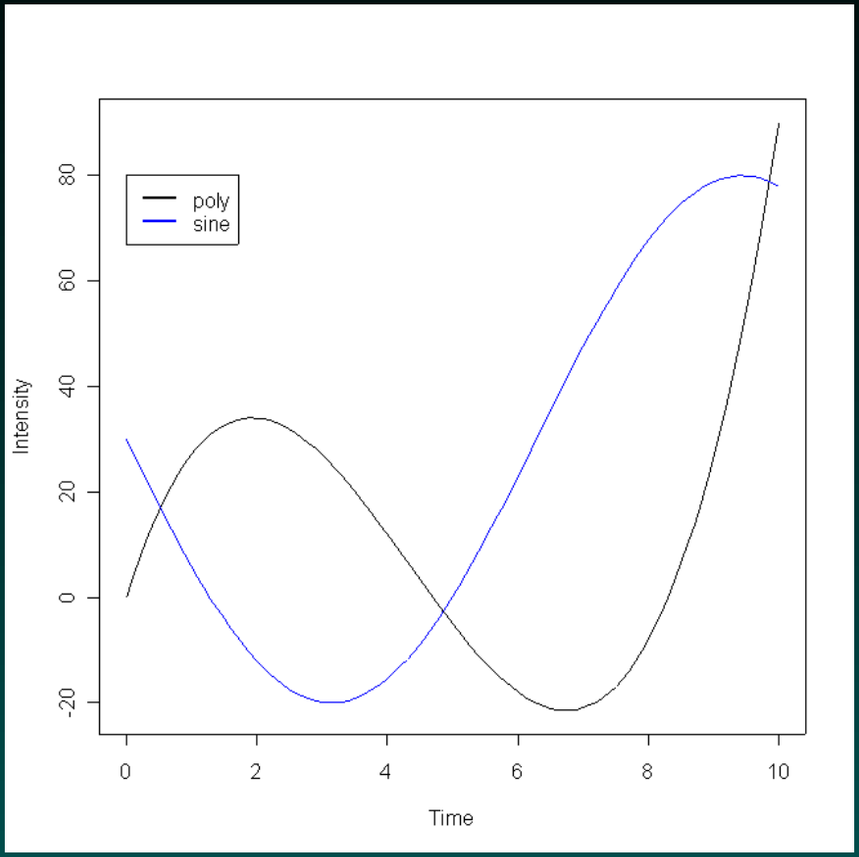
If you change your mind about how you want things like curves or axes displayed, you often have to regenerate the plot from scratch. There are very few things that can be changed after the fact.

You can, however, add points, arrows, text, and lines to existing plots.

```
> points(2, 34, col='red', pch=16, cex=2)
> arrows(4, 50, 2.2, 34.5)
> text(4.15, 50, 'local max', adj=0,
+     col='blue', cex=1.5)
> lines(x, 30-50*sin(x/2), col='blue')
```

Annotated plot






```
> plot(x, x^3-13*x^2+39*x, type='l')
> lines(x, 30-50*sin(x/2), col='blue')
> legend(0, 80, c('poly', 'sine'),
+     col=c('black', 'blue'), lwd=2)
```

Saving plots to use elsewhere

In the R GUI, first activate the window containing a plot that you want to save. On the “File” menu, choose “Save As ->”, which gives you several choices of file format. The two most useful formats are probably

- PNG; useful for including figures in PowerPoint or Word
- Postscript; often useful for submitting manuscripts.

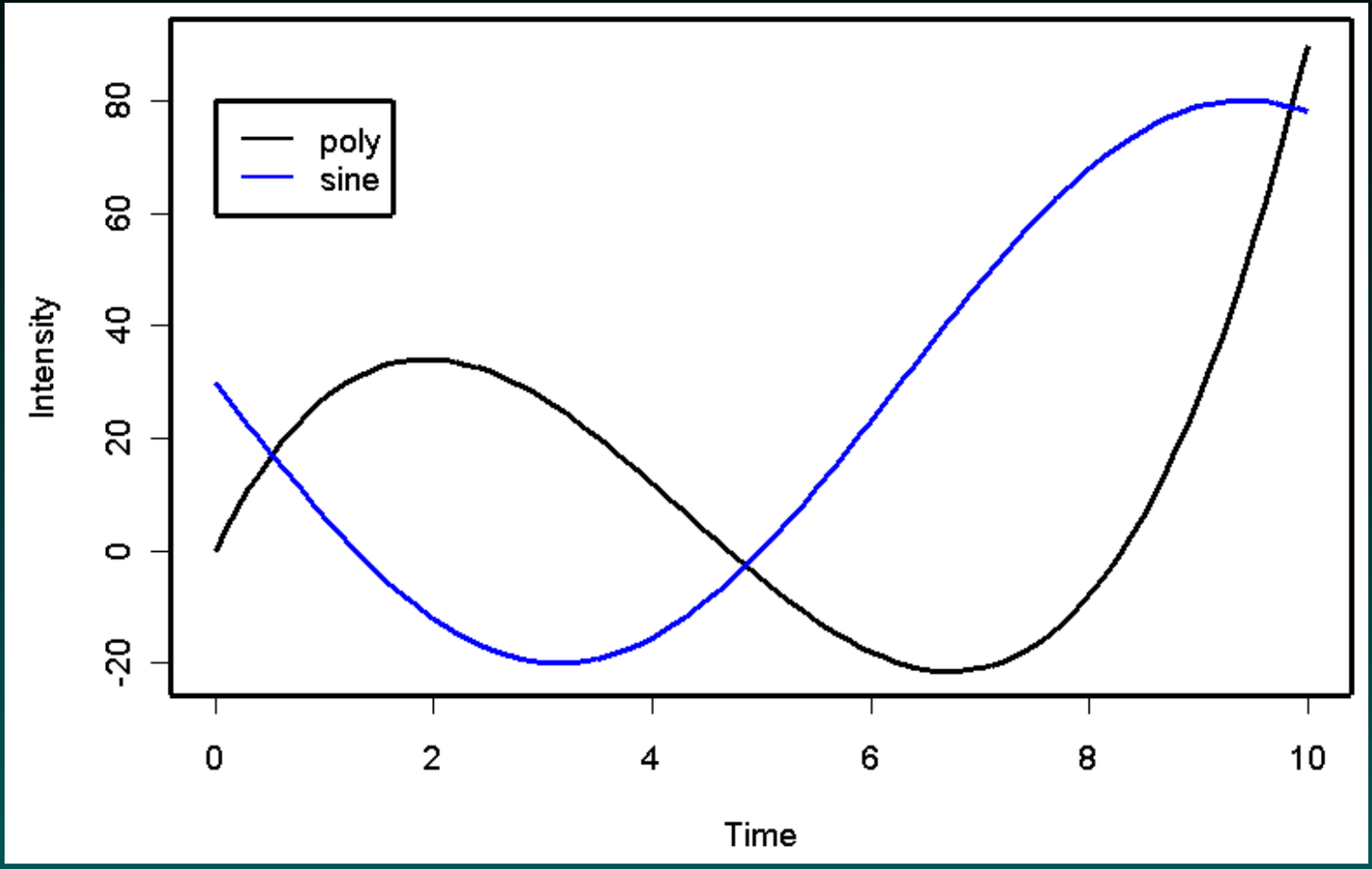
Graphics parameters

R includes a large number of additional parameters that can be used to control the layout of a graphics window. For a complete list, read the help pages on `par` and `windows`. The figures included here so far have been produced using the default settings. Remaining figures will be produced after running the commands

```
> windows(width=8, height=5, pointsize=14)
> par(mai=c(1, 1, 0.1, 0.1), lwd=3)
```

which will change the default window size, the size of characters used in the window, and the margin areas around the plot. Rerunning the last set of plot commands will then produce the following figure:

Same figure with new defaults

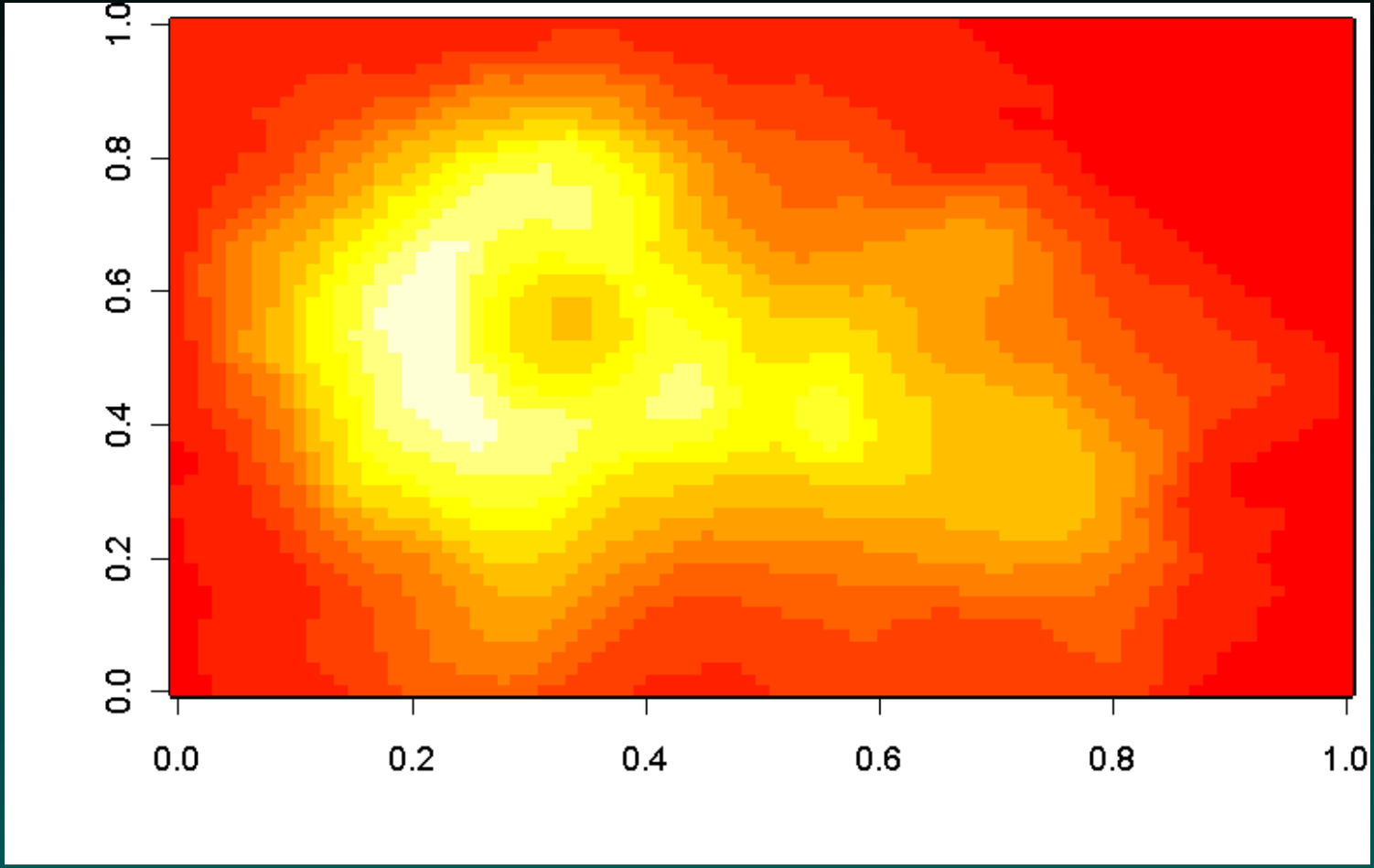


Additional graphics commands

R includes commands to generate a large number of different kinds of plots, including histograms (`hist`), box-and-whisker plots (`boxplot`), bar charts (`barplot`) ,dot plots (`dotplot`), strip charts (`stripchart`), and pie charts (`pie`). Dalgaard's book gives an overview of many of these graphics commands in Chapter 3.

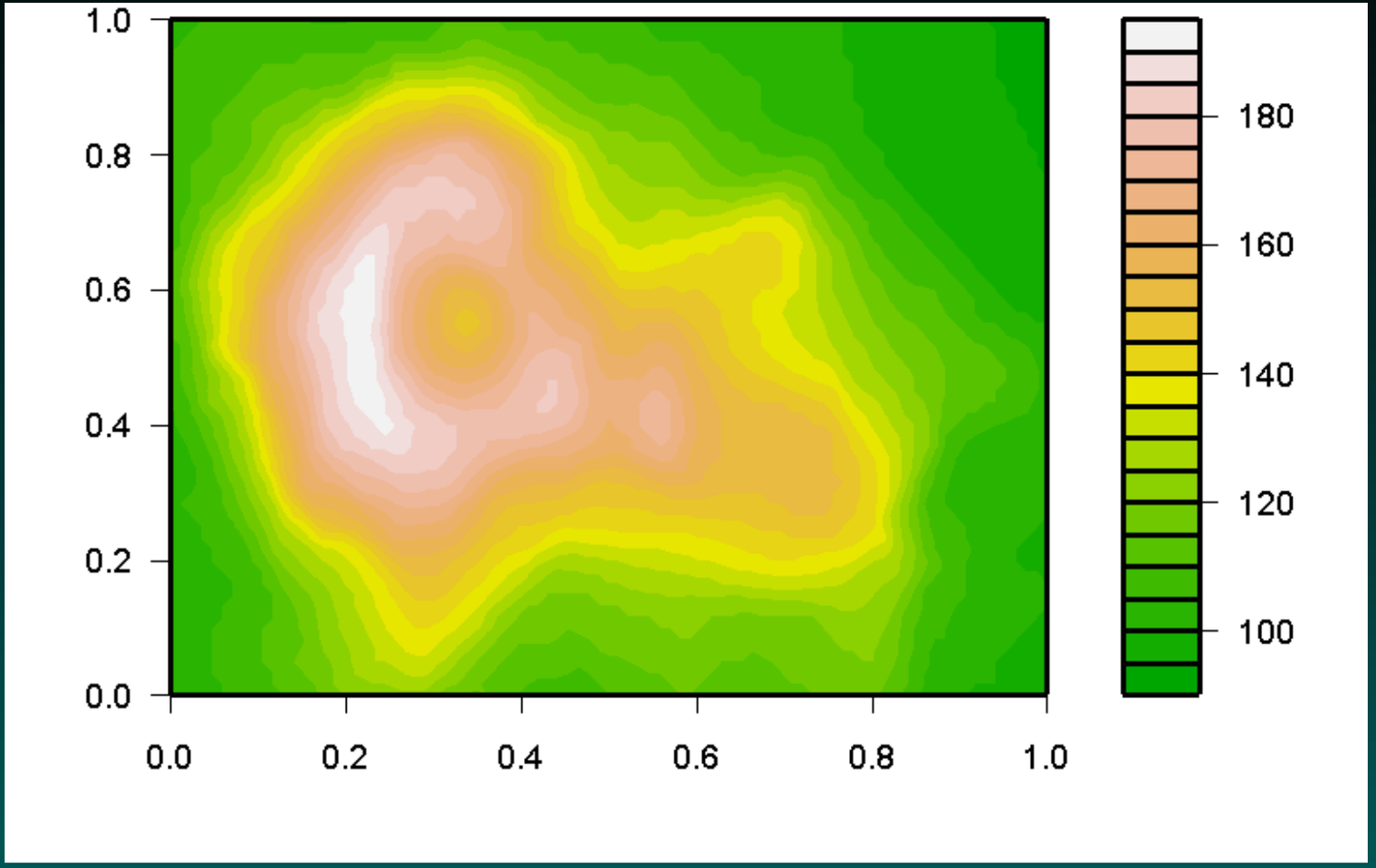
R also includes a number of commands to visualize matrices. On the next slide, we use the `data` command to load a sample data matrix that comes with R. We then produce an image of the matrix, treating the rows and columns as x-y coordinates and the matrix entries as intensities or heights.

Volcano




```
> data(volcano)
> image(volcano)
```

Volcano




```
> filled.contour(volcano, color=terrain.colors)
```

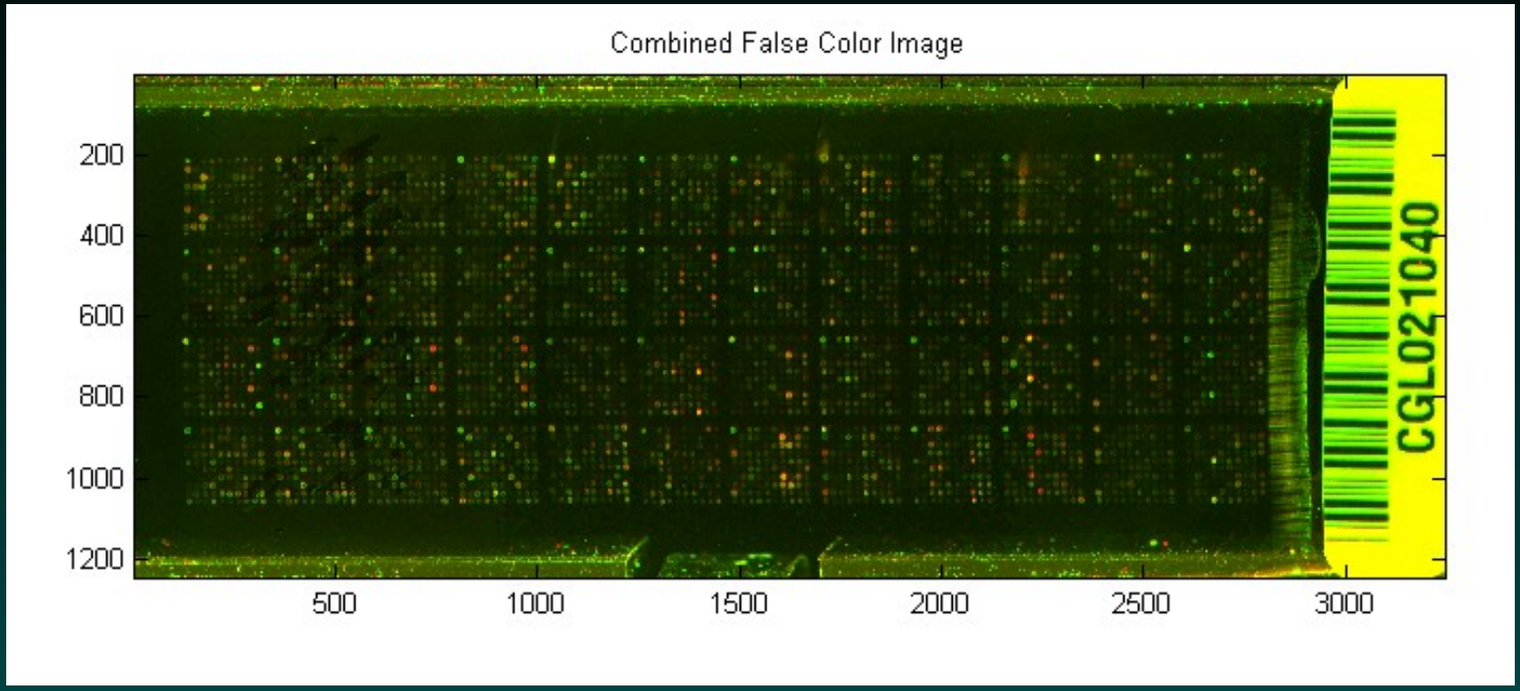
The Structure of Glass Microarrays

Recall: Glass microarrays are composed of spots of cDNA or long oligos, arranged in a regular geometric pattern. Each spot contains cDNA with a known sequence, pre-designed as a probe for a specific target gene. A typical array contains thousands (up to 50K) of spots. Two samples are fluorescently labeled and cohybridized to the array. After hybridization, the array is scanned and two images are produced, containing the raw data.

Arrayer robot

Glass microarrays are produced by robotically spotting cDNA or long oligos (60- or 70-mers) on glass microscope slides.

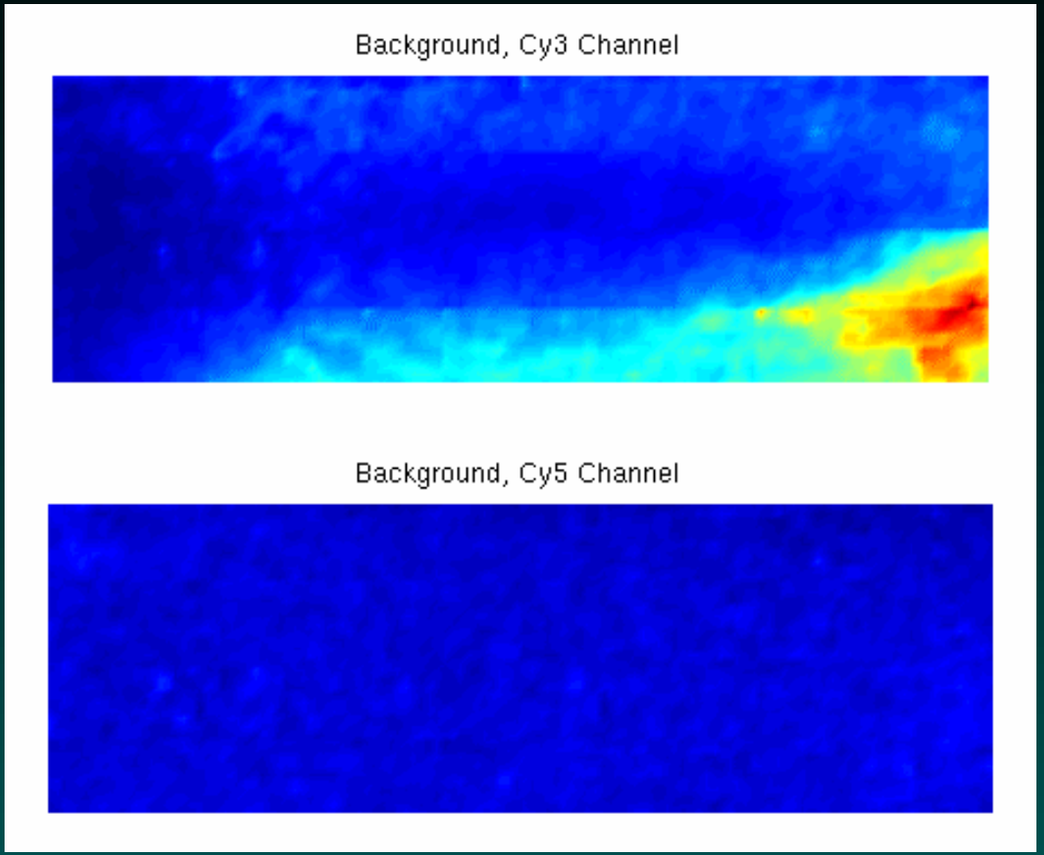
Robotic pins impose a grid substructure



This slide was produced by a robot using 48 pins, laid out in a 4×12 rectangle. Each 10×10 subgrid was produced by a separate pin.

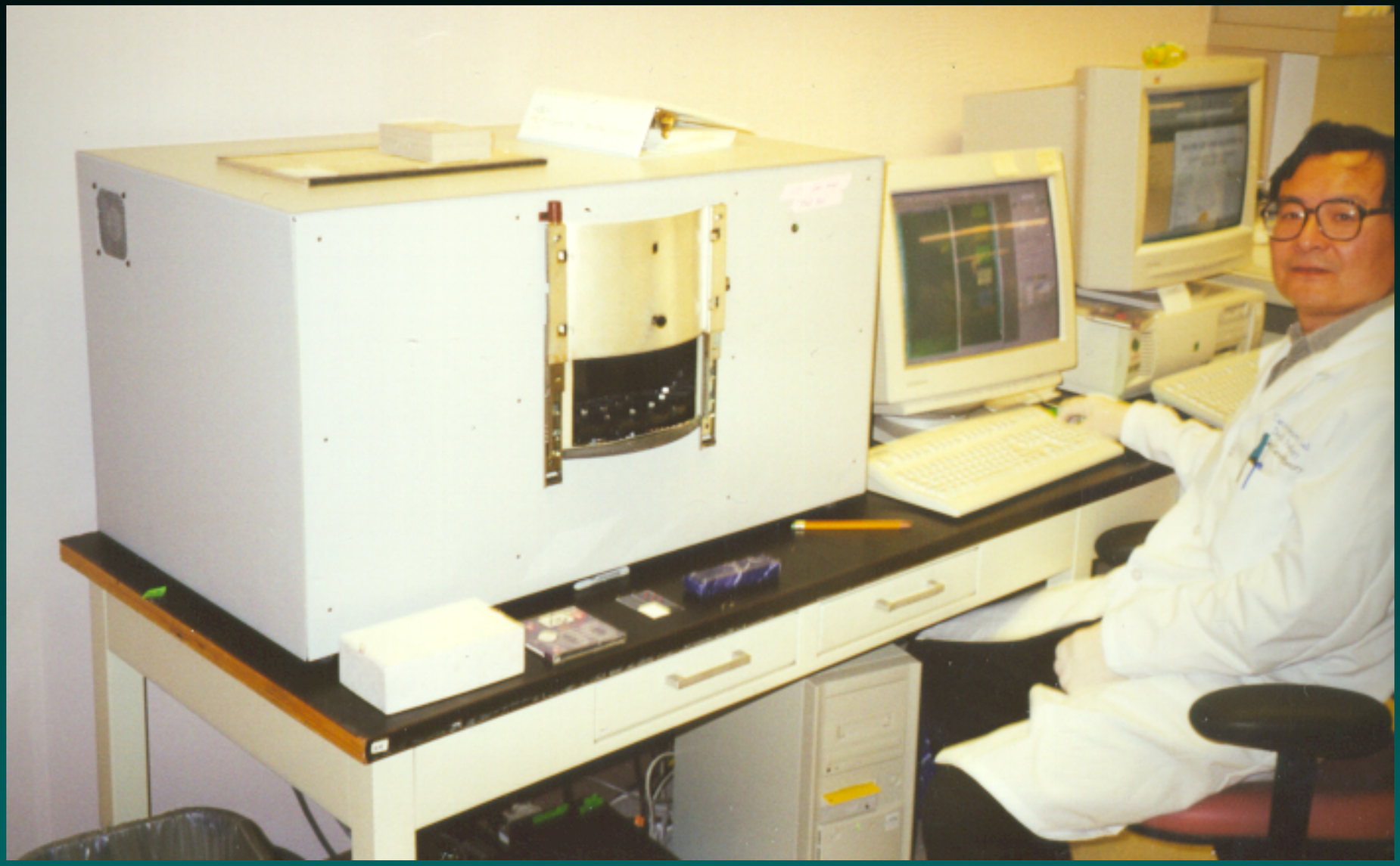
Note: Flaws in a pin (e.g., blunt tip) can cause one subgrid to be systematically different from the others.

Trends in the background

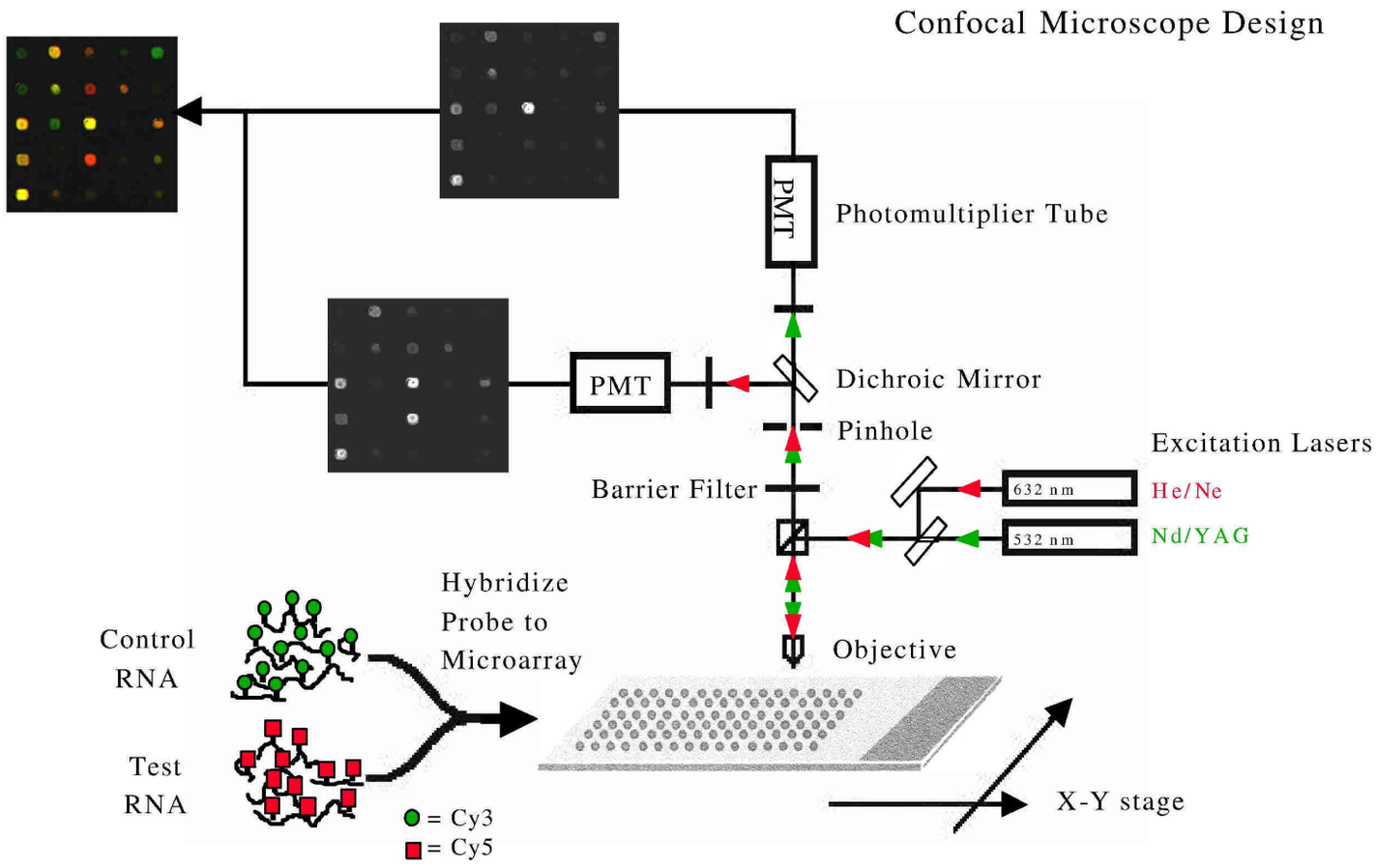


Systematic differences in grids can create spatially coherent artifacts. These can also be caused by incomplete mixing of the hybridization solution, often revealing itself in the background.

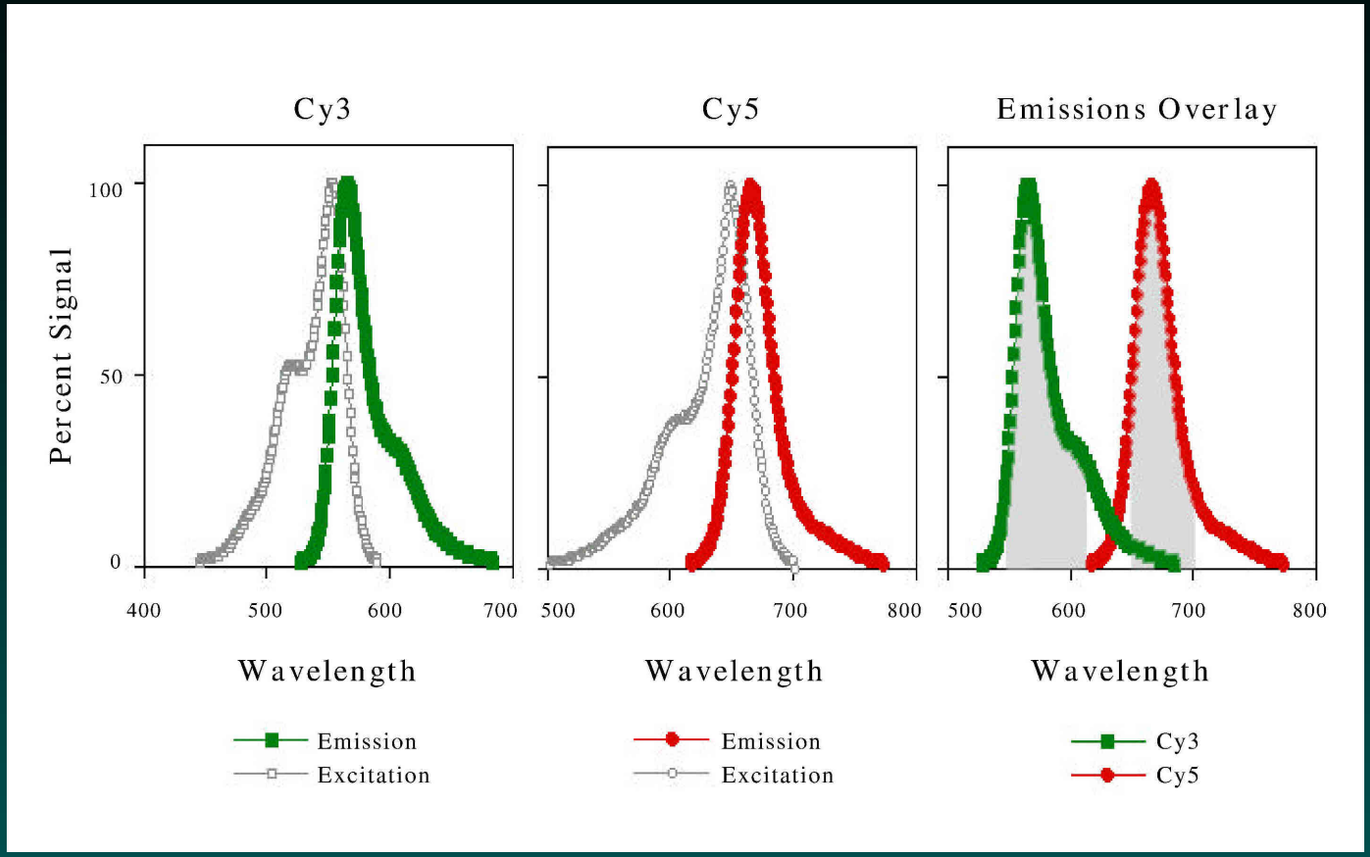
Scanners are boring beige boxes...



Scanning schematic



Fluorescent Emission Properties



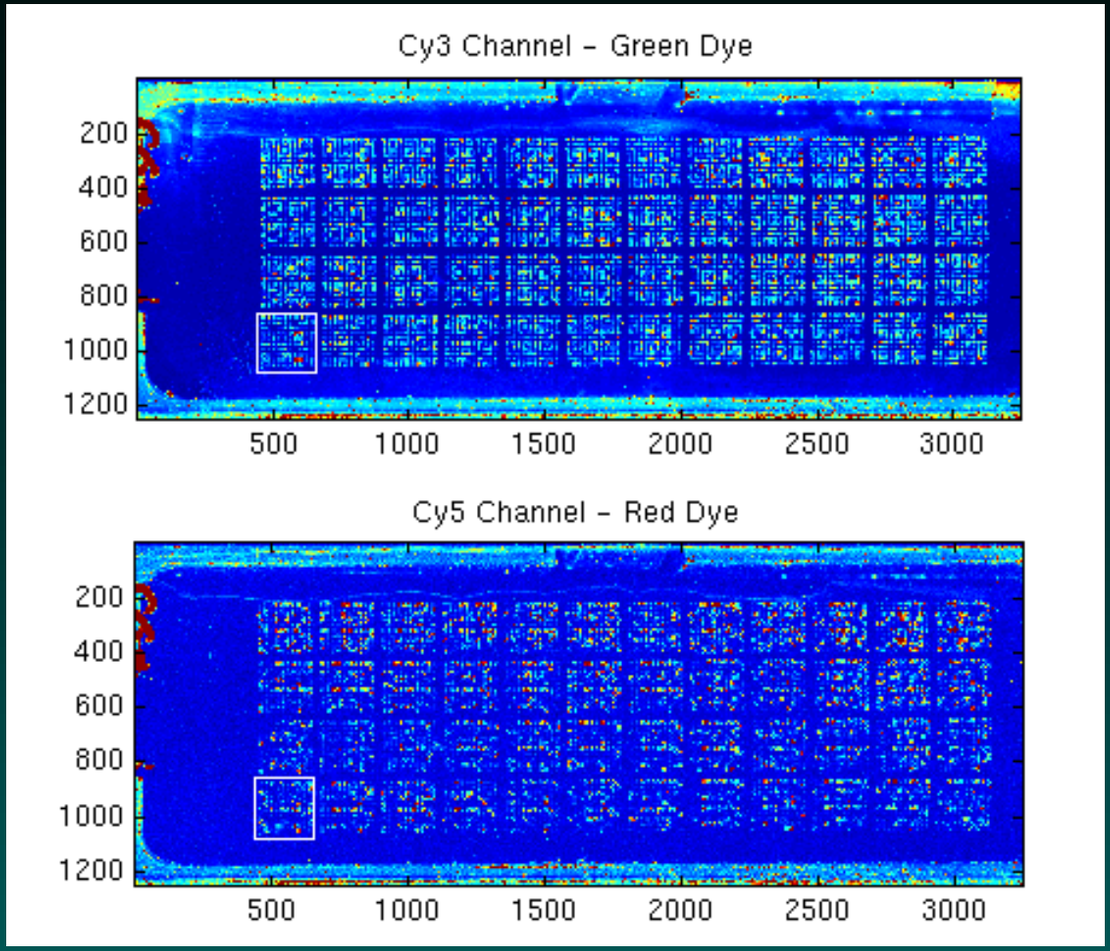
Fluorescent dyes (Cy3 = green, wavelength 532; Cy5 = red, wavelength 635) absorb light at a specific excitation wavelength, and emit light at a specific larger wavelength.

Dye effects

The basic difficulty in making sense of glass microarray data is deciding how to combine the information from the two channels. The two dyes have different chemical properties; they may be incorporated into genes at different rates. They may also fluoresce at different rates.

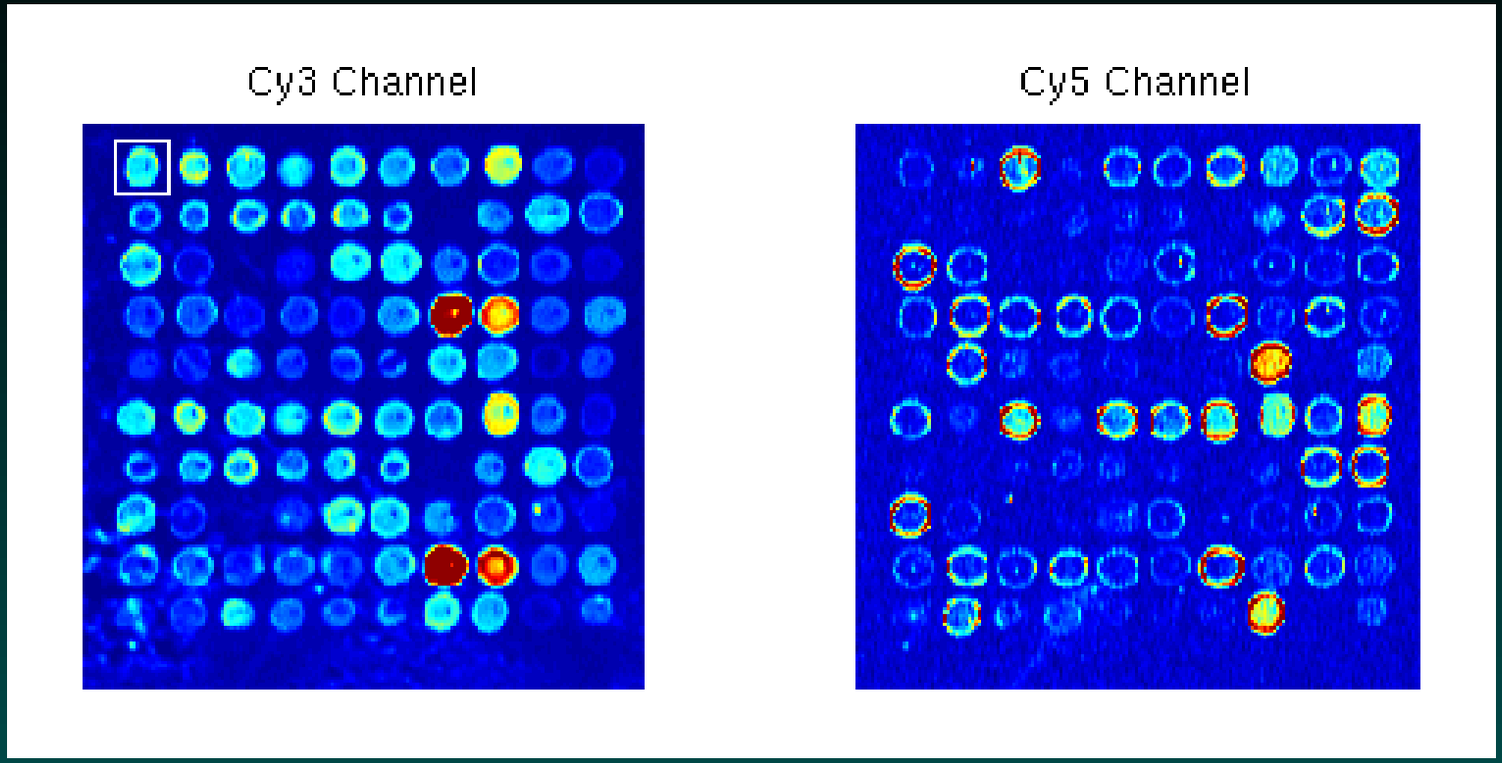
These differences can lead to array-wide differences in intensity and to gene-specific differences related to the DNA sequence of the target genes. These differences have implications for the downstream analysis, and will therefore affect how we think about designing microarray experiments.

A closer look at a microarray image



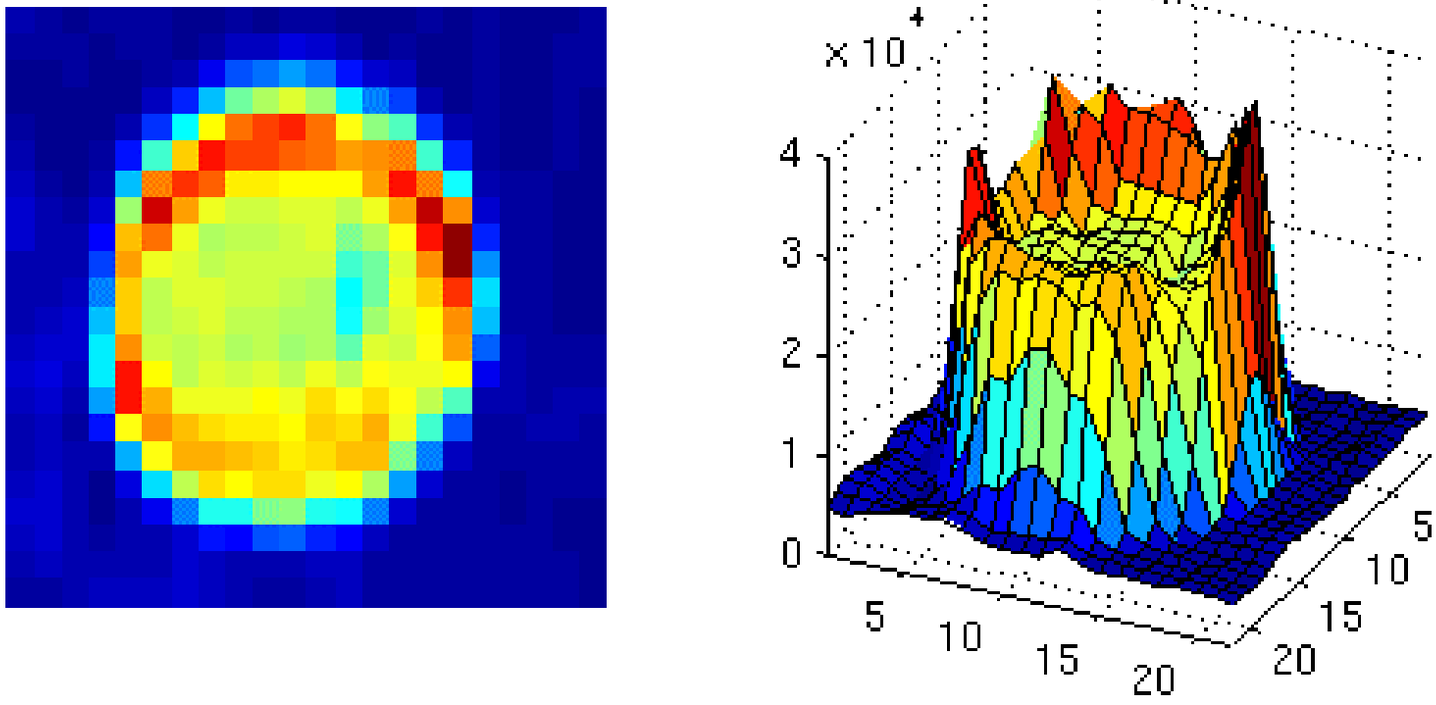
This is a fairly old microarray from M.D. Anderson; it doesn't yet have a barcode attached.

A closer look at one subgrid



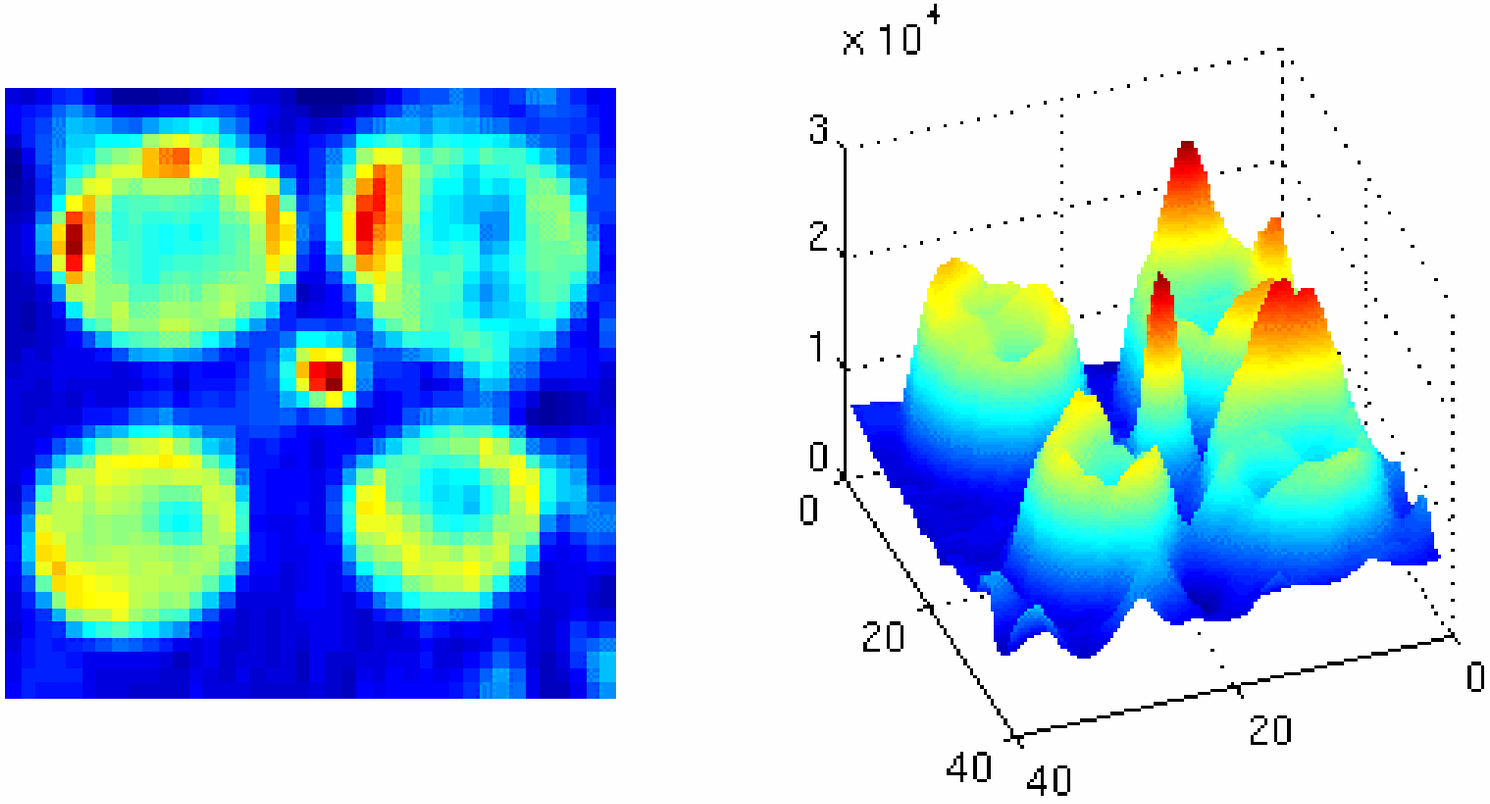
These arrays were printed with duplicate spots. The top 5 rows are duplicated in the bottom 5 rows. Note the “ring” or “donut” effect that is evident at many spots, especially in Cy5. [Deegan et al (1997). Capillary flow as the cause of ring stains from dried liquid drops. Nature, v.389, p.827-9.]

A closer look at one spot



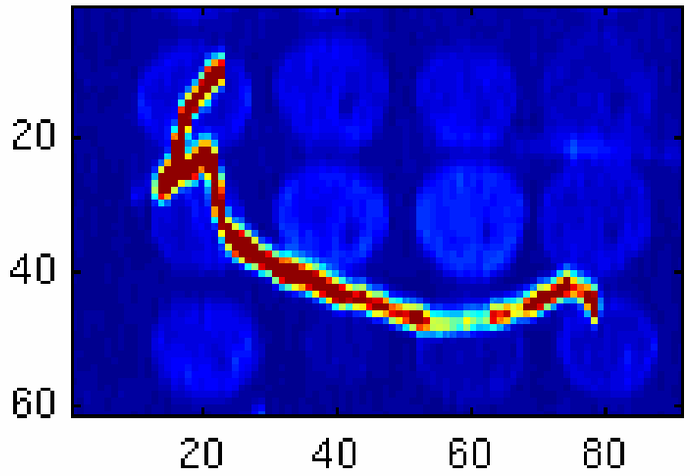
Notes: (1) The spot is not perfectly uniform; higher edges are evident even in bright spots. (2) The background is not uniform. (3) The spot appears to be contained in a box with 22×22 pixels.

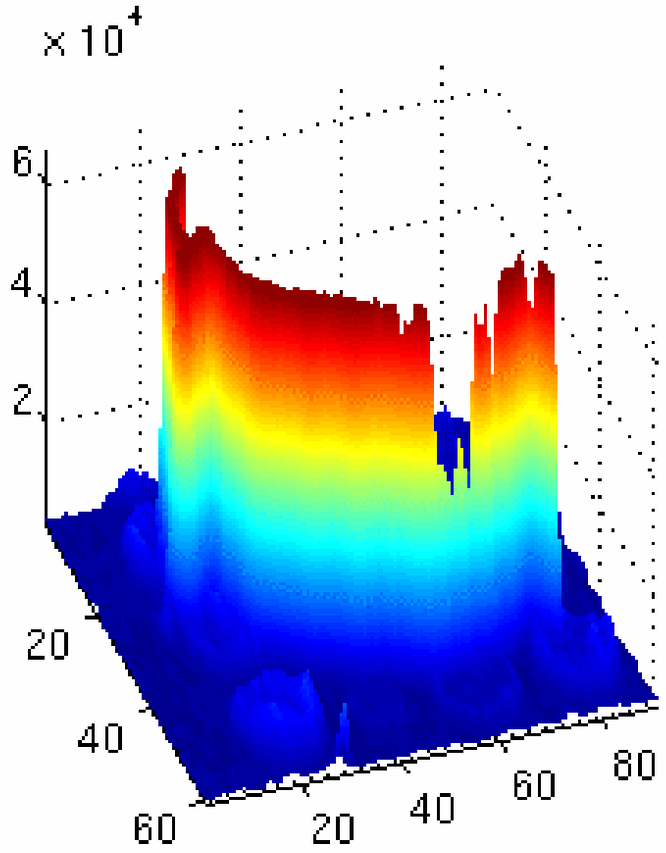
A dust speck



Automatic processing algorithms to locate the spots have to contend with large dust specks; quantification algorithms can be affected by smaller specks that overlap true spots.

A fiber





Why do you think the pixel values are uniform along the fiber?

Summary: glass microarrays

We've seen a number of potential difficulties with analyzing the images from glass microarrays.

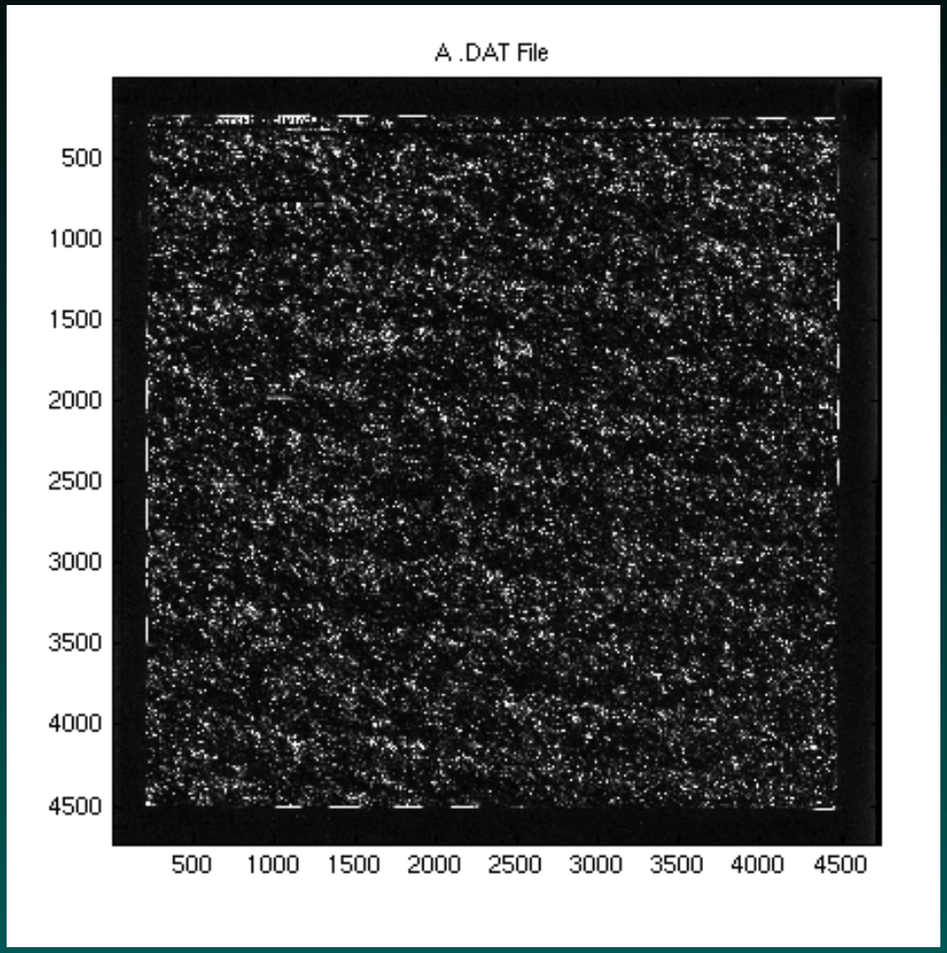
- Artifacts like dust specks, fibers, or water spots can cause small-scale problems with spot-finding or quantification.
- Differences in pins can cause systematic biases on the scale of subgrids.
- Channel differences, either directly related to chemical propoerties of the dyes or differences in laser intensity, can introduce systematic distortions.
- Insufficient mixing of the hybridization solution can cause large-scale differences in background and signal intensity.

The Structure of Affymetric Microarrays

Recall: Affymetrix arrays contain short (25-mer) oligonucleotide probes synthesized directly onto a silicon substrate. Deposition uses photolithography, the same technology used by Intel to produce computer chips. Compared to glass microarrays, this method allows denser feature packing, but forces the use of smaller sequences.

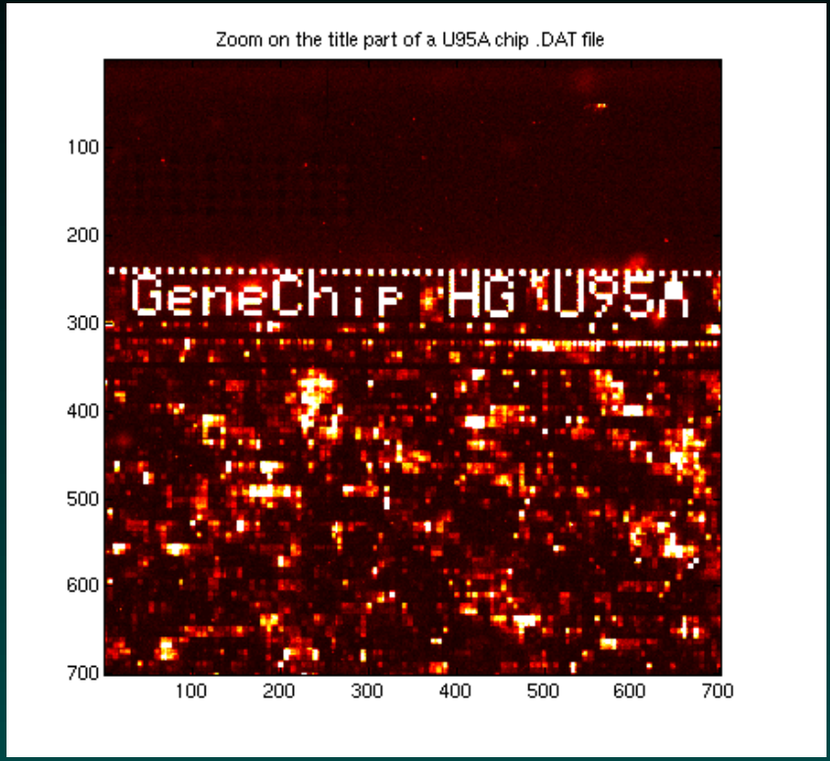
Short sequences may not bind their complement as well as long sequences; they may also more easily cross-hybridize with unintended targets.

An Affymetrix GeneChip microarry image



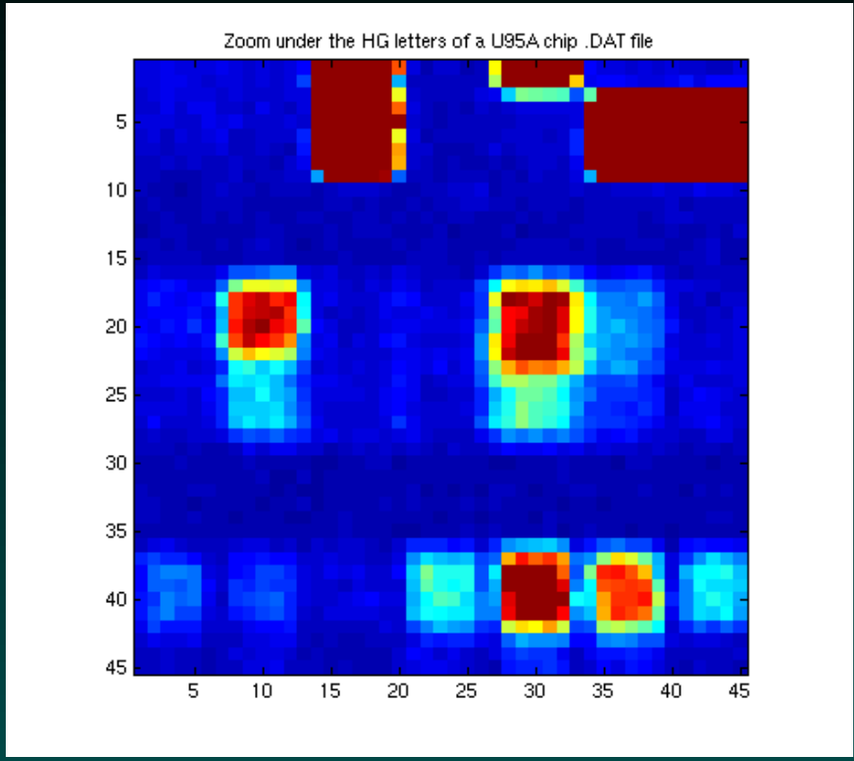
The array shown here has features in a 640×640 grid, for a total of $409,600$ features. Each feature represents a 25-bp probe.

Closeup of a GeneChip image



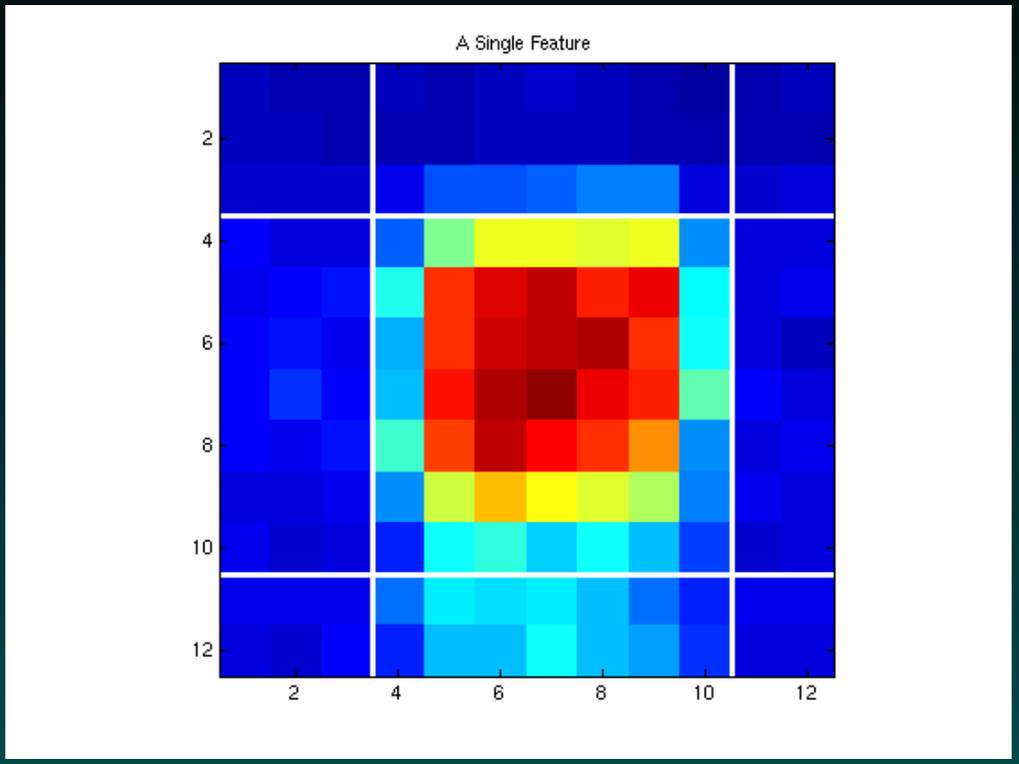
The pixelated features have been combined with positive controls to spell out the chip type – this helps ensures that the image is correctly oriented. Also note the border lattice of alternating dark and bright QC probes, making image alignment and feature detection easier.

GeneChip features



Features are squares instead of circles. Horizontal and vertical alignment with the edges of the image are pretty good. However, feature boundaries can be rather blurry.

GeneChip features

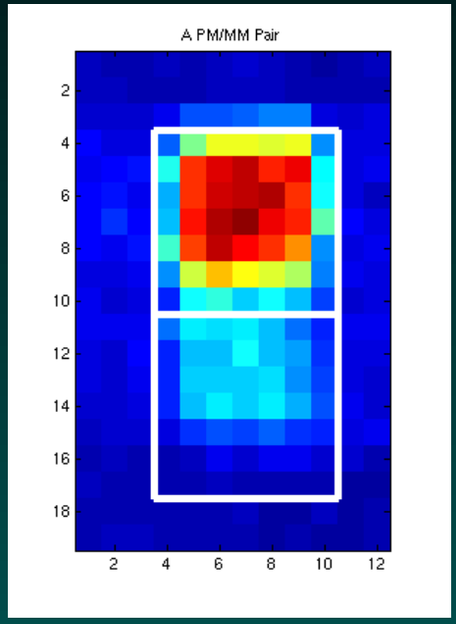


Each feature is approximately 7 pixels on a side. In general, Affymetrix features use many fewer pixels than are used for the round spots in the images of glass microarrays.

As with the glass array images, spots are not uniform.

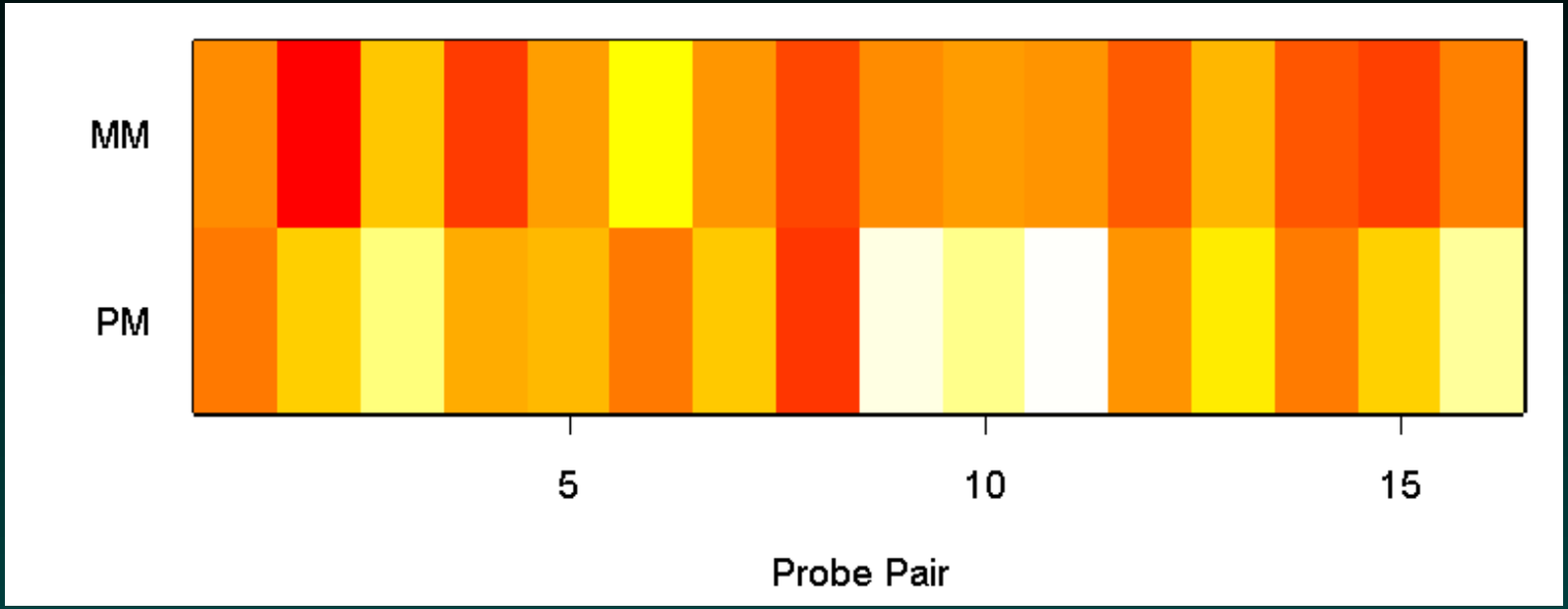
Perfect Match and Mismatch (Probe-Pairs)

PM: GCTAGTCGATGCTAGCTTACTAGTC
MM: GCTAGTCGATGCAAGCTTACTAGTC



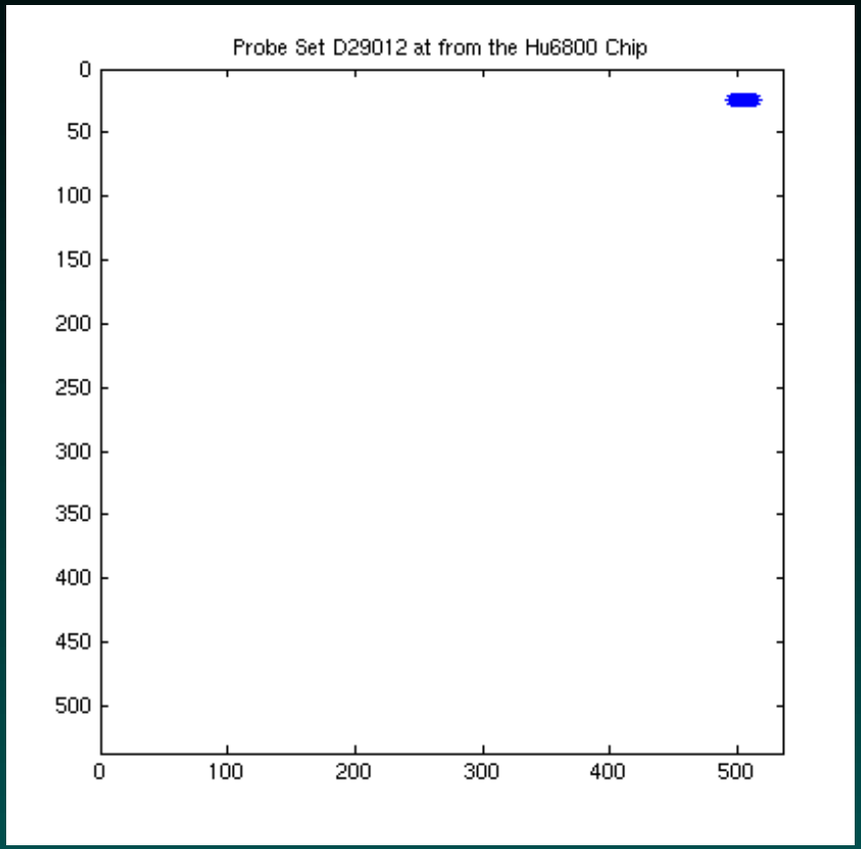
Affymetrix tries to control for cross-hybridization by using mutliple probes along with Perfect Match (PM) and Mismatch (MM) probes. The PM probe is always placed directly above the MM probe on the GeneChip.

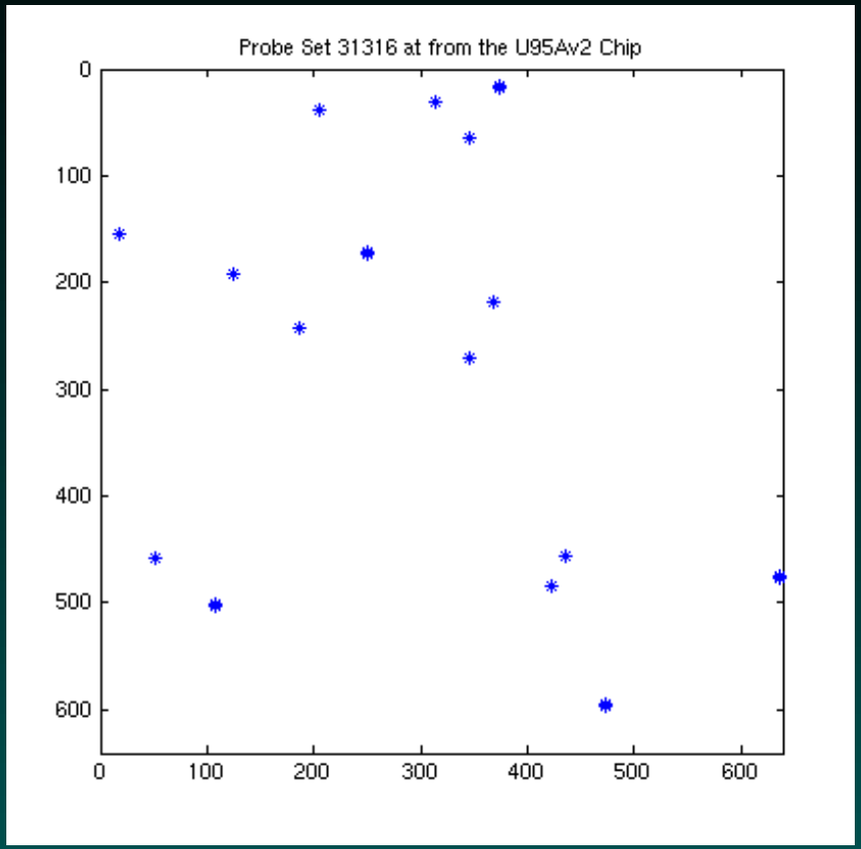
Probe Sets



For each target gene on an Affymetrix array,use between 11 and 20 probe-pairs. (The oldest GeneChips used 20 probe-pairs per target gene; the second generation used 16; the newest arrays use 11.) The first basic challenge in quantifying Affymetrix arrays is summarizing the 22 to 40 numbers from a probe set with a single estimate of the expression of the target gene.

Probe-Set Locations





Probe set locations change with the chip type. In older chips, the probe pairs were adjacent. In newer chips, they are randomized.

Data file formats

As we will see next week, there are many different scanners and many different commercial software packages to quantify the spots in an image from a glass microarray. These packages use different algorithms and measure different properties of the spots. While it is important to understand the metrics and algorithms used by those programs, they all have one thing in common: they save the data in plain text files. Each row represents a spot, and each column represents a measurement, with entries separated by tabs. This common format makes it very easy to read the data into a statistical package like R.

Affymetrix Data file formats

By contrast, all Affymetrix GeneChips are scanned in an Affymetrix scanner, and the initial quantification of features is performed using Affymetrix software. (The main difference of opinion arises in how to combine the feature quantifications from a probe set.) The software involves numerous files.

EXP Contains basic information about the experiment.

DAT Contains the raw image.

CEL Contains features Quantifications.

CDF Maps between features, probes, probe-sets, and genes.

CHP Contains gene expression levels.

Part of an EXP file

```
Affymetrix GeneChip Experiment Information
Version 1

[Sample Info]
Chip Type        HG_U95Av2
Chip Lot
Operator
Sample Type
Description
Project
Comments
Solution Type
Solution Lot
```

[Fluidics]
Protocol
Station
Module
Hybridize Date

[Scanner]
Pixel Size
Filter
Scan Temperature
Scan Date
Scanner ID
Number of Scans
Scanner Type

The DAT file

An example was shown earlier. Contains a 16-bit intensity image in a proprietary format. The file structure consists of a 512 byte header followed by the raw image data.

The CEL file

Contains the feature quantifications.

Through version 3, this was a plain text file. In version 4, the format changed to binary to permit more compact storage of the data. Affymetrix provides a free tool to convert between the file formats.

In the plain text version, sections are demarcated by headers in brackets, as in the EXP file. The header tells us which DAT file it came from, the feature geometry (e.g., 640×640, the pixel location of the grid corners in the DAT file, and the quantification algorithm used. This is followed by the actual measurements, consisting of the X and Y feature locations, the mean and standard deviation, and the number of pixels in the feature.

The CDF file

With any set of microarray experiments, one of the major challenges is keeping track of how the feature quantifications map back to information about genes, probes, and probe sets. There is one CDF file for each type of GeneChip, which contains this information.

```
[CDF]
Version=GC3.0

[Chip]
Name=HG_U95Av2
Rows=640
Cols=640
NumberOfUnits=12625
MaxUnit=102119
```

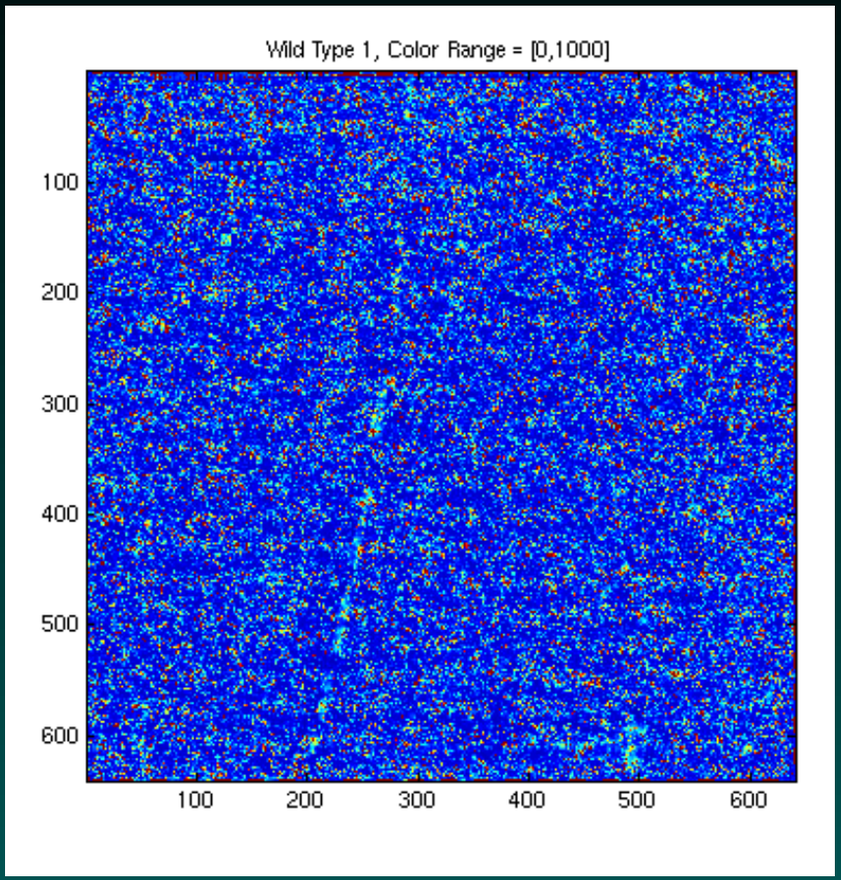
```
NumQCUnits=13
ChipReference=

....

[Unit250]
Name=NONE
Direction=2
NumAtoms=16
NumCells=32
UnitNumber=250
UnitType=3
NumberBlocks=1
```

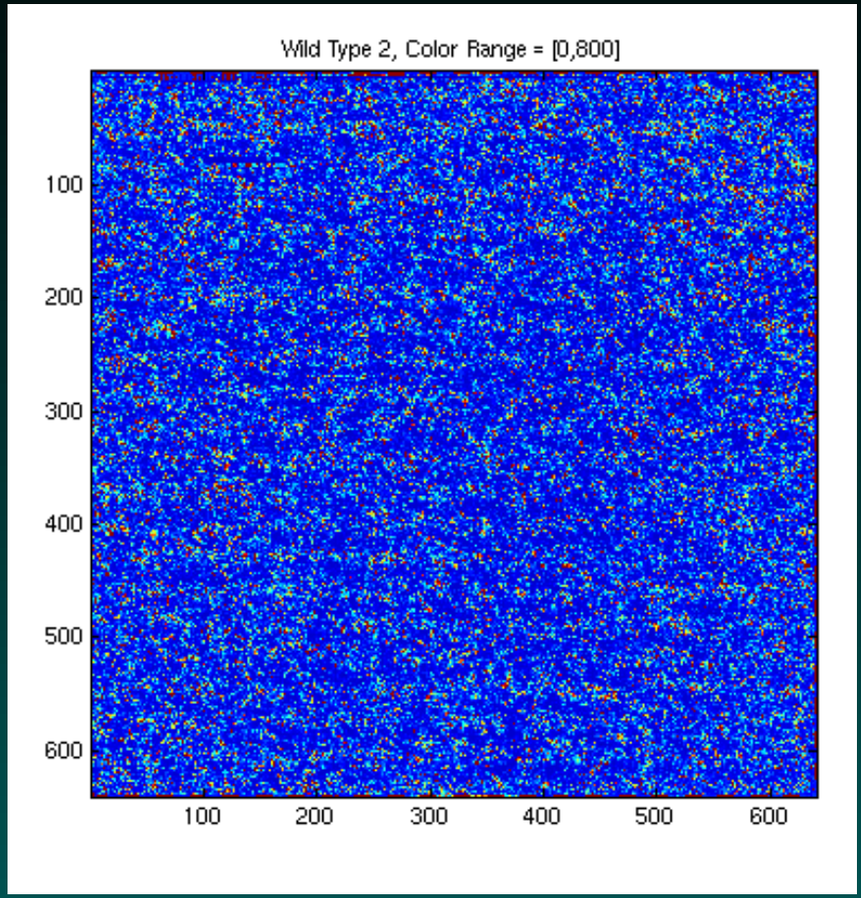
CDF file entries

```
[Unit250_Block1]
Name=31457_at
BlockNumber=1
NumAtoms=16
NumCells=32
StartPosition=0
StopPosition=15
CellHeader=X Y          PROBE   FEAT        QUAL        EXPOS
             POS        CBASE   PBASE       TBASE       ATOM     IN
             CODONIND   CODON   REGIONTYPE  REGION
Cell1=517    568        N       control     31457_at    0
             13         A       A           A           0        36
             -1         -1      99
Cell2=517    567        N       control     31457_at    0
             13         A       T           A           0        36
             -1         -1      99
Cell3=78     343        N       control     31457_at    1
             13         T       A           T           1        21
```

```
             -1      -1      99
Cell4=78     344     N       control     31457_at    1
             13      T       T           T           1       22
             -1      -1      99
Cell5=314    78      N       control     31457_at    2
             13      A       A           A           2       50
             -1      -1      99
```

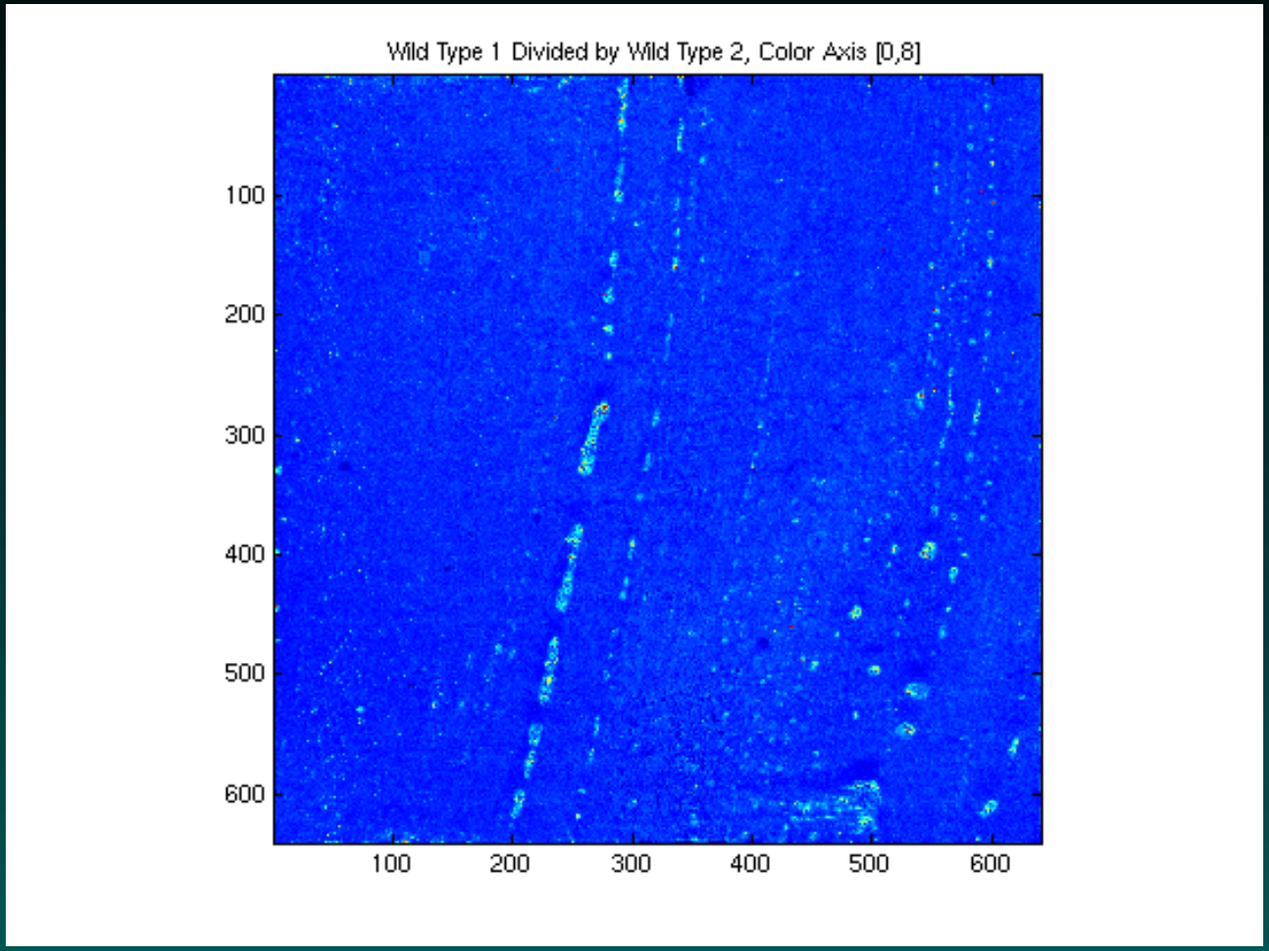
Replicate mouse chips





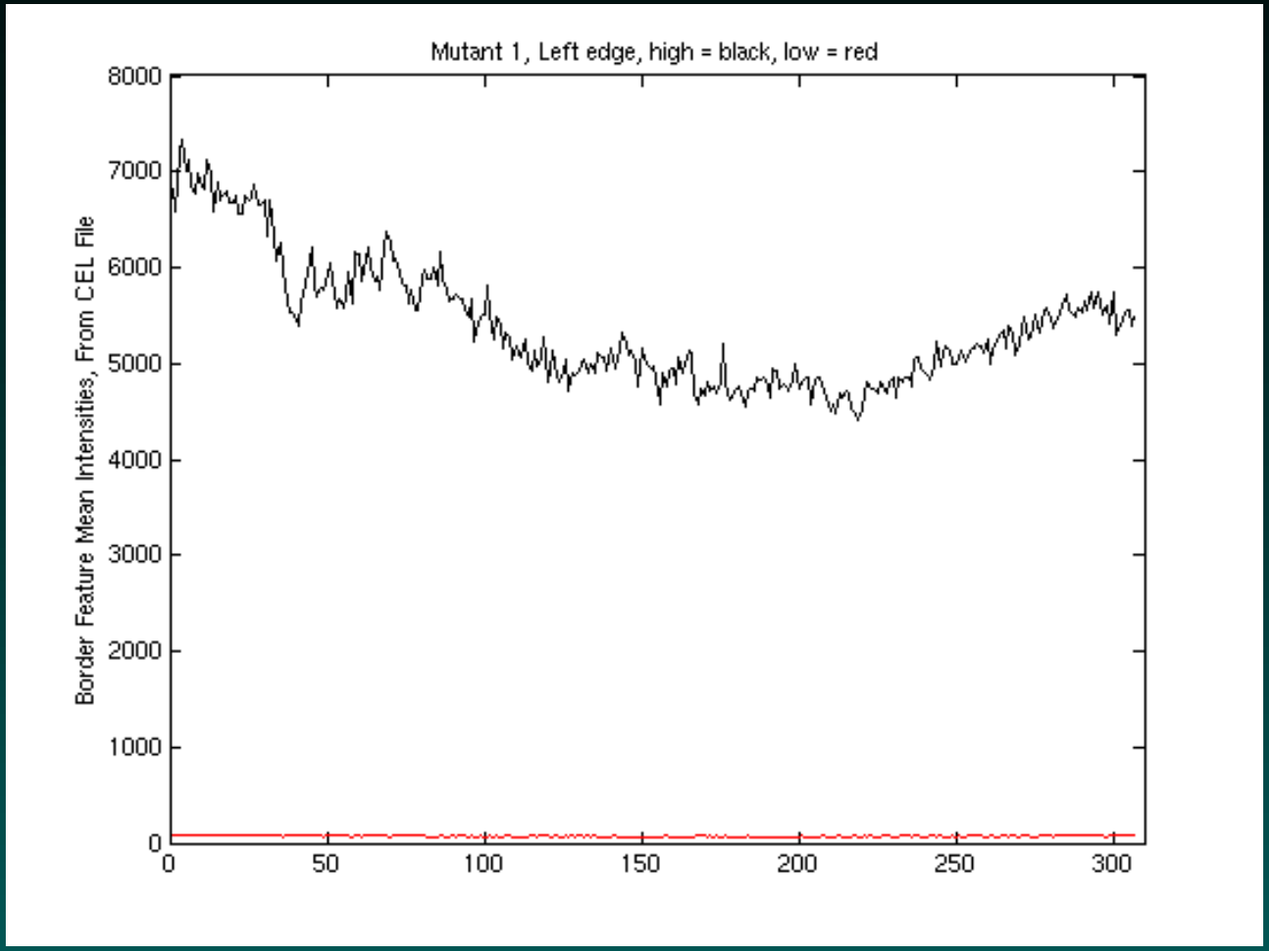
There is a slight of hint of some artifactual contamination in the image on the left.

Image plot of the difference



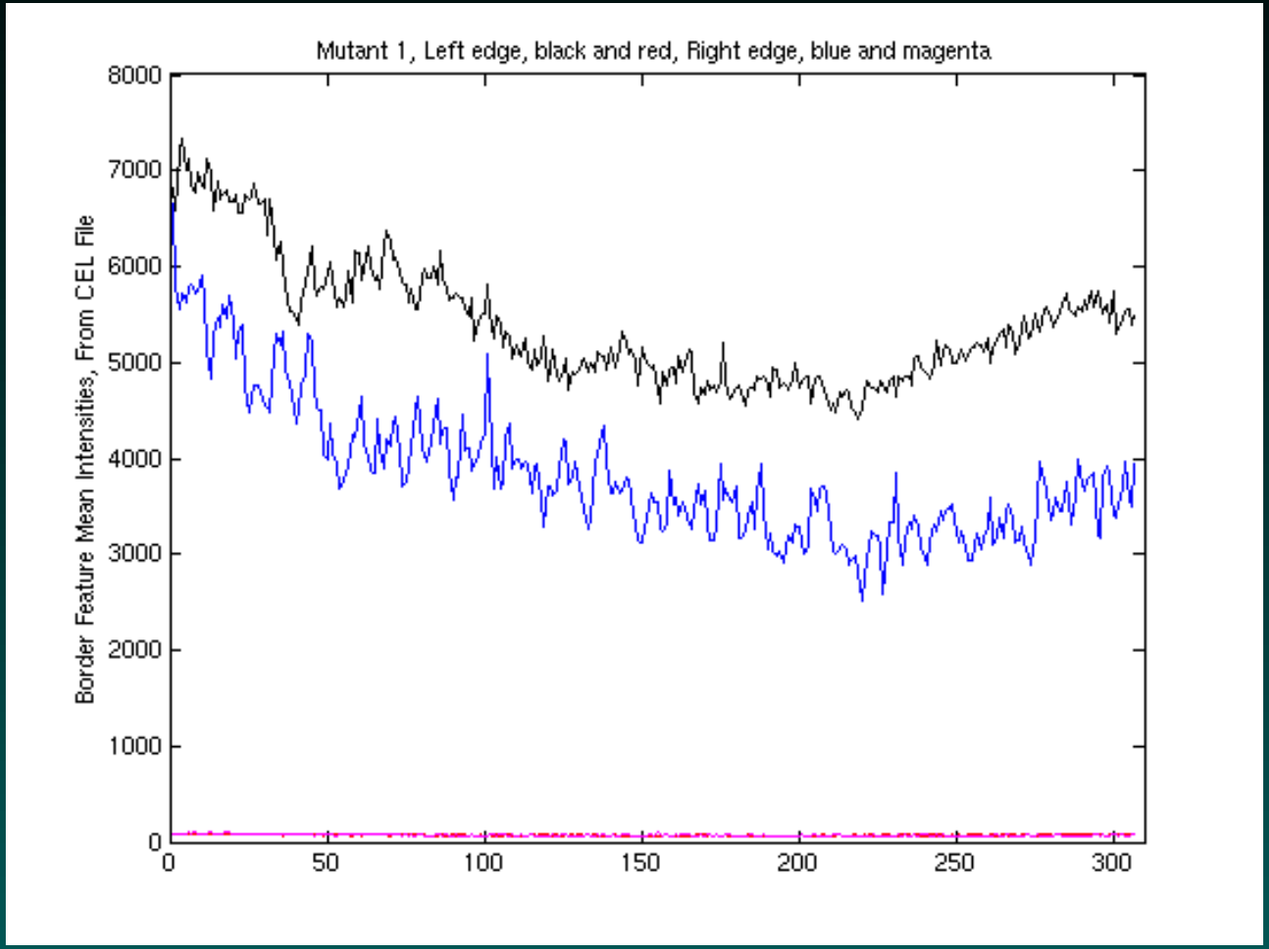
This method does not require exact replicates; merely similar samples and the same chip type.

Within chip replicates: Edge plots

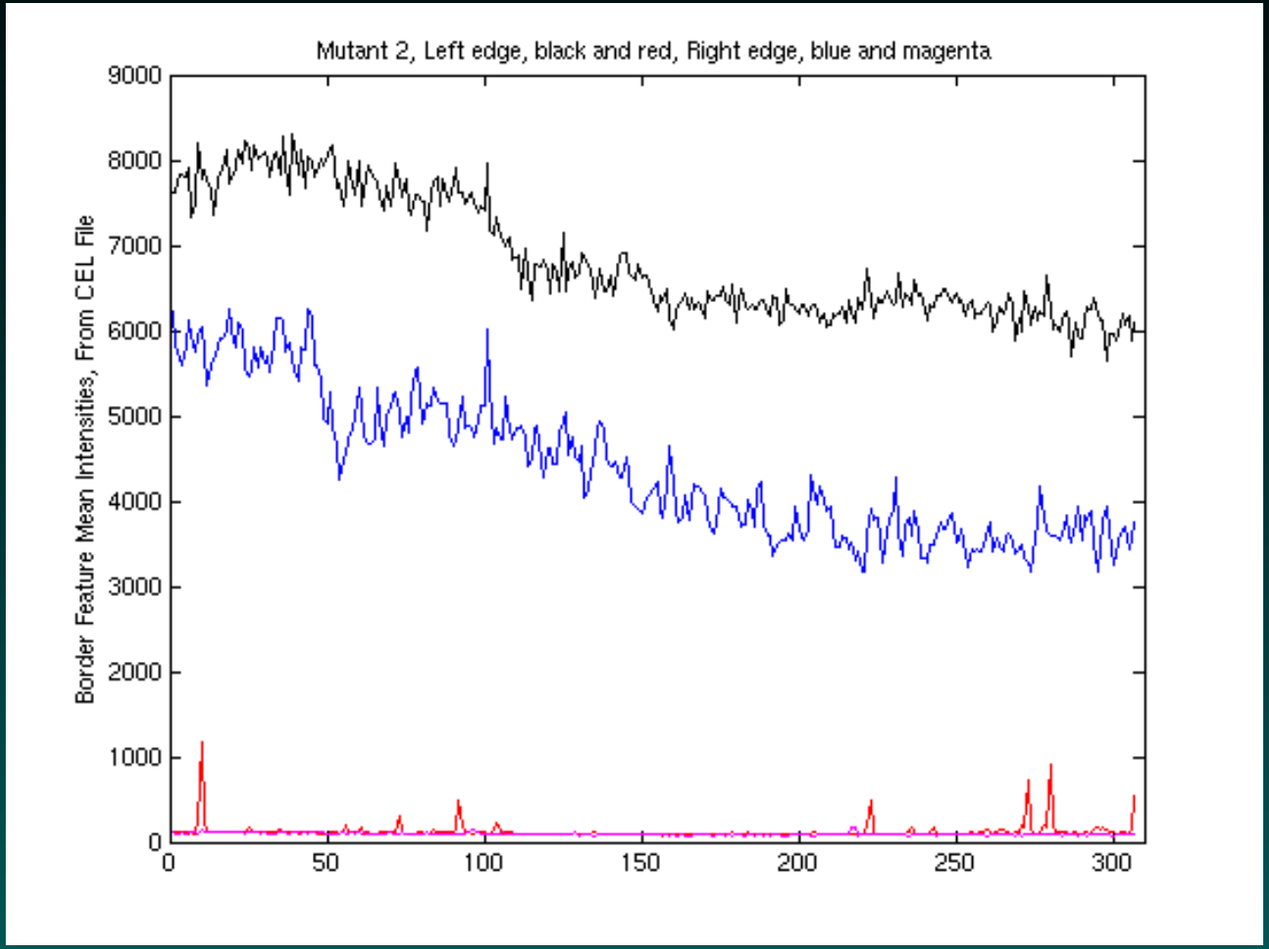


Takes advantage of the alternating pattern of high and low intensity QC spots around the edges.

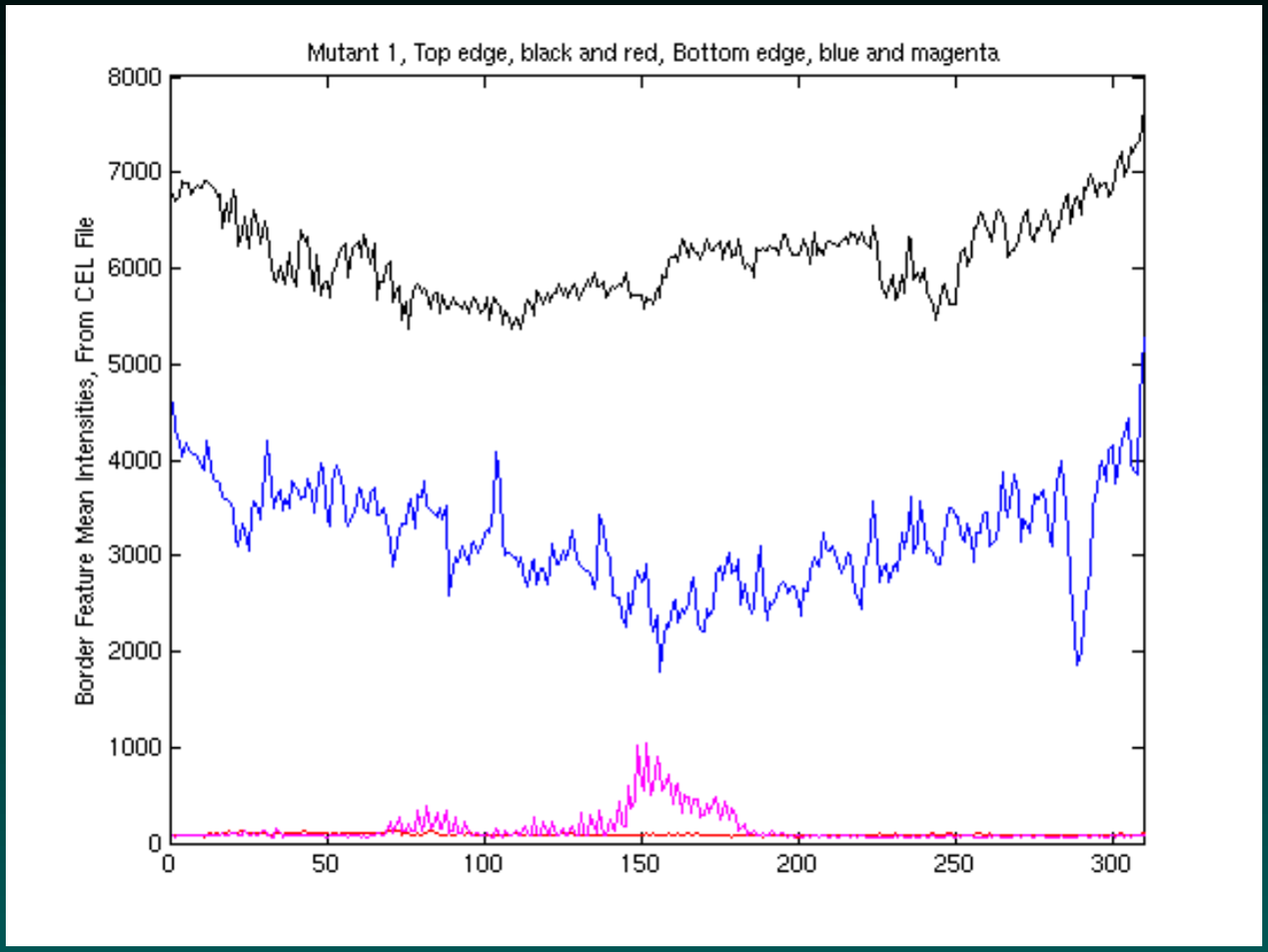
Side to side: There is a gradient



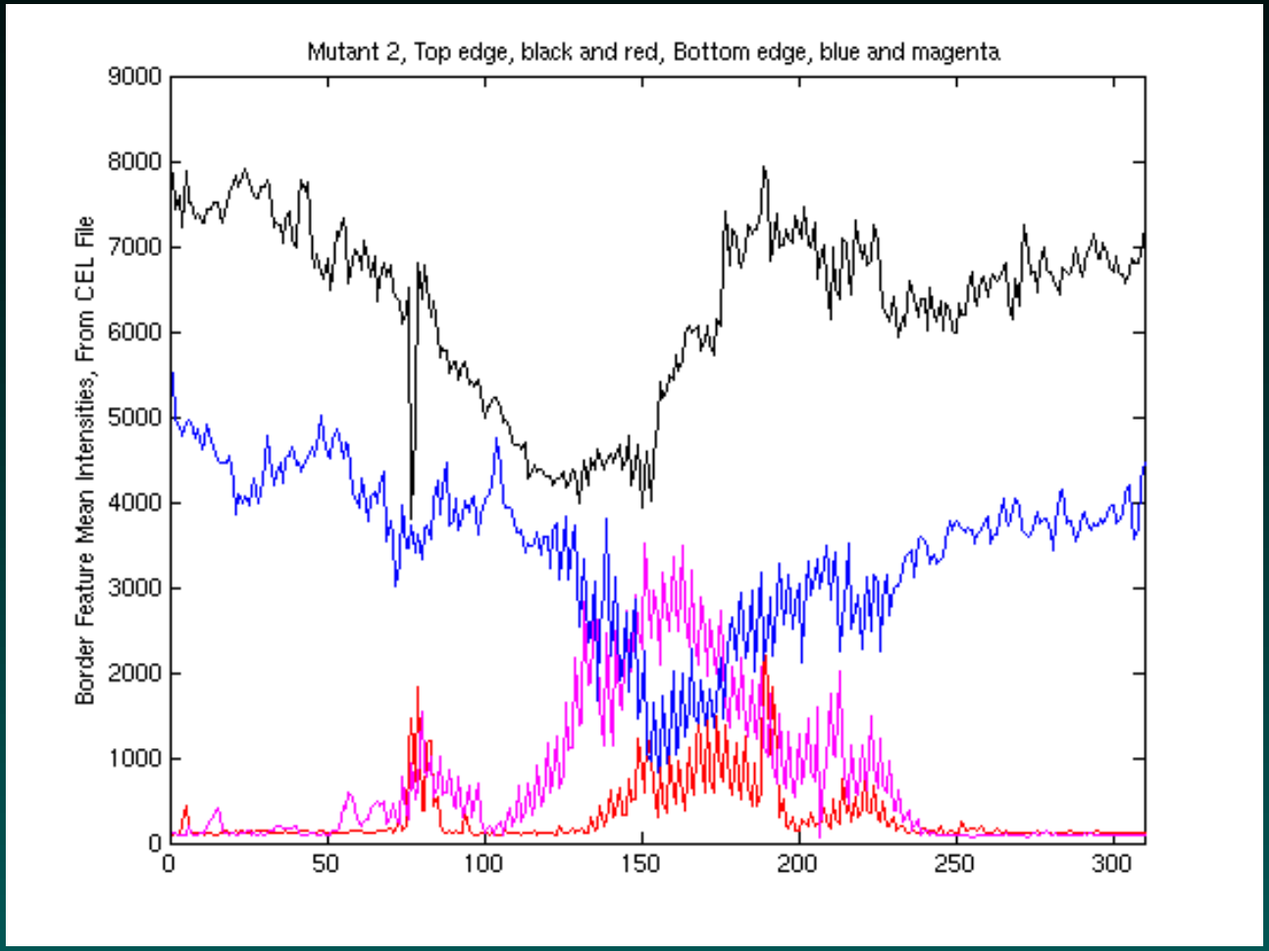
Side-to-side: A different chip



Top-to-bottom



Top-to-bottom: The second chip



Scanner needed to be serviced, and the feature boundaries drifted out of alignment across the slide.