GS01 0163
Analysis of Microarray Data

Keith Baggerly and Kevin Coombes
Department of Bioinformatics and Computational Biology
UT M. D. Anderson Cancer Center
kabagg@mdanderson.org
kcoombes@mdanderson.org

August 28, 2007

Lecture 1: The Structure of Affymetrix Data

- What are we trying to measure, and how?
- Mechanics – the data files produced and used
- Getting numbers: normalization and quantification

The role of gene expression

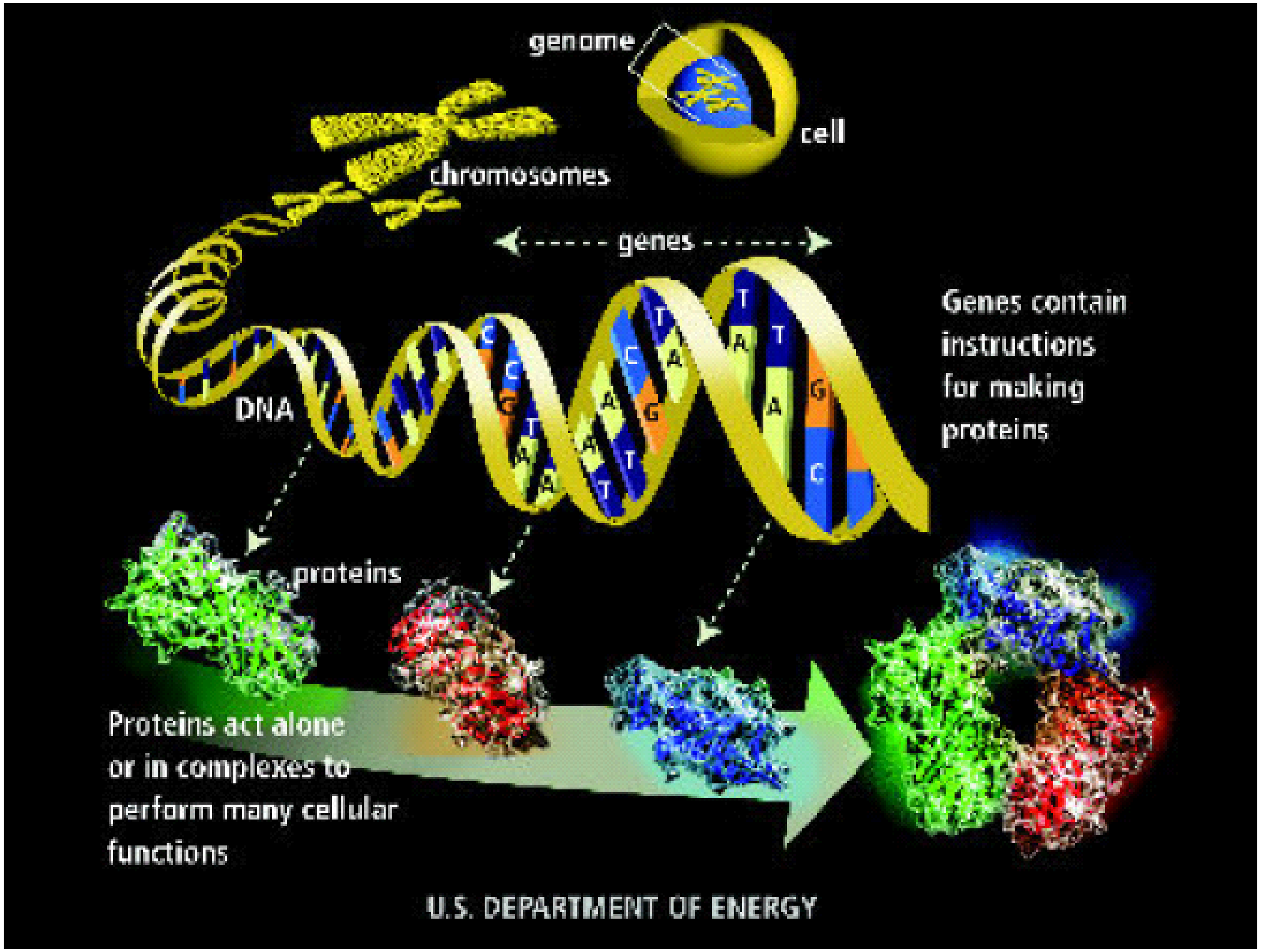
Your genome consists of pairs of DNA molecules (chromosomes) held together by complementary nucleotide base pairs (in total, about 3×10^9 base pairs). The structure of DNA provides an explanation for heredity, by copying individual strands and maintaining complementarity.

All of your cells contain the same genetic information, but your skin cells are different from liver cells or kidney cells or brain cells.

These differences come about because different genes are expressed at high levels in different tissues.

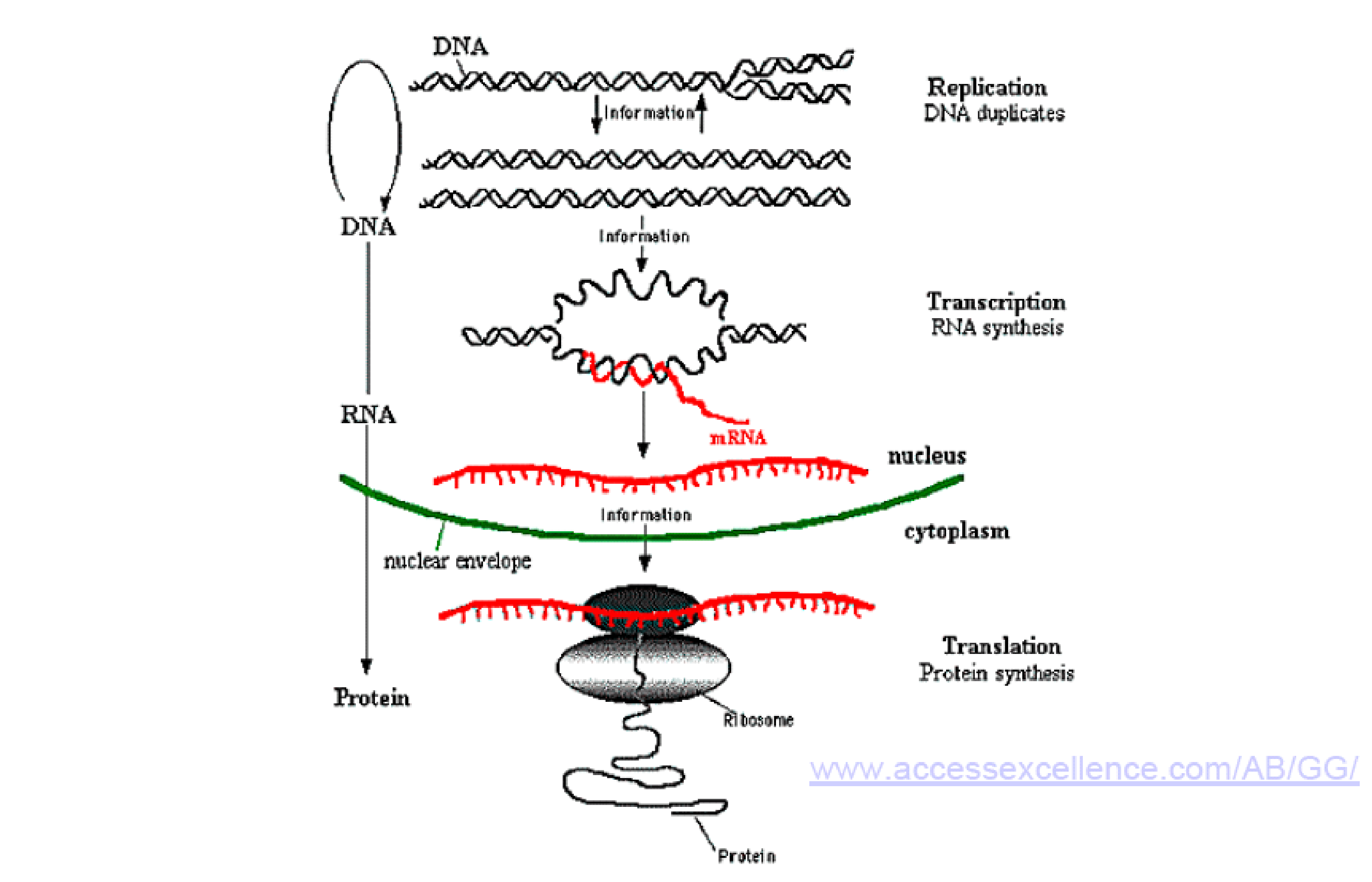
So, how are genes “expressed”?

The Central Dogma: DNA and Protein


genome
cell
chromosomes
genes
Genes contain instructions for making proteins
DNA
proteins
Proteins act alone or in complexes to perform many cellular functions
U.S. DEPARTMENT OF ENERGY

www.ornl.gov/hgmis/graphics/slides/images1.html

The Central Dogma: RNA



The Central Dogma

DNA makes RNA makes protein

DNA – sequence

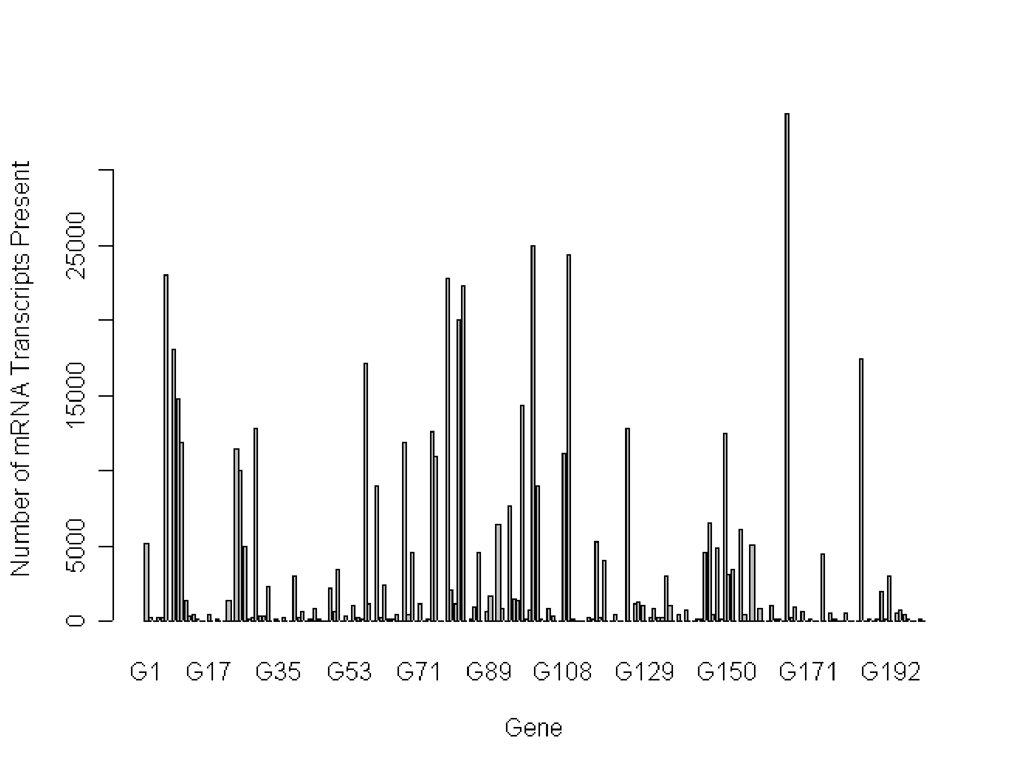
mRNA – sequence and abundance

protein – sequence, abundance and shape

Microarrays measure mRNA expression.

An idealized expression profile

If we could count the number of mRNA molecules from each gene in a single cell at a particular time, we might get this:



But how do we make these measurements?

How do microarrays work?

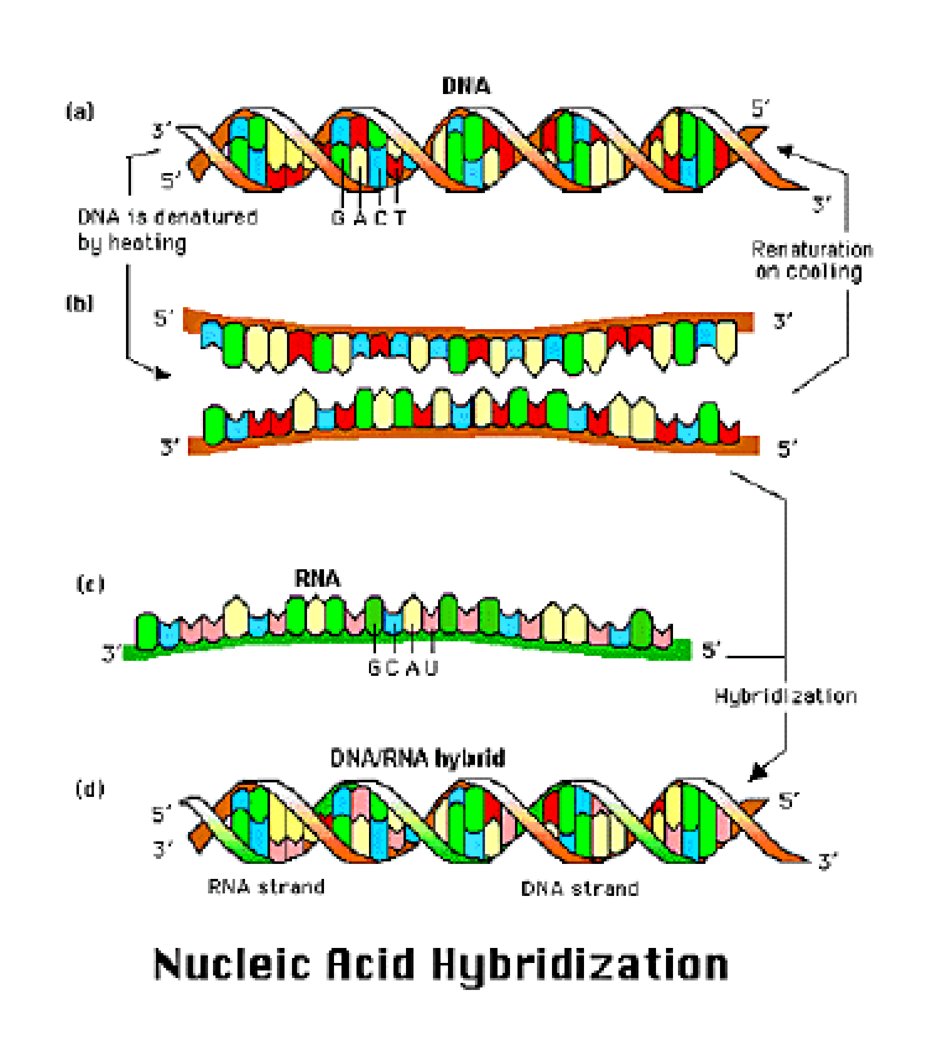
The biological principle involved is that sequences of DNA or RNA molecules containing complementary base pairs have a natural tendency to bind together.

```
....AAAAAGCTAGTCGATGCTAG....
....TTTTTCGATCAGCTACGATC....
```

If we know the target mRNA sequence we can build a probe for it using the complementary sequence. The probe location tells us the identity of the gene. Two variants are common:

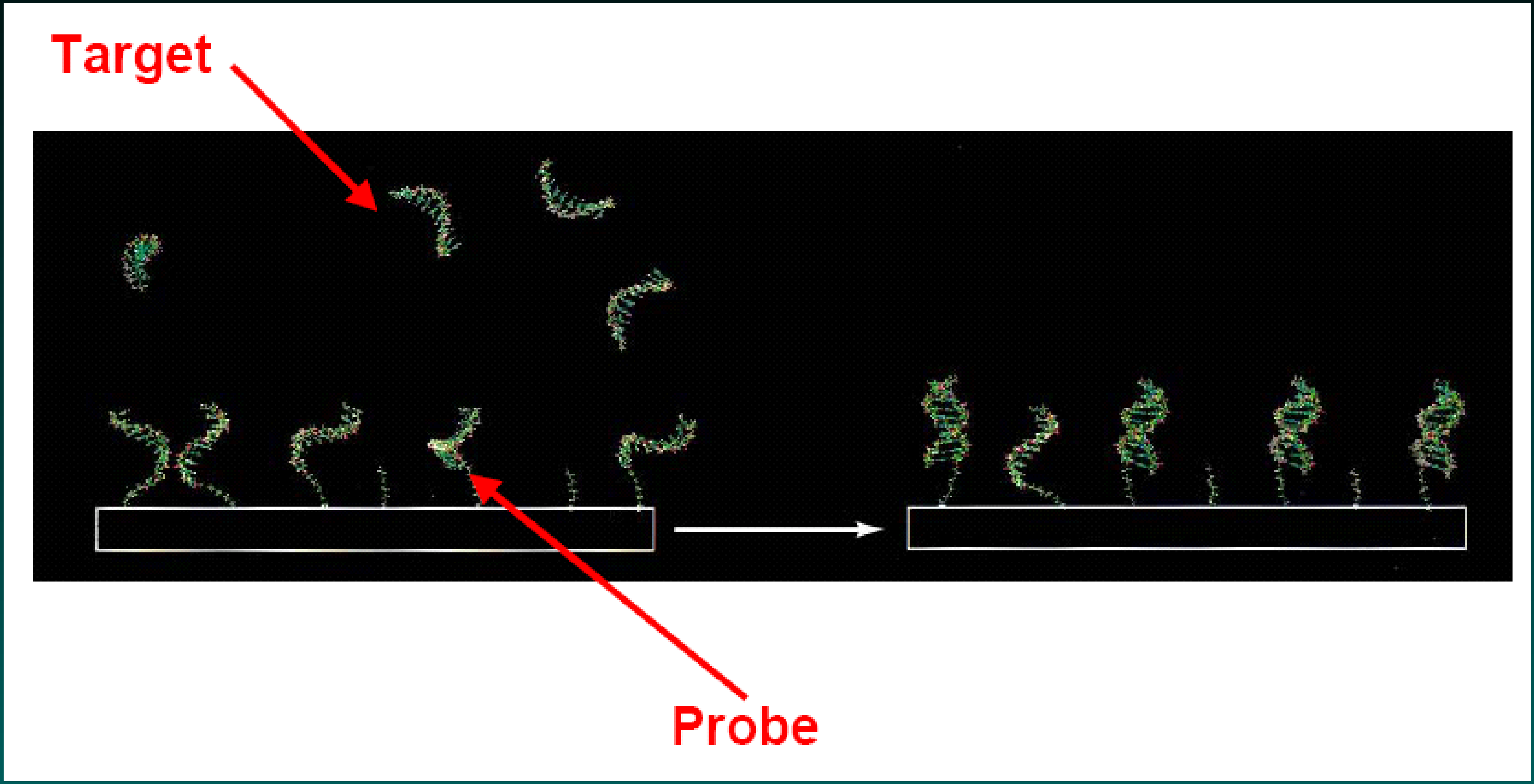
- Reverse transcription from mRNA to cDNA
- Photolithographic synthesis of short subsequences (oligos)

Hyb Pic 1



from Bioconductor, ENAR03 Workshop Slides

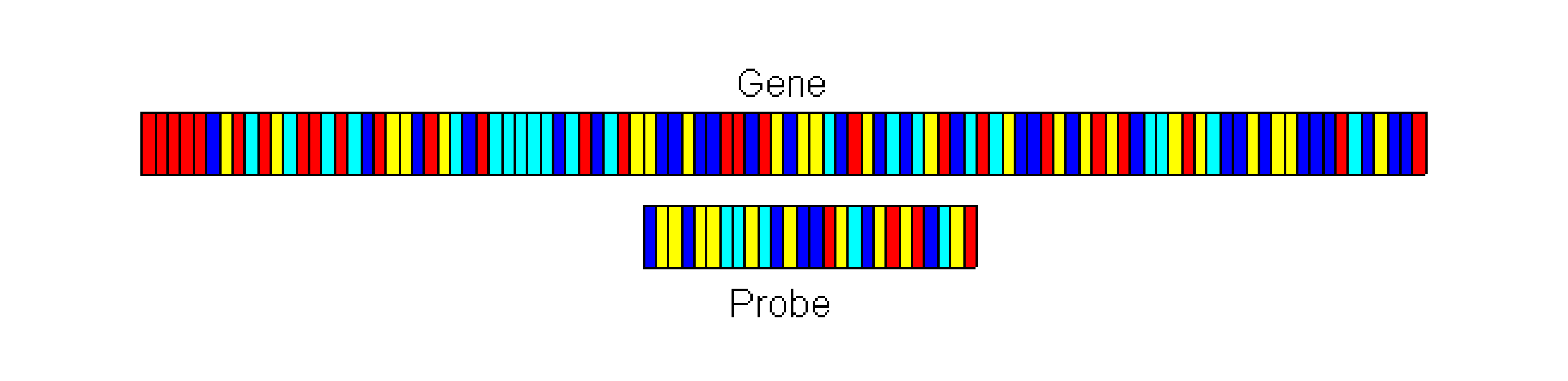
Hyb Pic 2



from Bioconductor, ENAR03 Workshop Slides

How do Affymetrix chips work?

In general, the probes are shorter than the genes. This is driven by the manufacturing process.



Critical note: Different probes for the same gene have different binding affinities. These affinities are unknown. Thus, it's difficult to say "gene A beats gene B", as opposed to "there's more gene A here than there was there". Microarrays produce relative measurements of gene expression.

How do we choose probes?

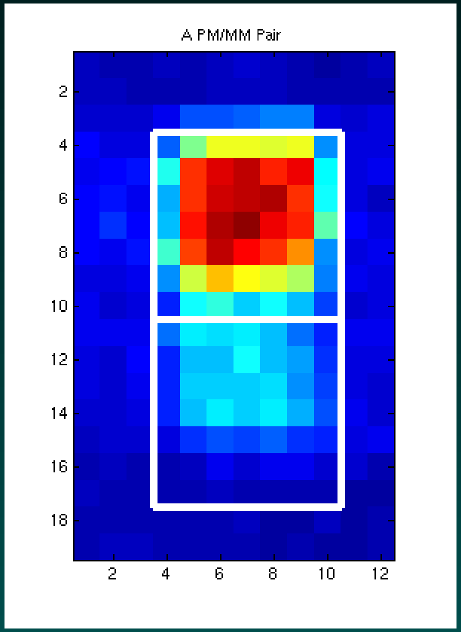
Given unknown affinities, we can provide some security by using several different probes for the same gene, but the optimal number is not clear. Sequential generations of Affy chips have used 20, 16, and 11 probes.

Some further difficulties:

- some genes are short (overlap)
- genes have an orientation (3' bias)
- the gene may not be what we think it is (DB evolution)
- probes can "cross-hybridize" to the wrong targets

Perfect Match and Mismatch (Probe-Pairs)

PM: GCTAGTCGATGCTAGCTTACTAGTC
MM: GCTAGTCGATGCAAGCTTACTAGTC



Affymetrix has tried to control for cross-hybridization by coupling multiple probes that should work paired with those that shouldn't. These are known as the Perfect Match (PM) and Mismatch (MM) probes, and constitute “probe pairs”.

The Ensemble: A Probeset

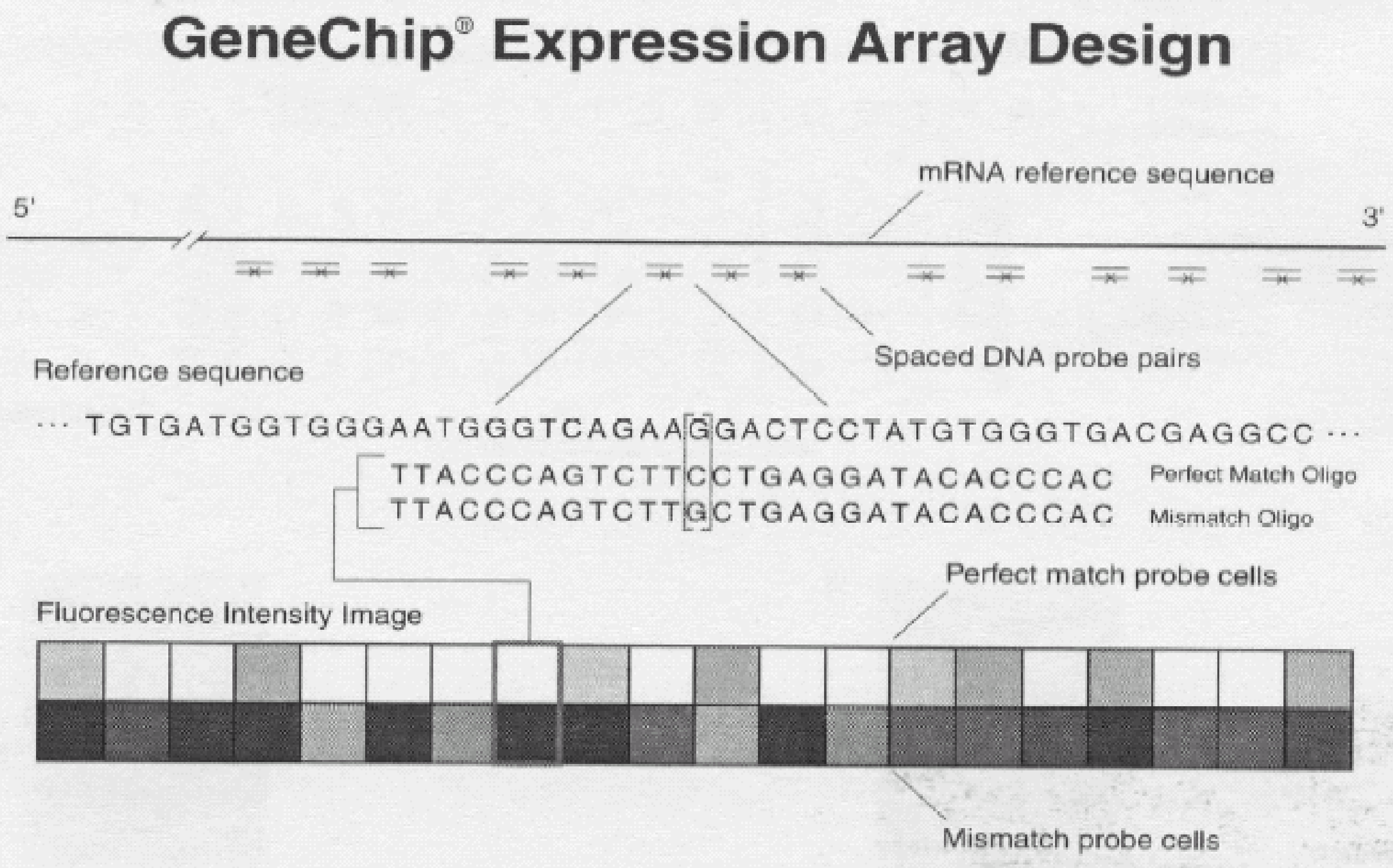

GeneChip® Expression Array Design
mRNA reference sequence
5'
3'
Spaced DNA probe pairs
Reference sequence
··· TGTGATGGTGGGAATGGGTCAGAAGGACTCCTATGTGGGTGACGAGGCC ···
TTACCCAGTCTTCCTGAGGATACACCCAC
Perfect Match Oligo
TTACCCAGTCTTGCTGAGGATACACCCAC
Mismatch Oligo
Perfect match probe cells
Fluorescence Intensity Image
Mismatch probe cells

Figure 1-3 Expression tiling strategy

Affy Microarray Suite 4.0 User Guide

How de we measure amounts bound?

When we extract mRNA from a sample of cells, we do not measure this mRNA directly. Rather, we make copies, and incorporate molecules of fluorescent dye into the copies. We can measure fluorescence!

Thus, after selecting the desired probes for all genes of interest, these are printed on a solid substrate. Samples containing the target genes are processed, labeled with a fluorescent dye, hybridized to the array, and scanned, producing an image file.

The image is the data.

How de we measure amounts bound?

When we extract mRNA from a sample of cells, we do not measure this mRNA directly. Rather, we make copies, and incorporate molecules of fluorescent dye into the copies. We can measure fluorescence!

Thus, after selecting the desired probes for all genes of interest, these are printed on a solid substrate. Samples containing the target genes are processed, labeled with a fluorescent dye, hybridized to the array, and scanned, producing an image file.

The image is the data.

Ah, but what *files* do we get?

Affymetrix Data file formats

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.

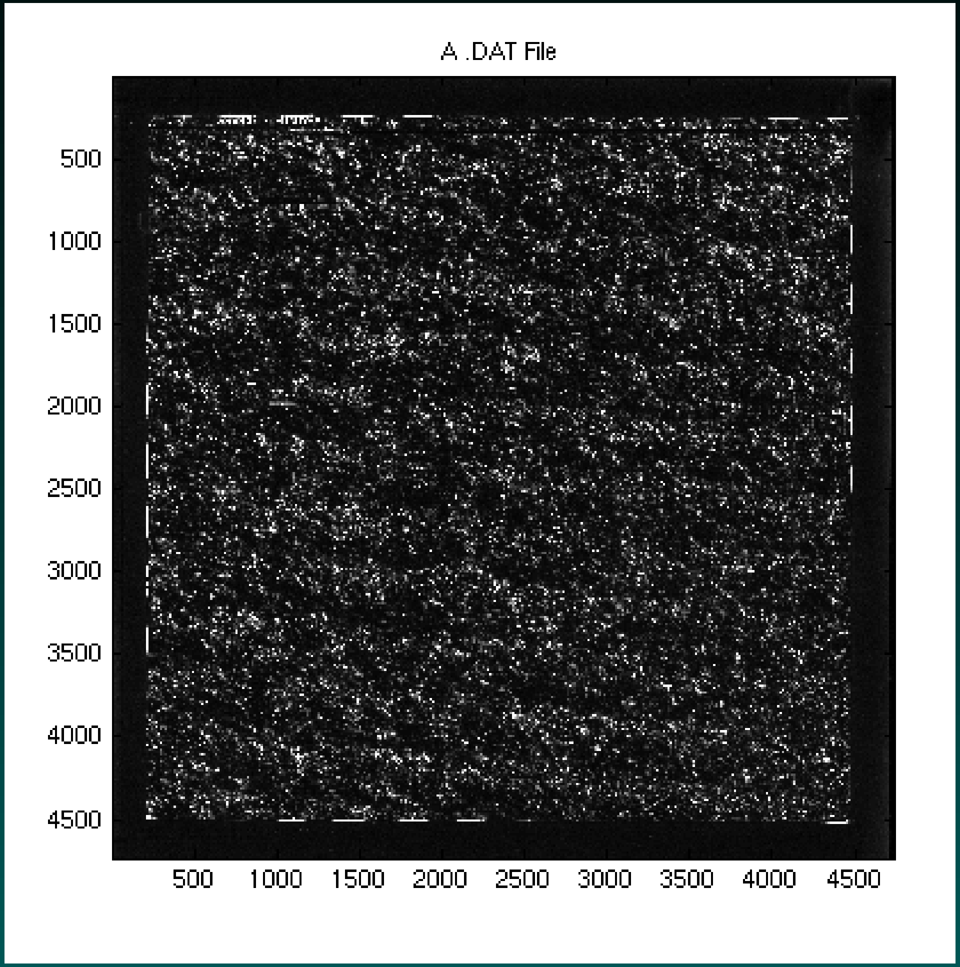
DAT Contains the raw image.

CEL Contains features Quantifications.

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

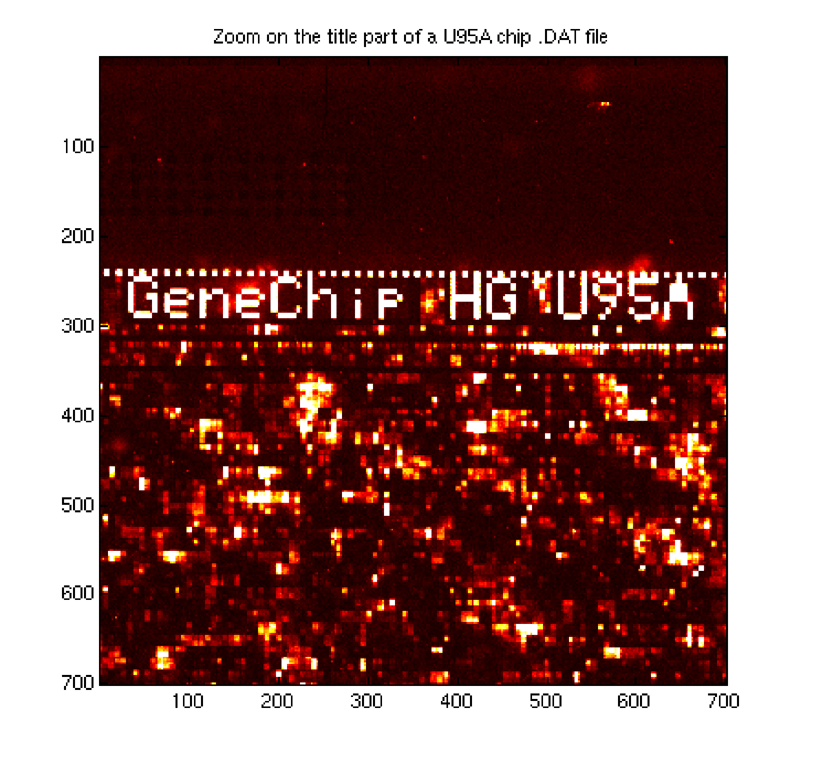
CHP Contains gene expression levels, as assessed by the Affy software.

An Affymetrix GeneChip microarray image



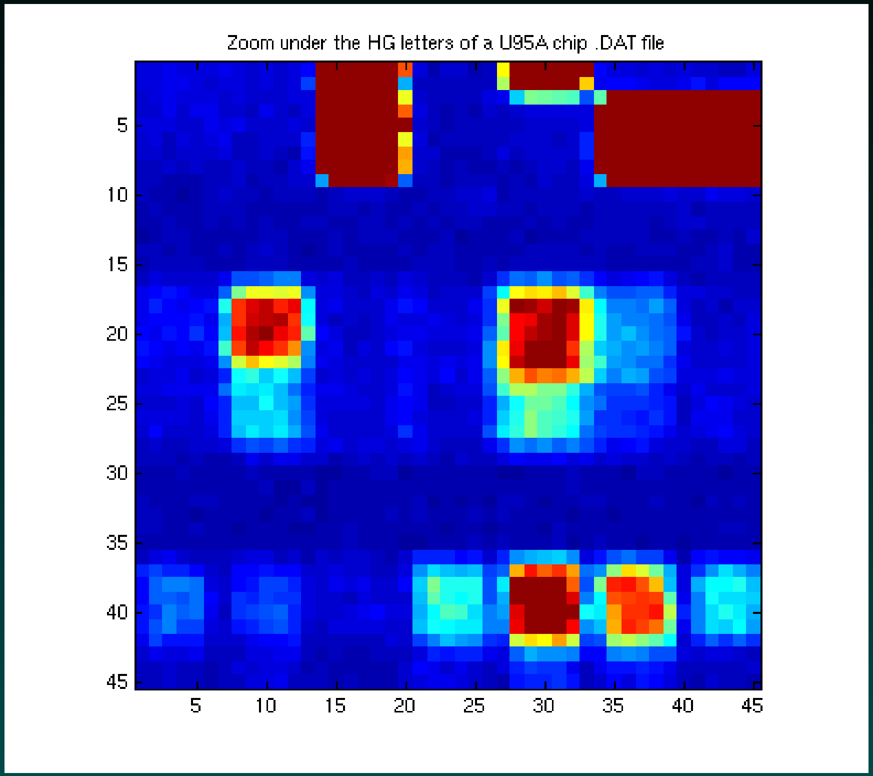
This array has a 640×640 grid of "features" $(409, 600$ total). A "feature" has many copies of one 25-bp seq.

Closeup of a GeneChip image



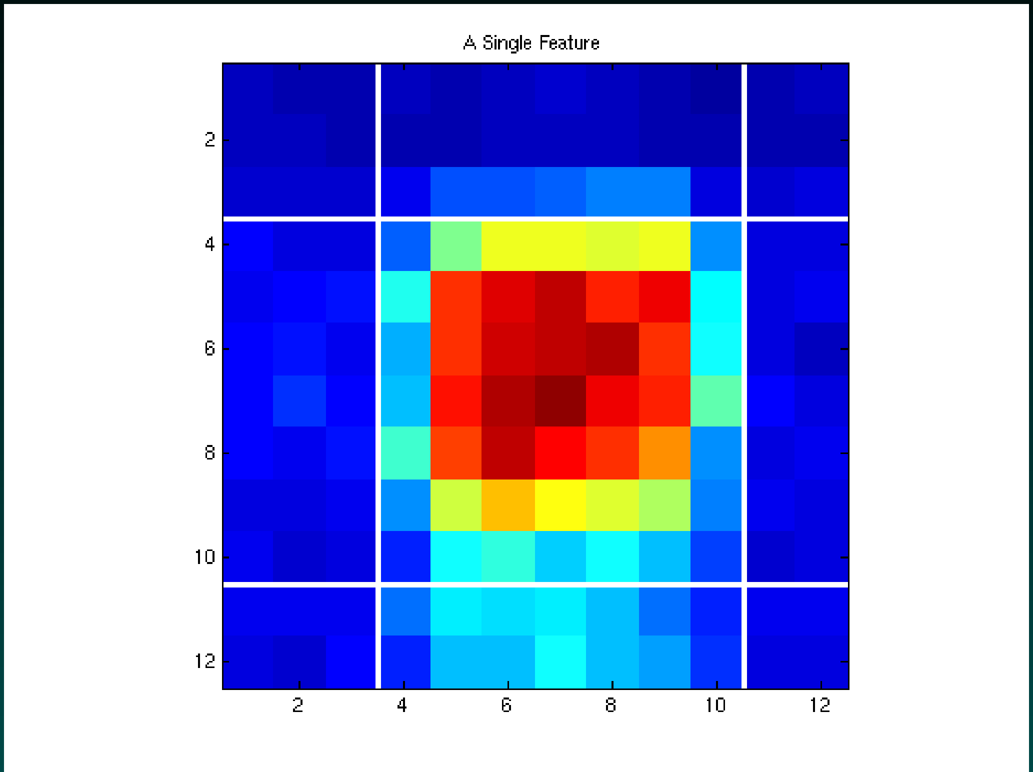
The pixelated features have been combined with positive controls to spell out the chip type – this helps ensure that the image is correctly oriented. Also note the border lattice of alternating dark and bright QC probes, making image alignment and feature detection easier.

GeneChip features



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

GeneChip features



Each feature on this chip is approximately 7 pixels (20 microns) on a side. In general, Affymetrix features use many fewer pixels than are used for the round spots in the images of other types of microarrays. Features are not perfectly uniform!

The DAT file size

The DAT file 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 image shown above involved a 4733 by 4733 grid of pixels, so the total file size is $2 * 4733^2 + 512 = 44803090$ bytes (45M).

This is BIG.

File size is a nontrivial issue with Affy data; the earlier versions of the software could only work with a limited number of chips (say 30).

Compression 1: The CEL file

Contains the feature quantifications.

```
[CEL]
Version=3
[HEADER]
Cols=640
Rows=640
TotalX=640
TotalY=640
OffsetX=0
OffsetY=0
GridCornerUL=219 235
GridCornerUR=4484 253
GridCornerLR=4469 4518
GridCornerLL=205 4501
Axis-invertX=0
AxisInvertY=0
```

```
swapXY=0
DatHeader=[0..19412]  U95Av2_CDDO_12_14_01: CLS=4
RWS=4733 XIN=3  YIN=3  VE=17 2.0 12/14/01 12:23:3
HG_U95Av2.1sq  6 Algorithm=Percentile
AlgorithmParameters=Percentile:75;CellMargin:2;
OutlierHigh:1.500;OutlierLow:1.004
[INTENSITY]
NumberCells=409600
CellHeader=X Y MEAN STDV NPIXELS
  0   0 133.0 16.6  25
  1   0 8150.0 1301.3  20
```

Cuts things down to about 12M here!

But we could do better...

Things learned

The X and Y fields aren't necessary; these can be inferred from position.

Keeping that 1 decimal place of accuracy doubles the storage space required and supplies effectively no information.

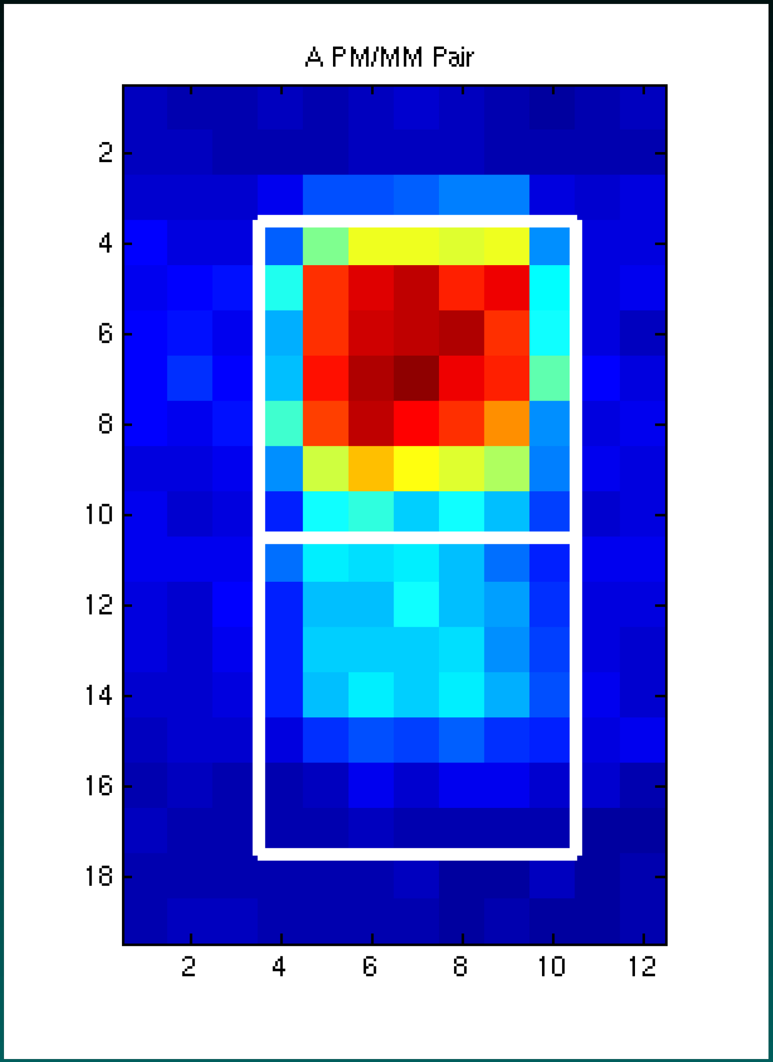
Most people don't use the STDV and NPIXELS fields.

The above description covered Affy's version 3.0 files.

Version 4.0, they shifted to a binary format, and each row got stored as a MEAN-STDV-NPIXEL or float-float-short triplet. Cut space, but not enough.

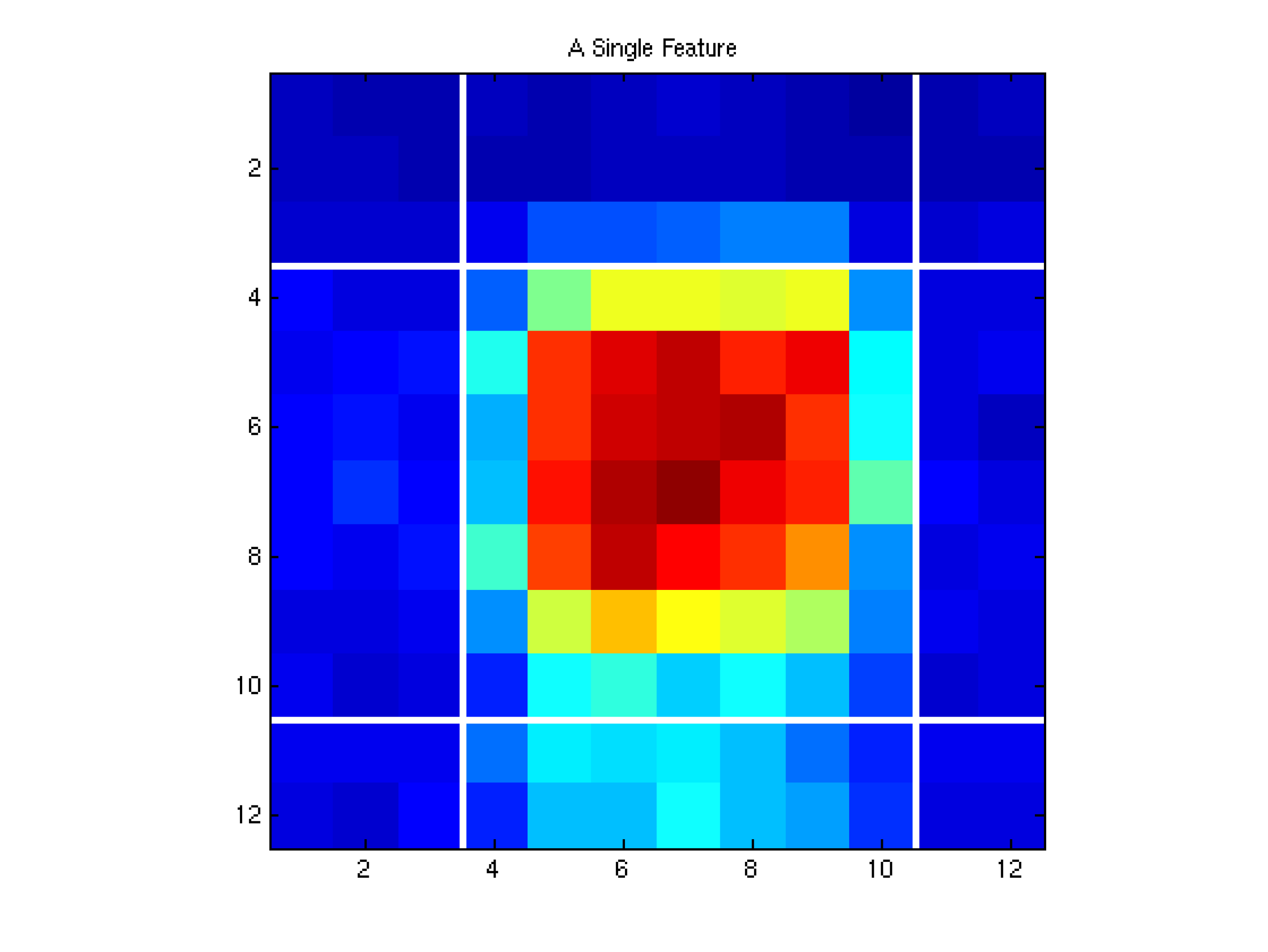
In late 05, Affy introduced the "Compact CEL" format, keeping just the MEAN as a short (no fraction).

DAT to CEL: Quantifying features



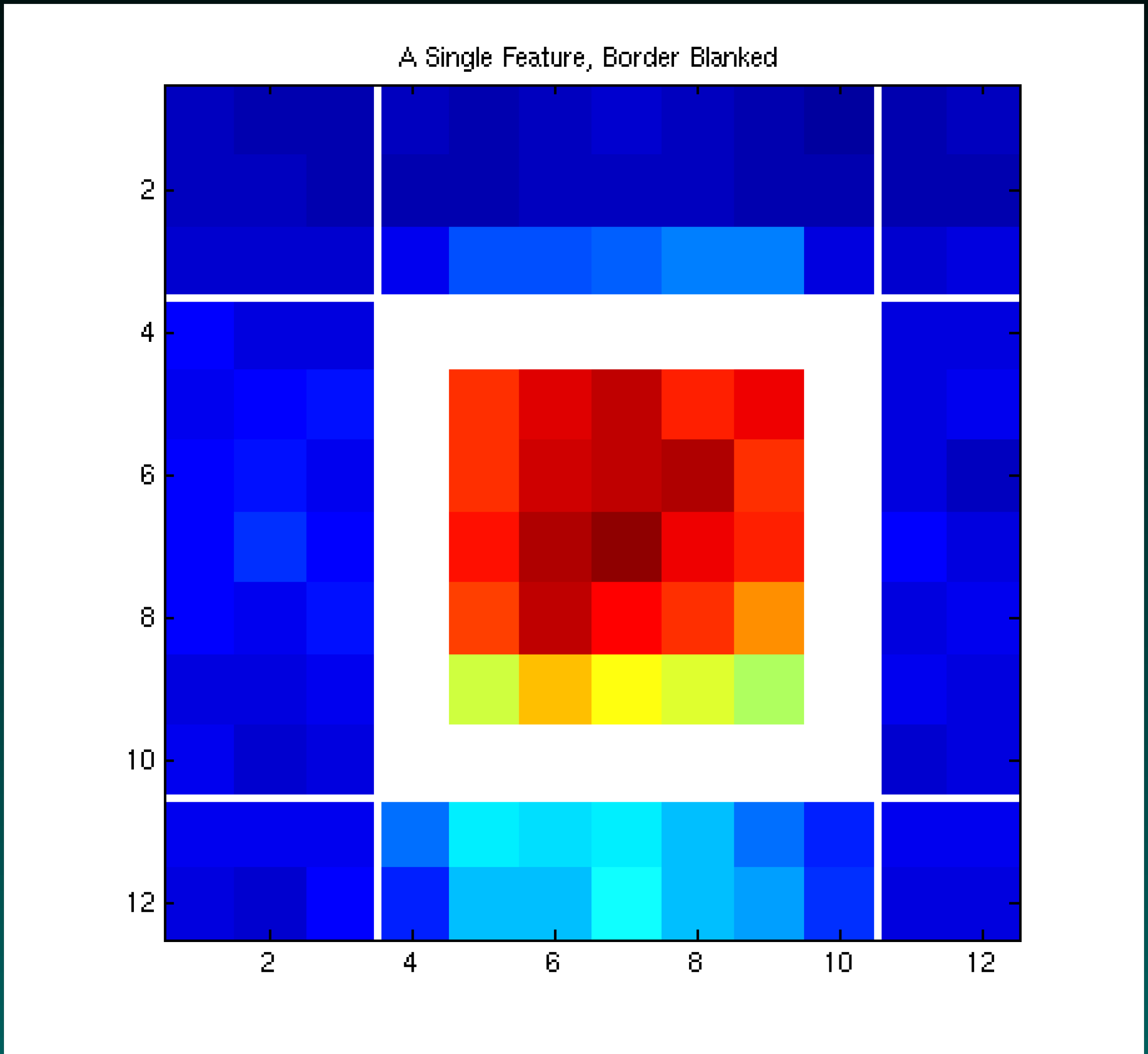
Locate a probe pair of interest

Zoom to a single feature



start with the pixel region

Trim the feature



trim off the outermost boundary

Get numbers

record the 75th percentile value of the stuff remaining.

Why trim?

Why the 75th percentile, and not the median? or the mean?

A shift in focus

the above problem, going from the image to the feature quantification, has been a major part of the discussion for quantification of other types of arrays, in part because we get one spot per gene.

Here, pretty much everybody uses Affy's algorithm. Not so much because it's perfect, as because it's reasonable.

The real challenge here comes from summarizing multiple measurements of the same thing.

Where the Probesets Are: The CDF file

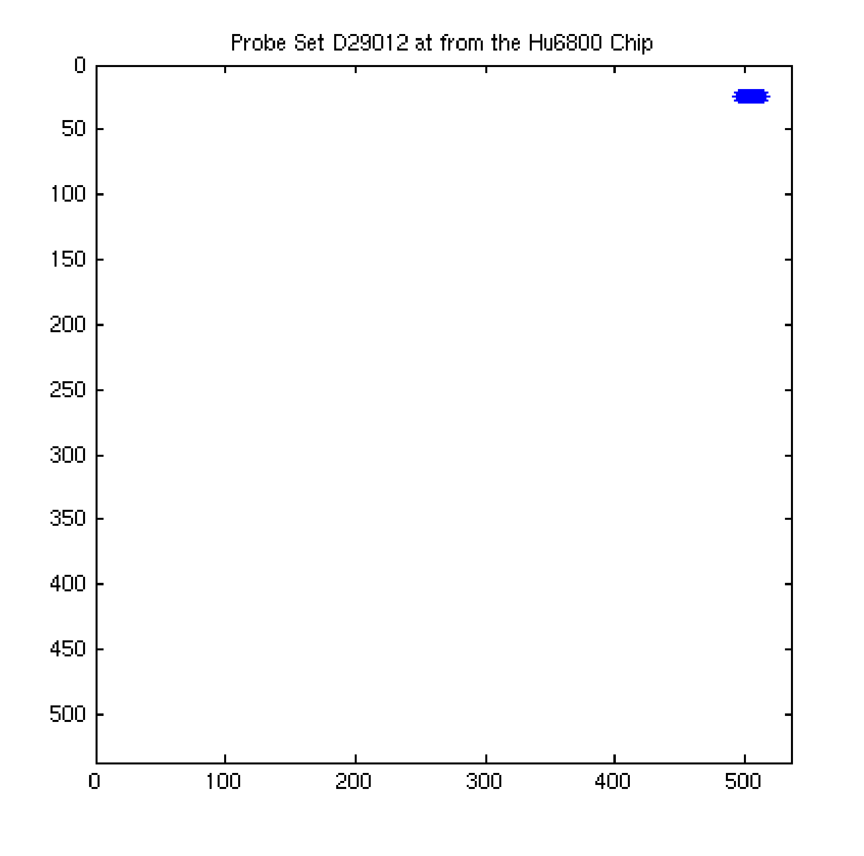
With any set of microarray experiments, one of the major challenges is keeping track of how the feature quantifications map back to information about genes, probes, and probe sets. There is one CDF file for each type of GeneChip, which contains this information.

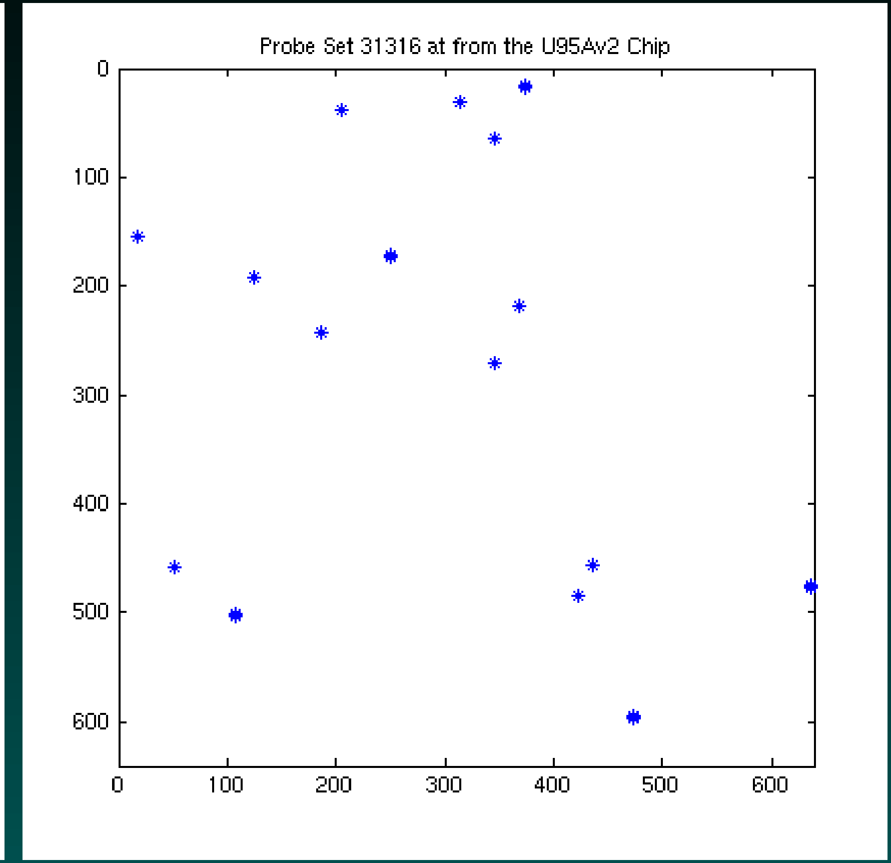
```
[CDF]
Version=GC3.0

[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
             CODONIND CODON   REGIONTYPE REGION
Cell1=517    568     N        control    31457_at   0
             13      A        A          A          0
             -1      -1       99
Cell2=517    567     N        control    31457_at   0
             13      A        T          A          0
             -1      -1       99
Cell3=78     343     N        control    31457_at   1
             13      T        A          T          1
```

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.

Quantifying a Dataset

For this, we need a set of data.

Fortunately for us, there's lots of Affy data on the web. Today, we'll be using some data from Todd Golub's lab on Leukemia differentiation.

http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi

HL60_undiff_PMA_ATRA_CELfiles.tar 21 CEL files, Hu6800 chips (aka HuGeneFL).

(CEL files from the web; we have the CDF files here).

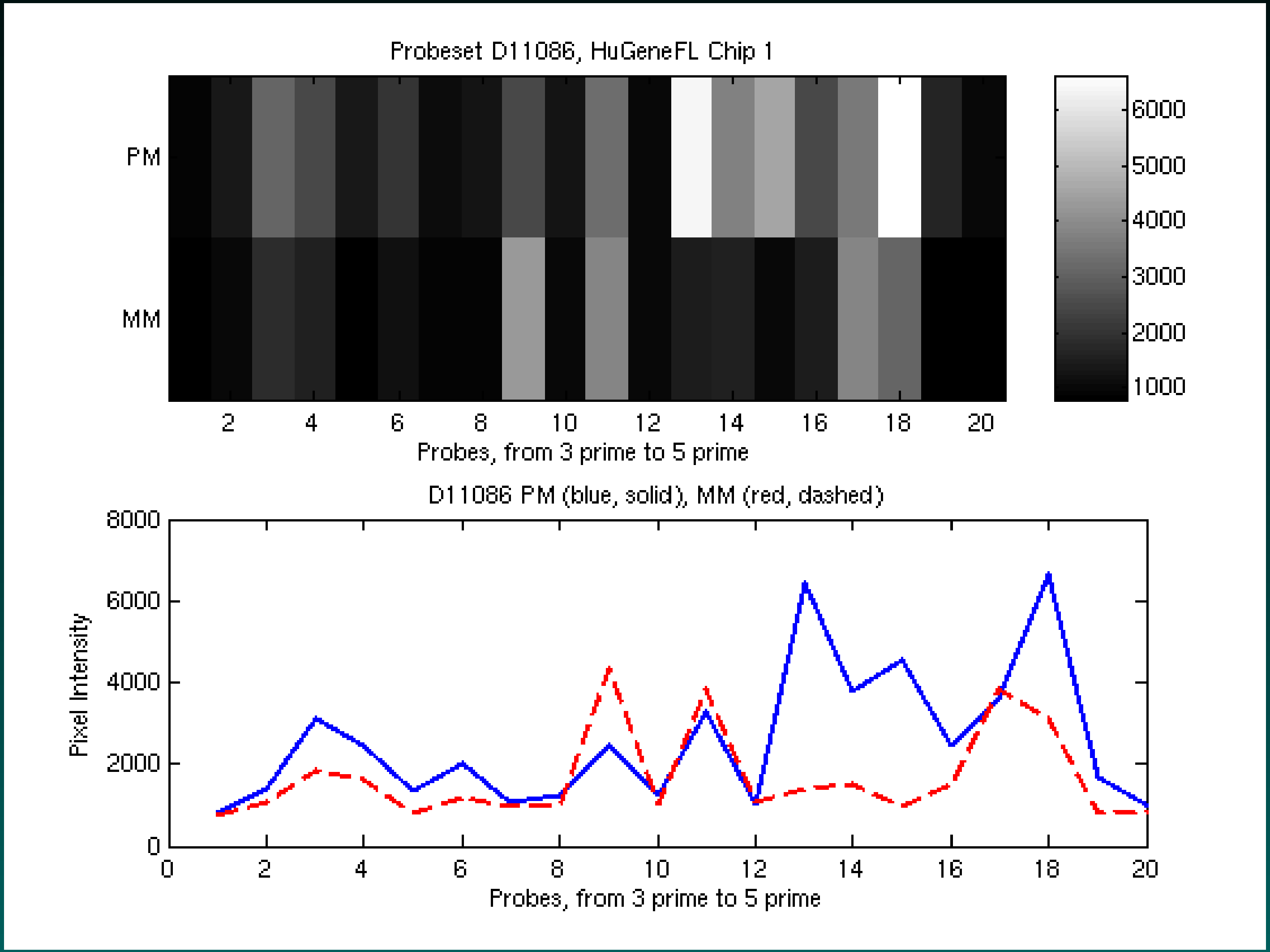
Normalization

Before we quantify individual probesets, we need to check whether the image data is roughly comparable in intensity. Adding twice as much sample may make the resultant image brighter, but it doesn't tell us anything new about the underlying biology.

In most microarray experiments, we are comparing samples of a single tissue type (eg brain), and in such cases we assume that "most genes don't change".

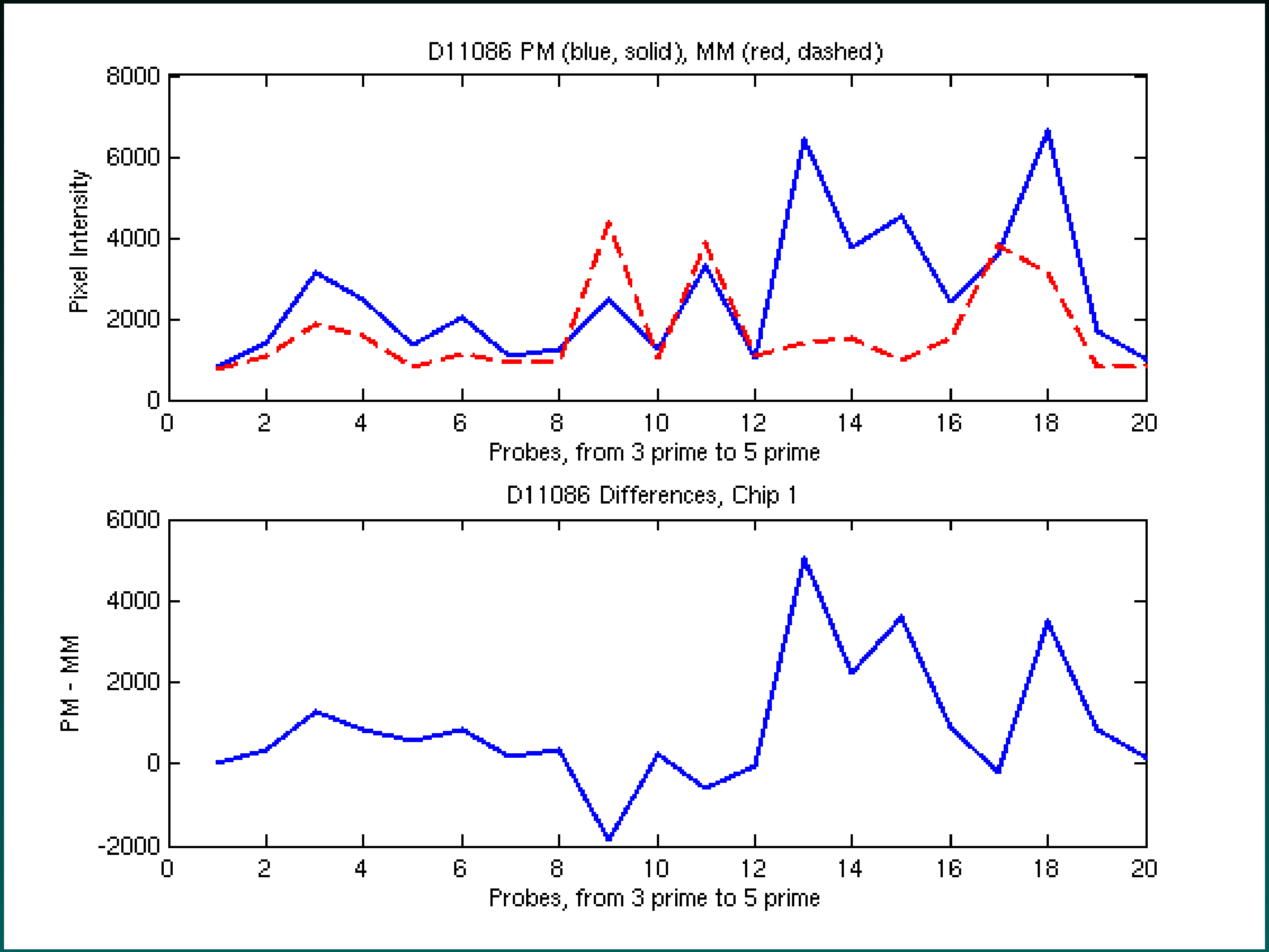
Typically, we enforce this by matching quantiles of the feature intensity distributions.

Probeset D11086_at, chip 1



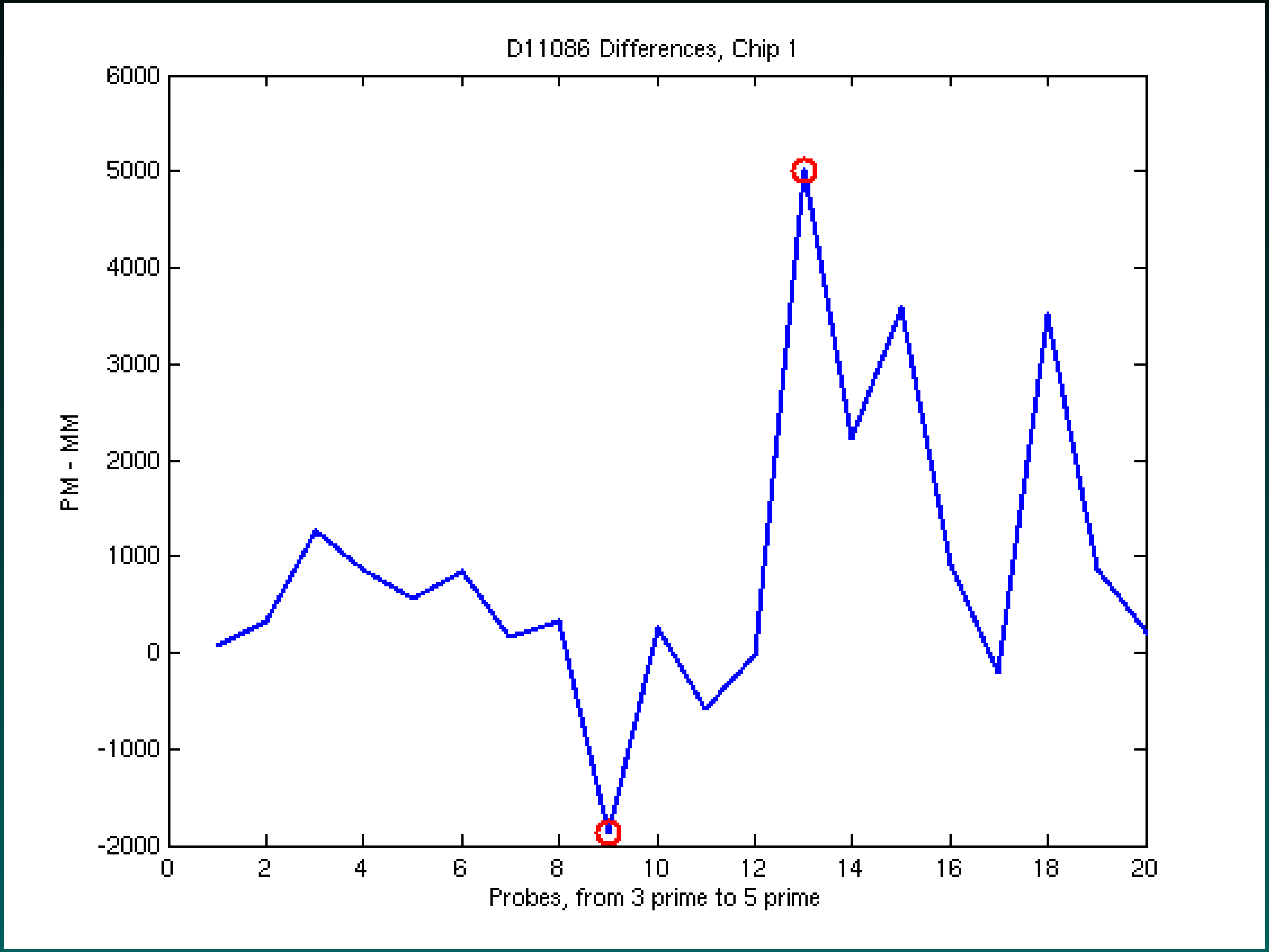
So, how do we summarize this?

In the beginning: MAS 4.0, aka AvDiff

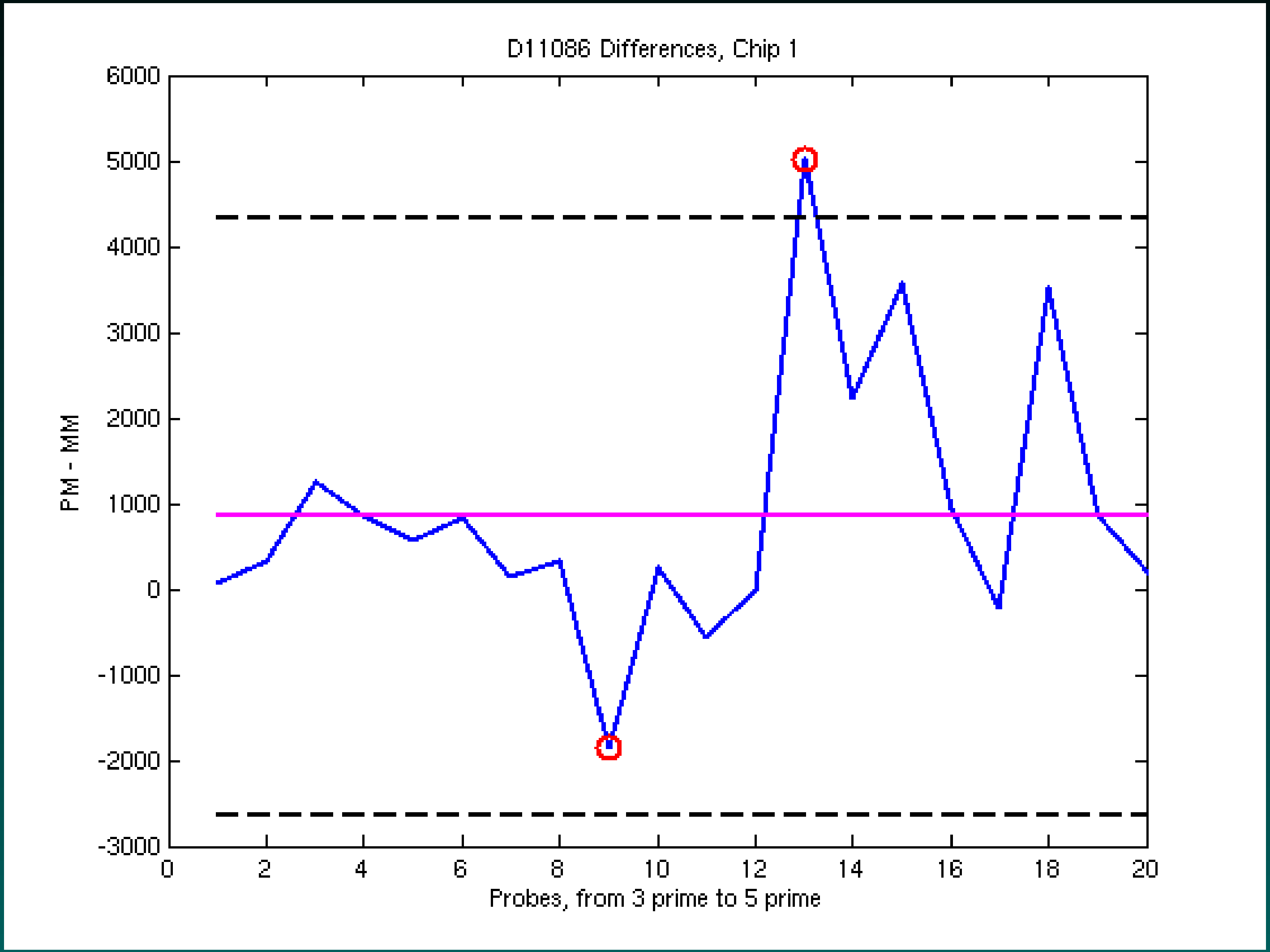


First, shift to PM-MM differences. (Cross-hyb?)

AvDiff Processing 1: Flag extremes

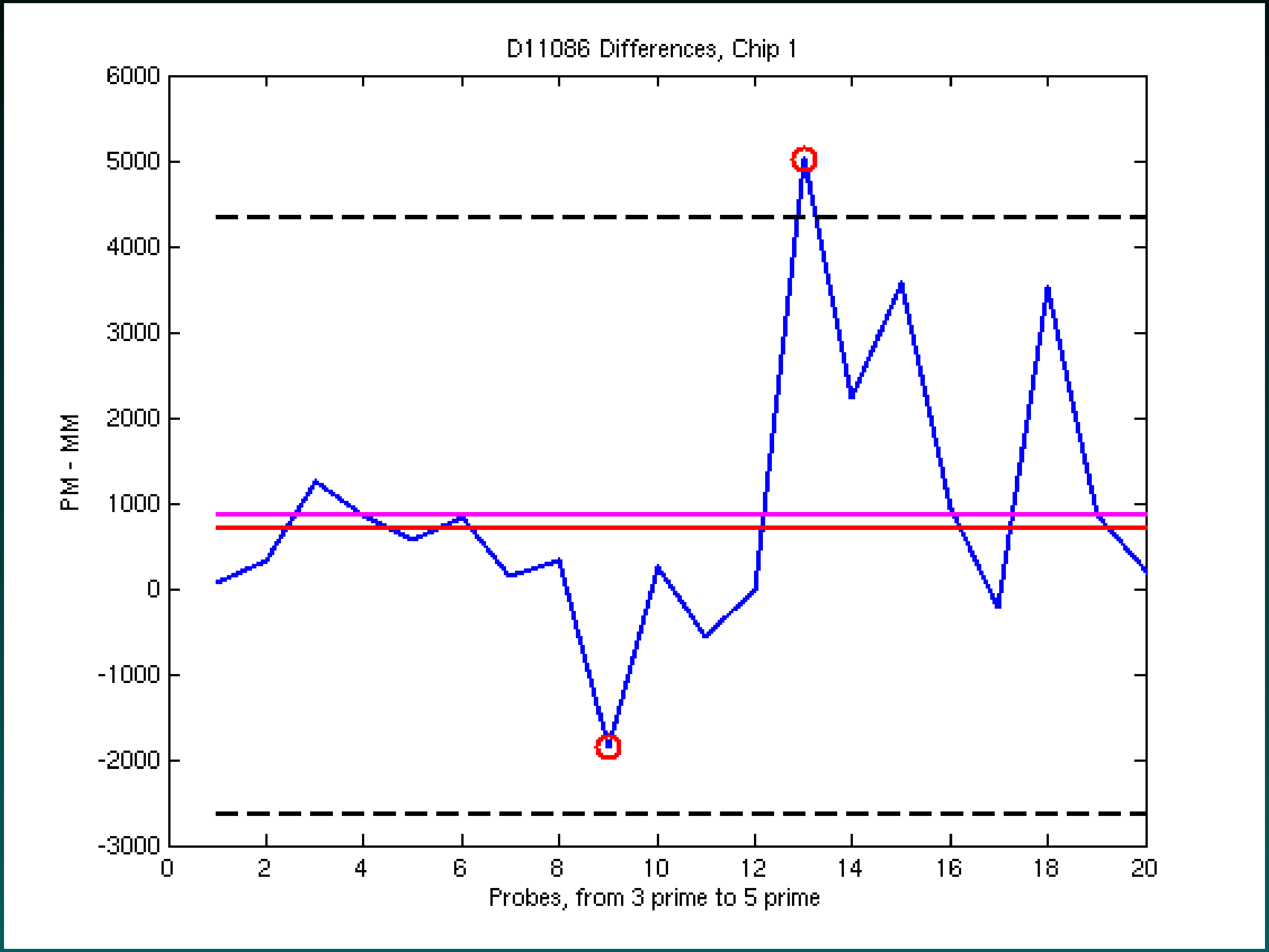


AvDiff Processing 2: Define “ok” Bounds



mean ± 3*sd, computed omitting extremes

AvDiff Processing 3: Average “ok” Diffs



Not quite what we had before!

Comments on AvDiff?

It does combine measurements across probes, and tries to exploit redundancy.

Comments on AvDiff?

It does combine measurements across probes, and tries to exploit redundancy.

It weights all probes equally.

Comments on AvDiff?

It does combine measurements across probes, and tries to exploit redundancy.

It weights all probes equally.

It works on the PM-MM differences in an additive fashion.

Comments on AvDiff?

It does combine measurements across probes, and tries to exploit redundancy.

It weights all probes equally.

It works on the PM-MM differences in an additive fashion.

It can give negative values.

Comments on AvDiff?

It does combine measurements across probes, and tries to exploit redundancy.

It weights all probes equally.

It works on the PM-MM differences in an additive fashion.

It can give negative values.

It can omit interesting probes.

Comments on AvDiff?

It does combine measurements across probes, and tries to exploit redundancy.

It weights all probes equally.

It works on the PM-MM differences in an additive fashion.

It can give negative values.

It can omit interesting probes.

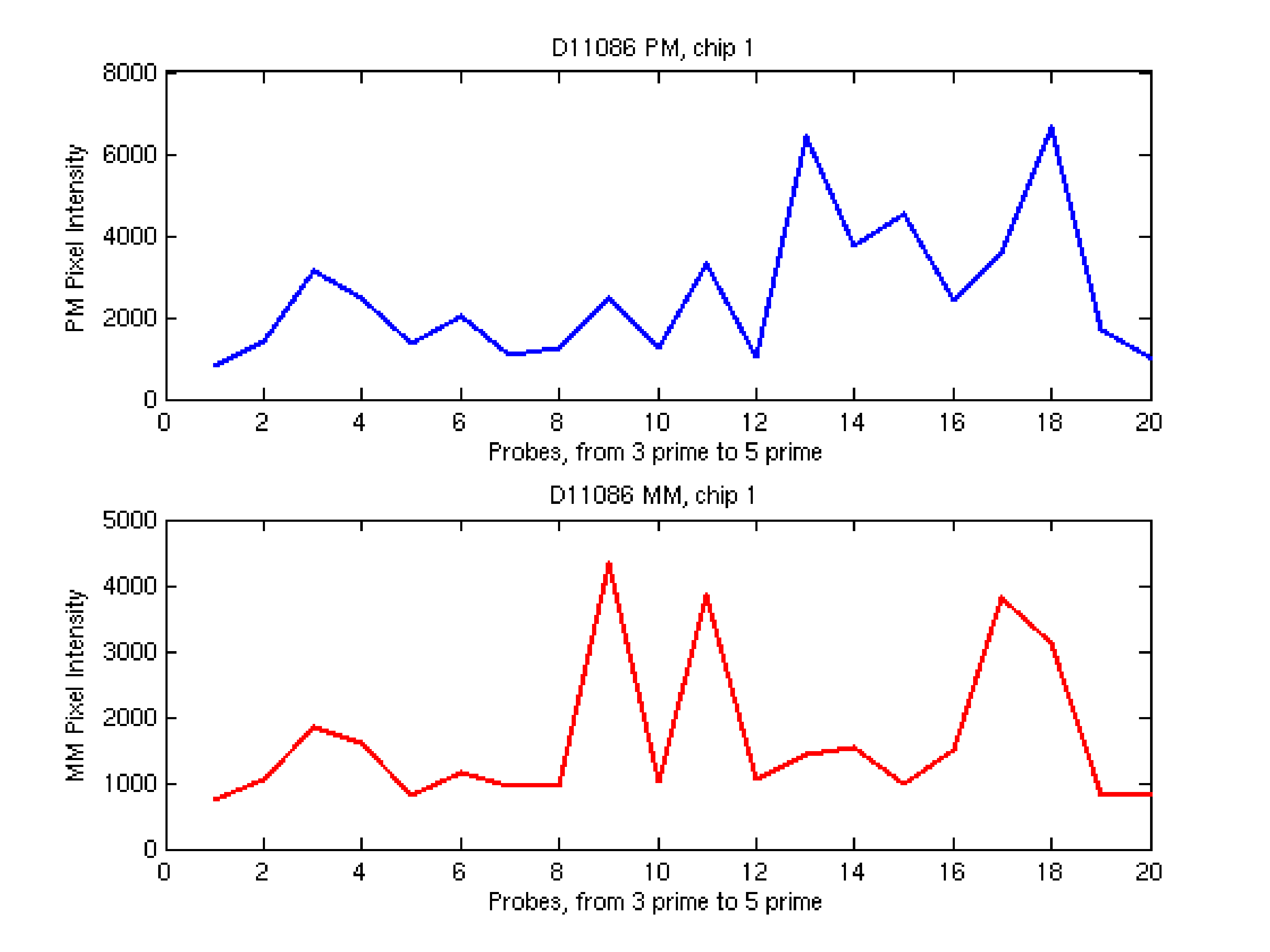
It works one chip at a time. It does not learn.

Using models: dChip

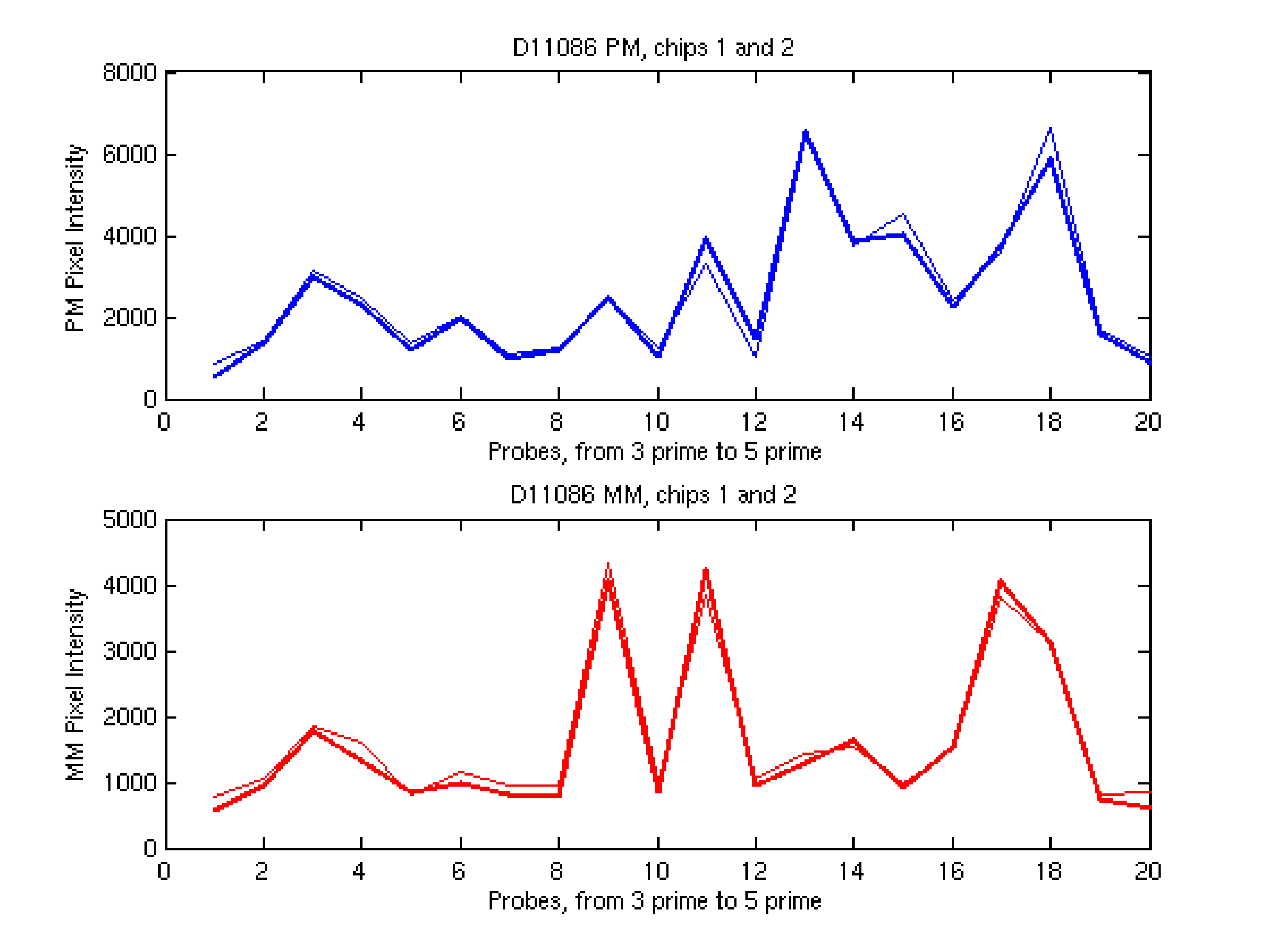
In 2001, Cheng Li and Wing Wong introduced a new method of summarizing probeset intensities, “model-based expression indices”, or MBEI. (PNAS, v.98, p.31-36).

At the crux of their argument was a very simple observation – the relative expression values of probes within a probeset were very stable across multiple arrays.

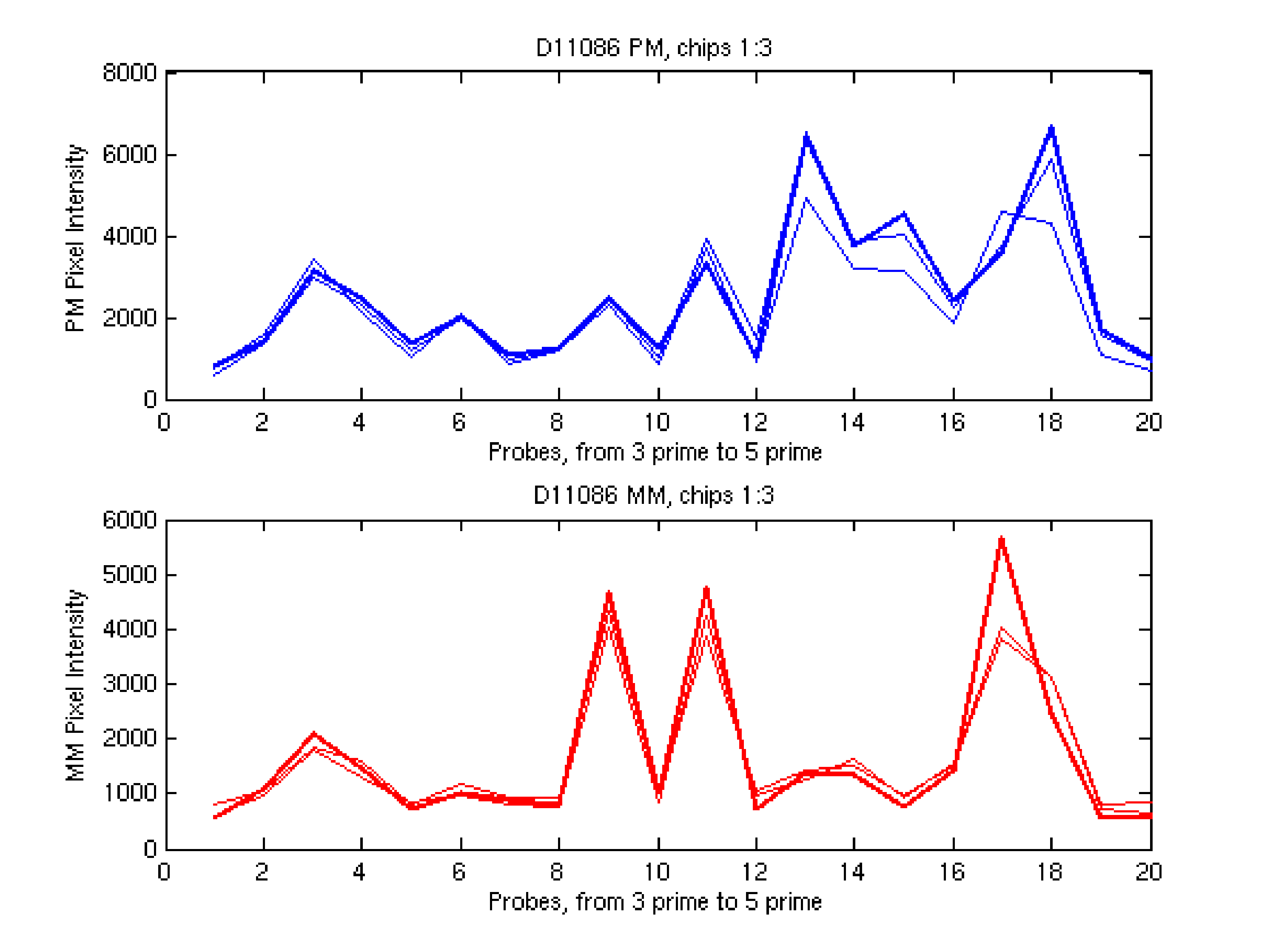
Stability: Our First Chip



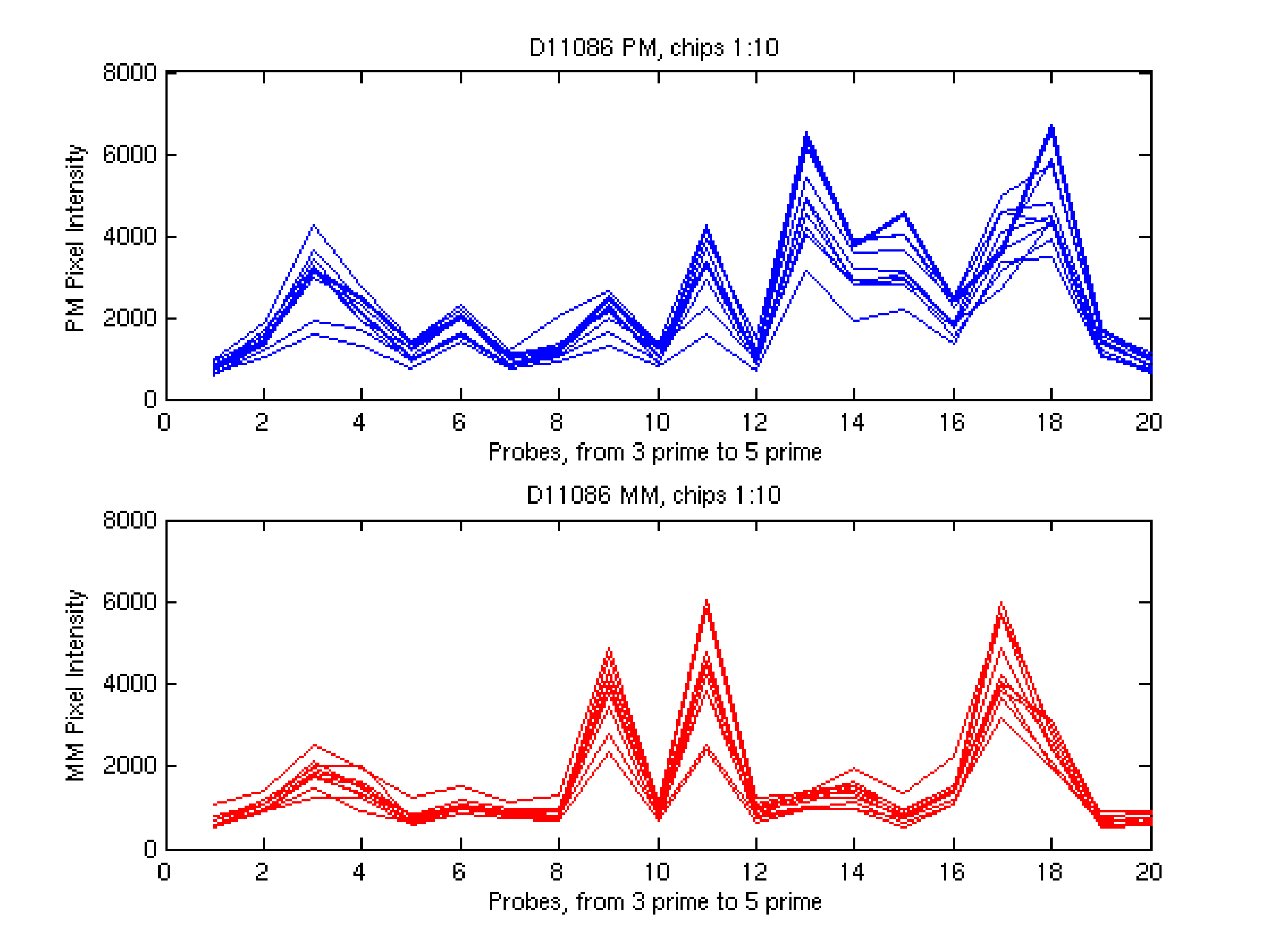
Stability: Two Chips



Stability: Three Chips



Stability: Ten Chips



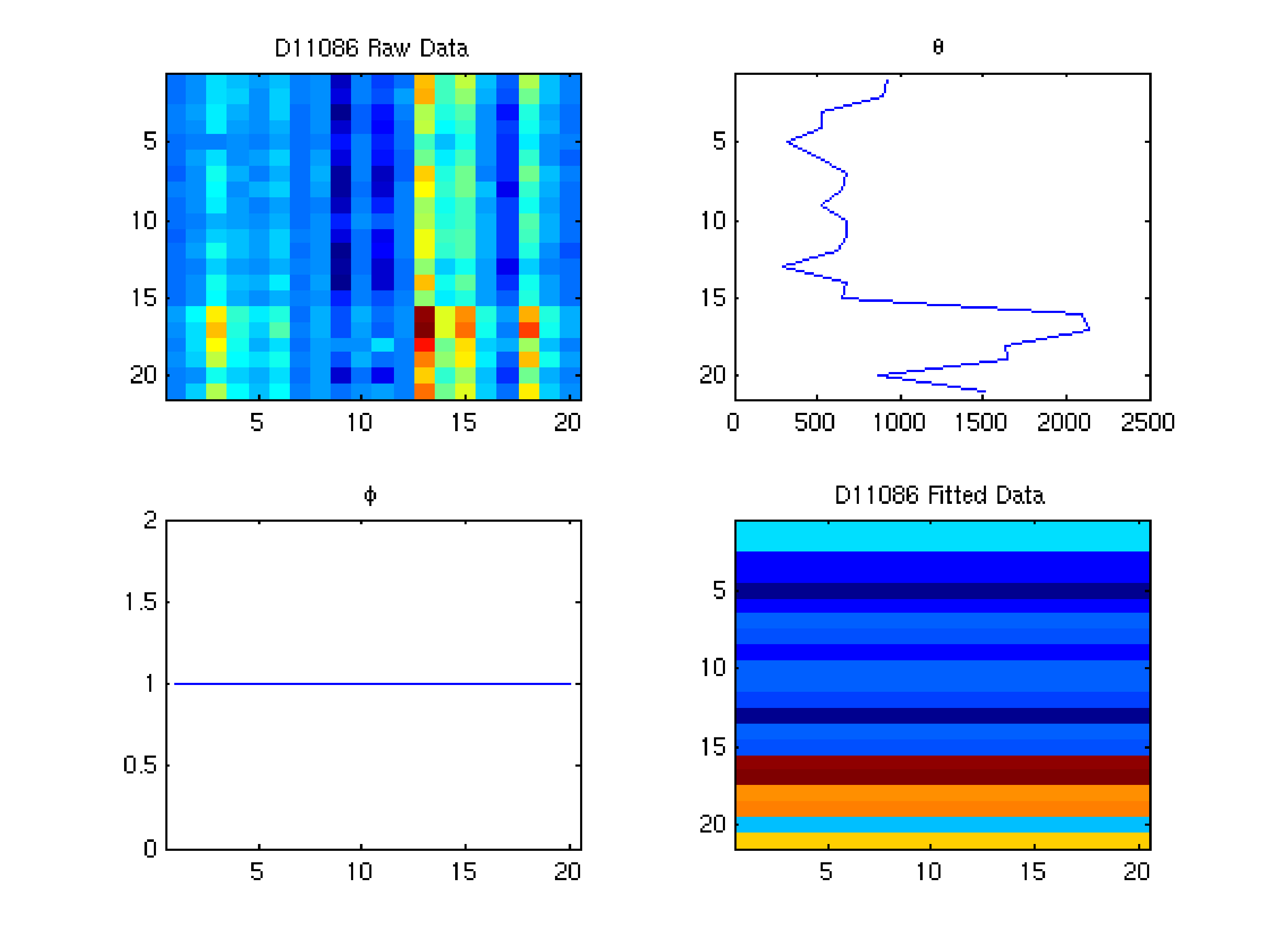
So, how to exploit this?

Fit a model: For sample i, and probe pair j, they posit that

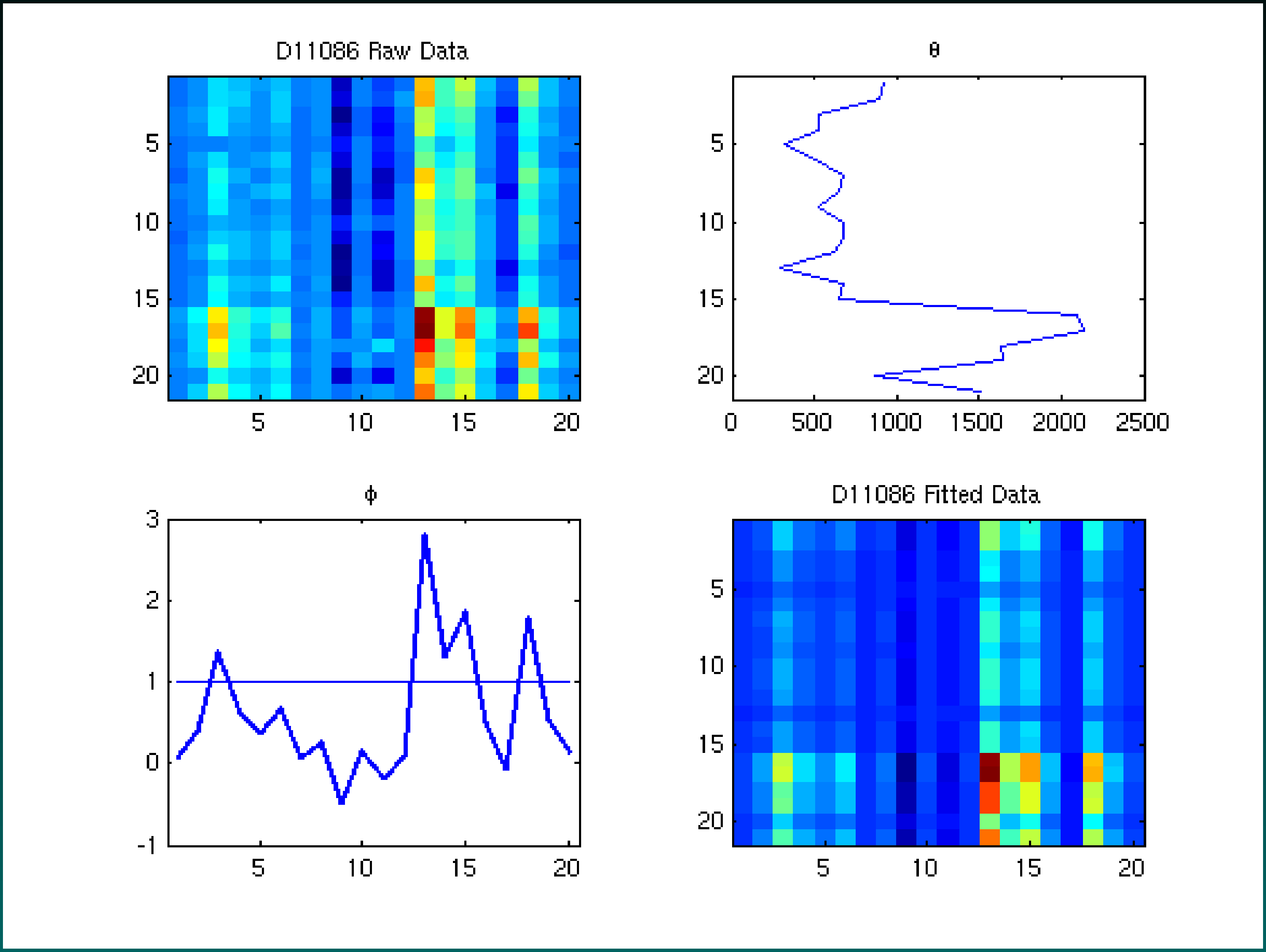
$$\begin{aligned} PM_{ij} &= \nu_j + \theta_i \alpha_j + \theta_i \phi_j + \epsilon \\ MM_{ij} &= \nu_j + \theta_i \alpha_j + \epsilon \end{aligned}$$

Focusing on the PM-MM differences, this model condenses to one with two sets of unknowns: θ_i and ϕ_j.

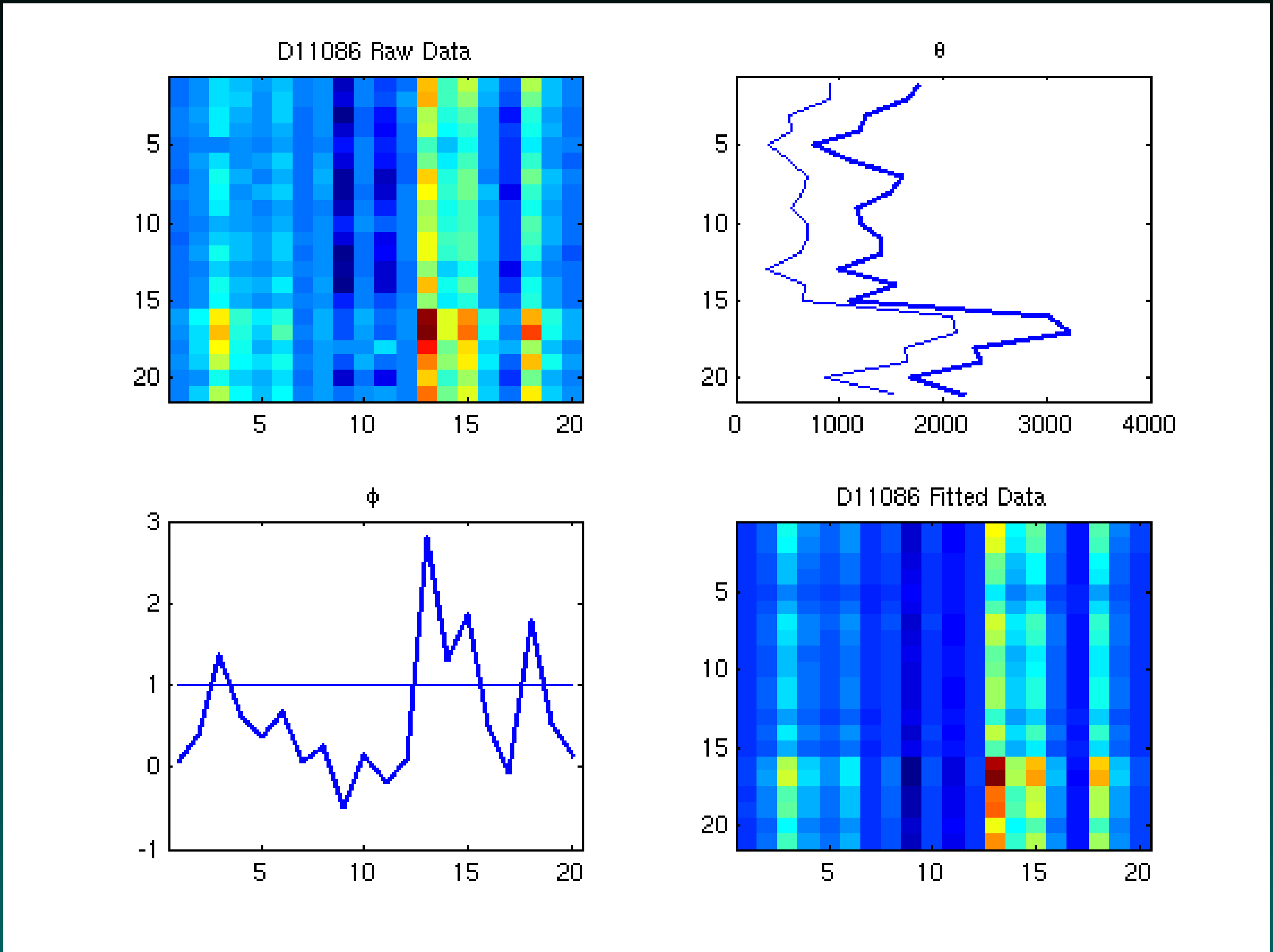
Fit using several chips at once!



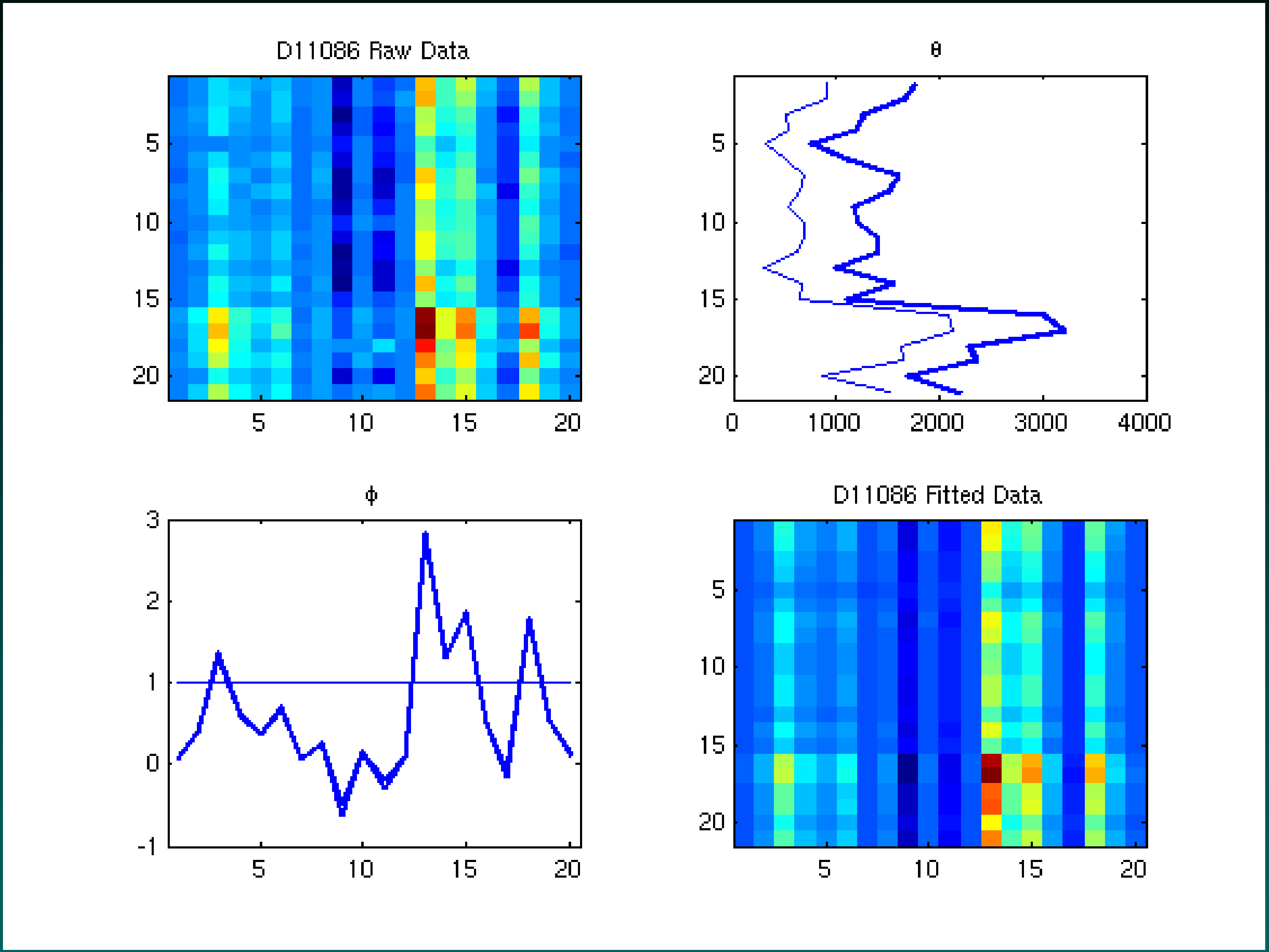
The next step: ϕ



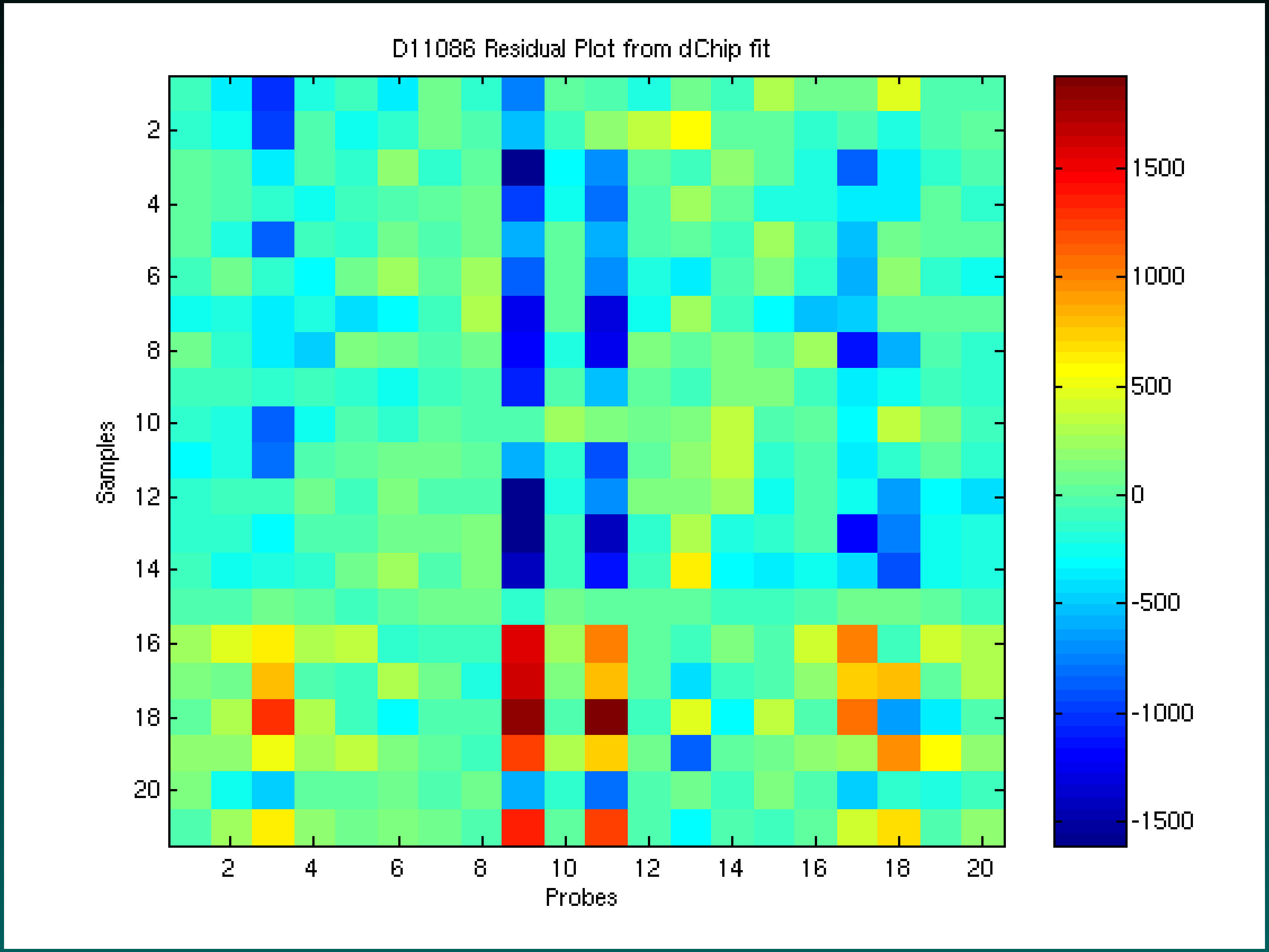
and back to θ



and after 5 of each



What do the residuals look like?



Note potential outlying probes!

Comments on dChip?

By using multiple chips, it can keep all of the probes; no tossing of the most informative ones.

Comments on dChip?

By using multiple chips, it can keep all of the probes; no tossing of the most informative ones.

It captures effects that are multiplicative.

Comments on dChip?

By using multiple chips, it can keep all of the probes; no tossing of the most informative ones.

It captures effects that are multiplicative.

By checking the residuals from the model, it is possible to identify outliers due to artifacts (that replication idea again).

Comments on dChip?

By using multiple chips, it can keep all of the probes; no tossing of the most informative ones.

It captures effects that are multiplicative.

By checking the residuals from the model, it is possible to identify outliers due to artifacts (that replication idea again).

Using the hypothesized error model, confidence bands for the fold change can be computed.

Comments on dChip?

By using multiple chips, it can keep all of the probes; no tossing of the most informative ones.

It captures effects that are multiplicative.

By checking the residuals from the model, it is possible to identify outliers due to artifacts (that replication idea again).

Using the hypothesized error model, confidence bands for the fold change can be computed.

Probe profiles can be computed in one experiment and used in another.

Comments on dChip?

By using multiple chips, it can keep all of the probes; no tossing of the most informative ones.

It captures effects that are multiplicative.

By checking the residuals from the model, it is possible to identify outliers due to artifacts (that replication idea again).

Using the hypothesized error model, confidence bands for the fold change can be computed.

Probe profiles can be computed in one experiment and used in another.

The downside(?)s

dChip requires several chips to get a good fit for its model. It is not a good idea to trust the fits too much if they are based on just one or two chips.

I'm not convinced that this is altogether a bad thing.

The error model is too simplistic – larger intensity probes will typically also have larger variances.

Why did dChip catch on?

The model works pretty well.

Why did dChip catch on?

The model works pretty well.

The software package was (and remains) easy to acquire, learn and use. It incorporates several of the most common tricks that people want to play with array data.

It could handle large numbers of chips. This last is in part due to the fact that their file structures are better – a version 3.0 Affy CEL file for a U95Av2 chip takes about 12M to store, but the dChip summary takes about 1.6M.

Is dChip the Best Model?

Probably not.

Others out there include

MAS5.0

$$\mathrm{Tukey\ Biweight}(\log(\mathrm{PM_j} - \mathrm{CT_j}))$$

RMA

$$log(PM_{ij} - BG) = \mu_i + \alpha_j + \sigma\epsilon_{ij}$$

PDNN

all of these are implemented in R in BioConductor.

How do we know which models are better?

Well, after MAS5.0, Affy decided it wasn't going to fight to have the best algorithm; it would let others play that game. Indeed, it could reap the benefits of better algorithms by selling more chips.

To let people test their own models, they posted a test dataset: The Affy Latin Square Experiment.

so, why dChip?

dChip is more user-friendly than R.

The supporting data structures and annotation files are well-organized.

We can teach the basics quickly...