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

September 9 2004

Lecture 4: Quantifying Affy Chips

- DAT to CEL
- MAS 4.0 = AvDiff
- dChip
- MAS 5.0
- RMA
- PDNN

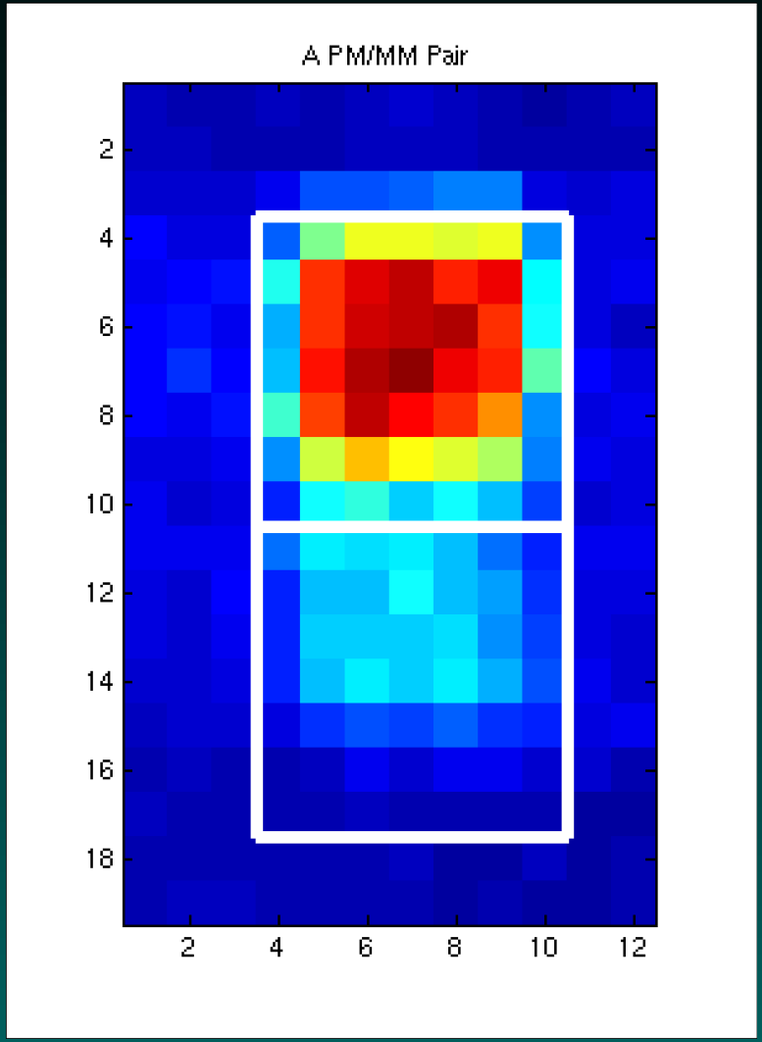
Starting with the DAT file...

A DAT file consists of a 512 byte header followed by the 16-bit pixel values: these values are arranged in a 4733 by 4733 square grid.

Features in the dat file are printed in the central portion of the chip in a grid arrangement. Each feature is typically several pixels in each dimension.

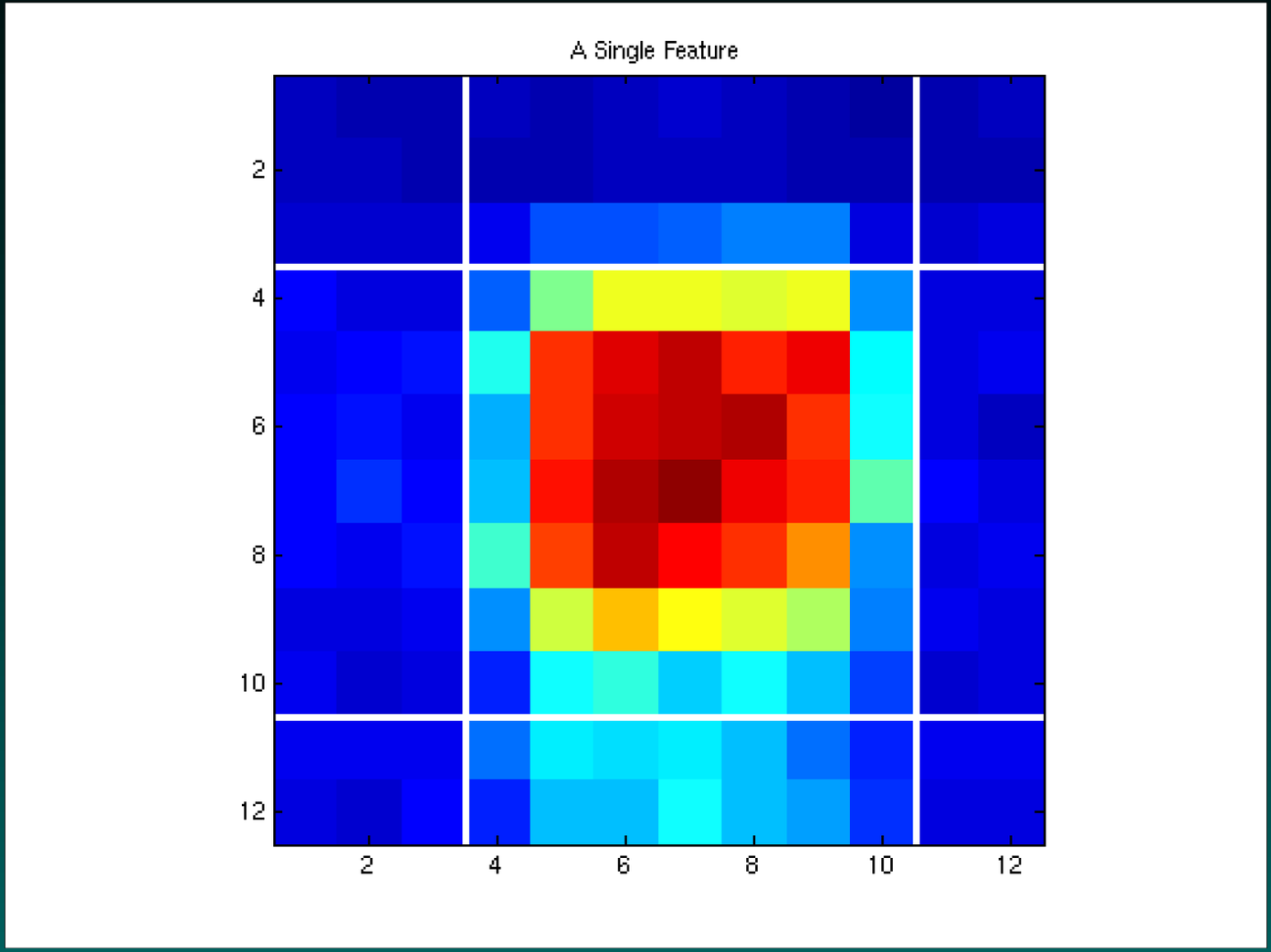
Affy automatically finds the corners of this grid in the main image, and partitions things from there.

DAT to CEL: Quantifying features



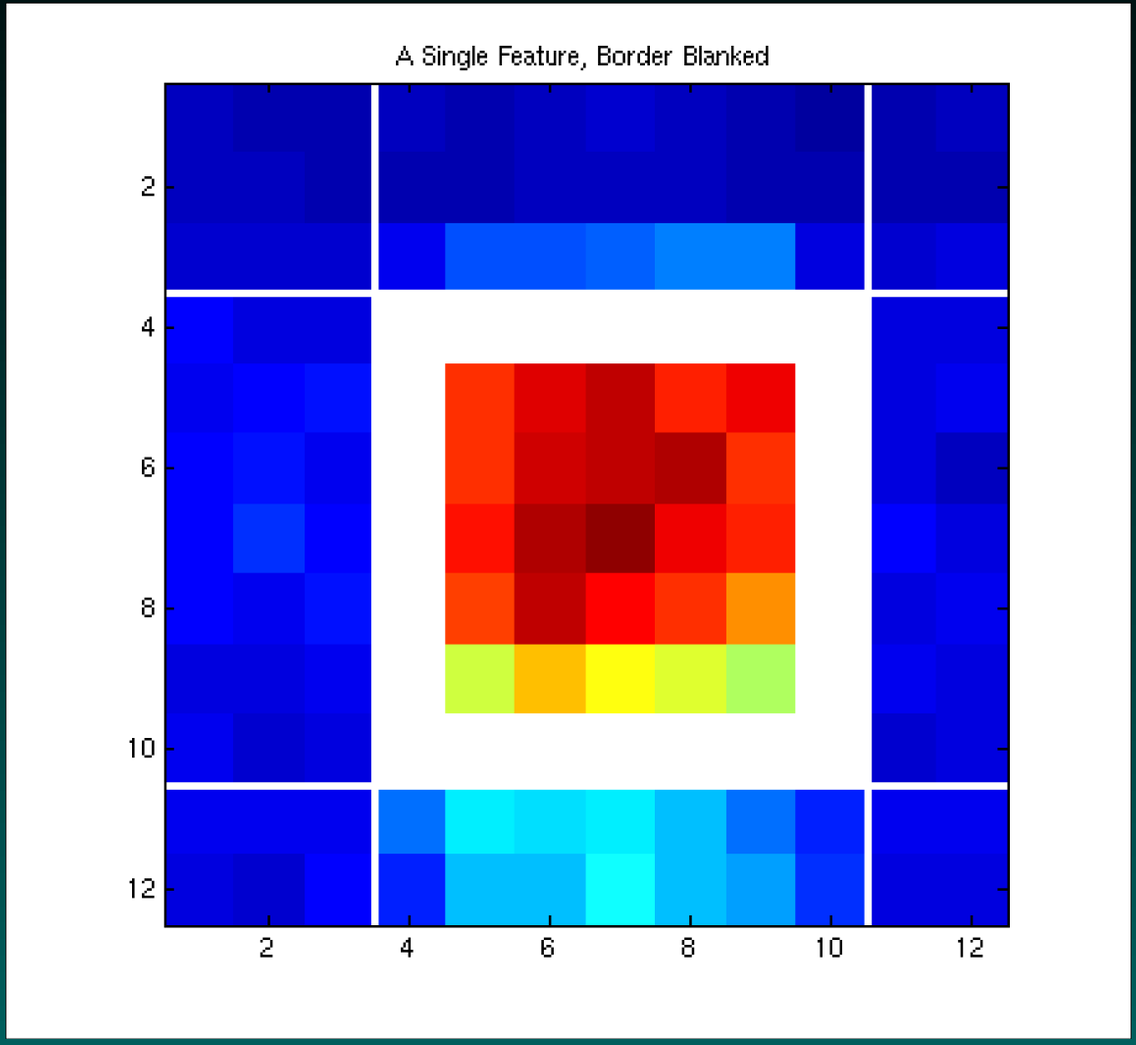
Locate a probe pair of interest

Zoom to a single feature



start with the pixel region

Trim the feature



trim off the outermost boundary

Final steps

record the 75th percentile value of the stuff remaining.

Why trim?

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

A shift in focus

the above problem, going from the image to the feature quantification, largely dominated are discussion of quantification for cDNA arrays.

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.

Of course, for this we need data.

Checking a probeset

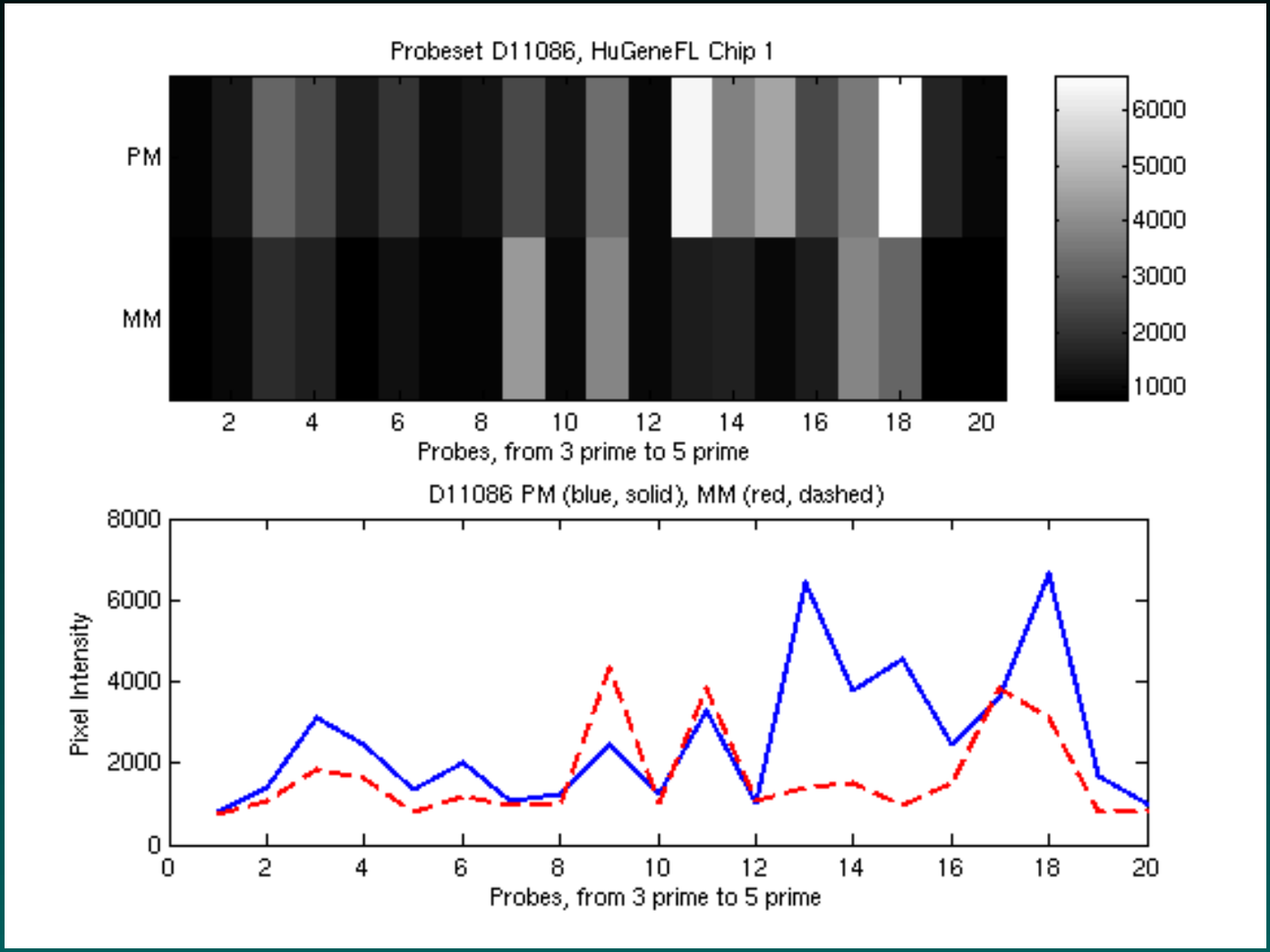
Fortunately for us, there's lots of Affy data on the web. Today, we'll be using some data from Todd Golub's lab on Leukemia differentiation.

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

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

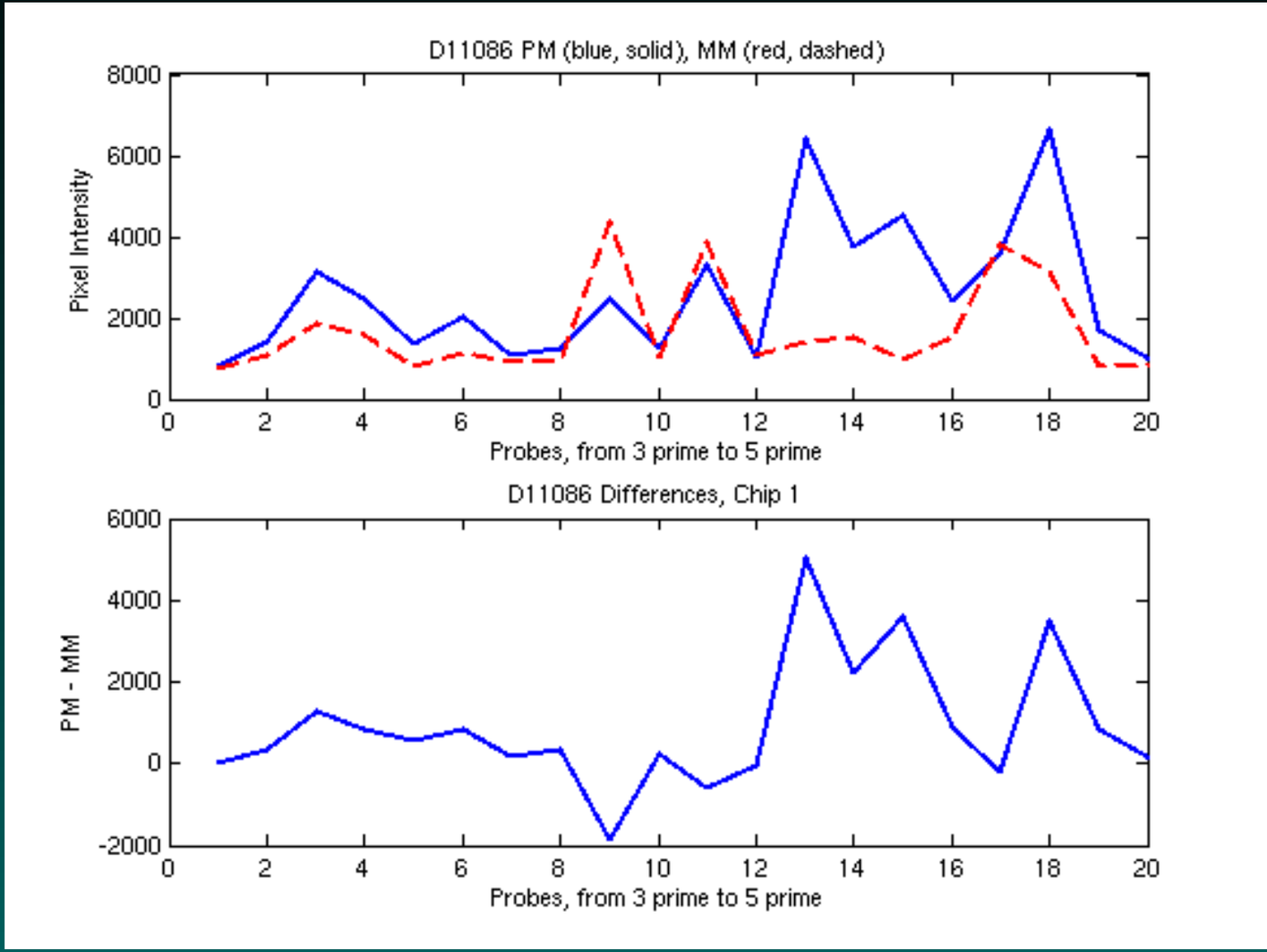
We'll look at probeset D11086_at.

Probeset D11086_at, chip 1



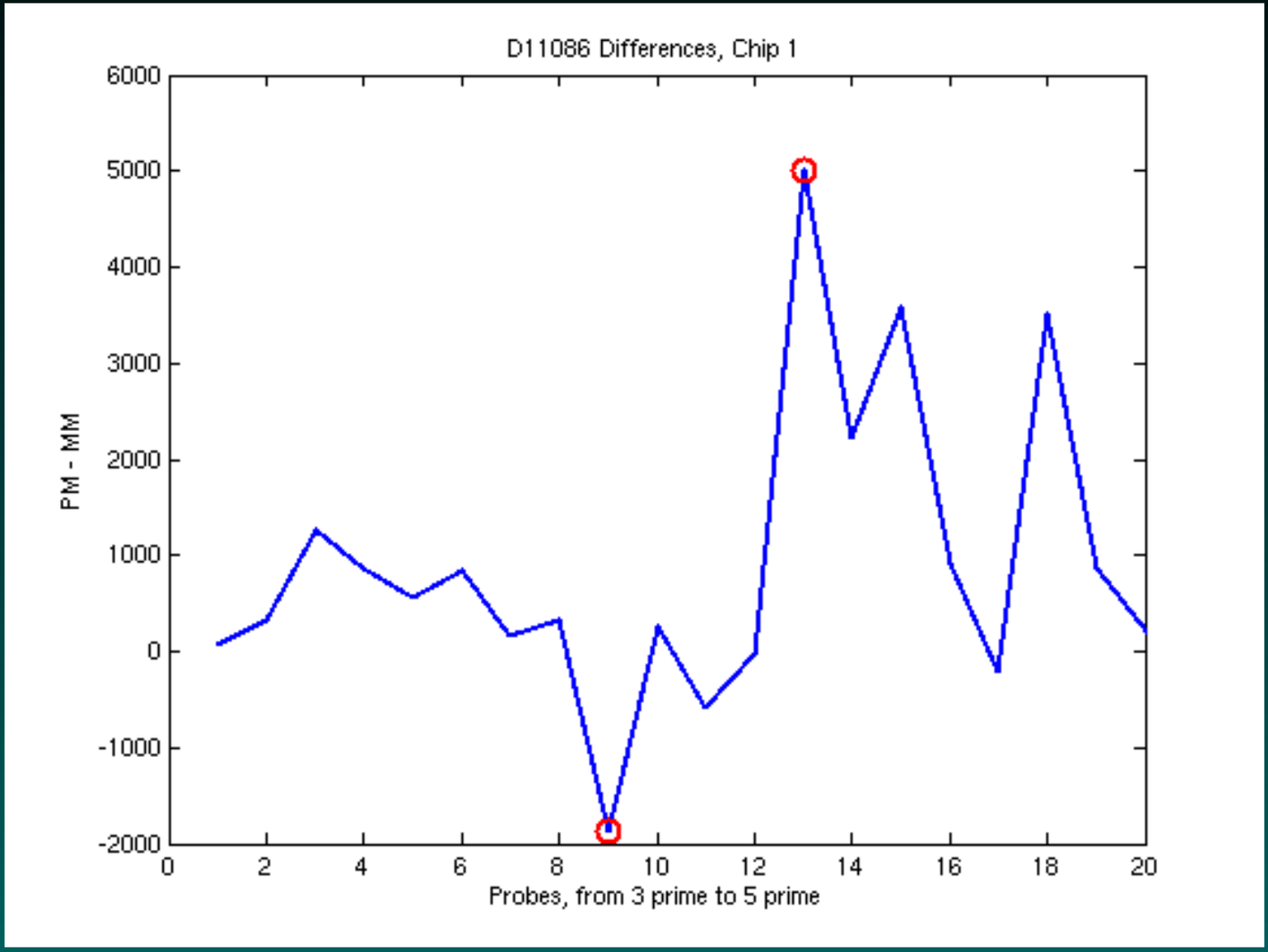
So, how do we summarize this?

In the beginning: MAS 4.0, aka AvDiff

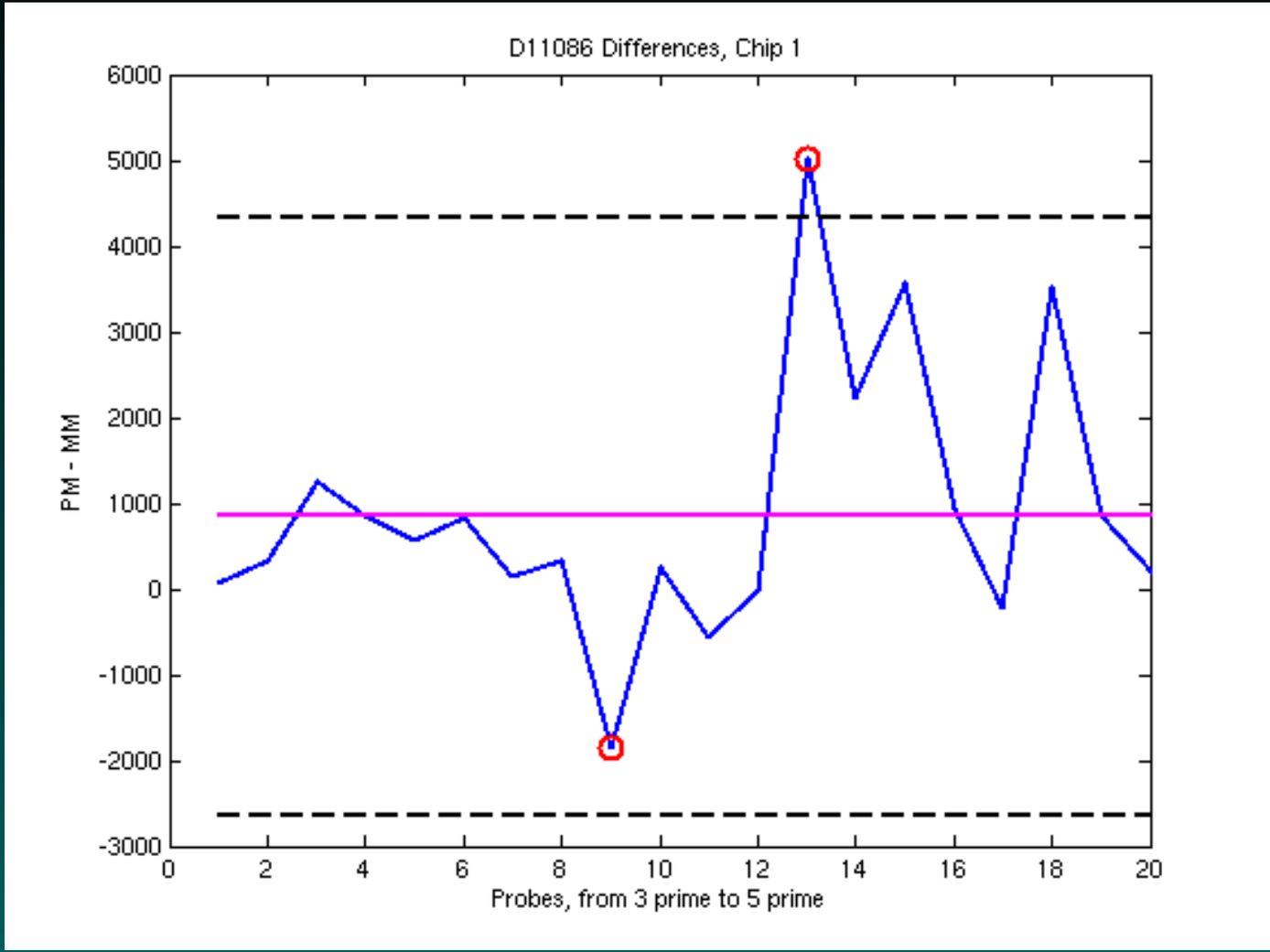


First, shift to PM-MM differences. (Cross-hyb?)

AvDiff Processing 1: Flag extremes

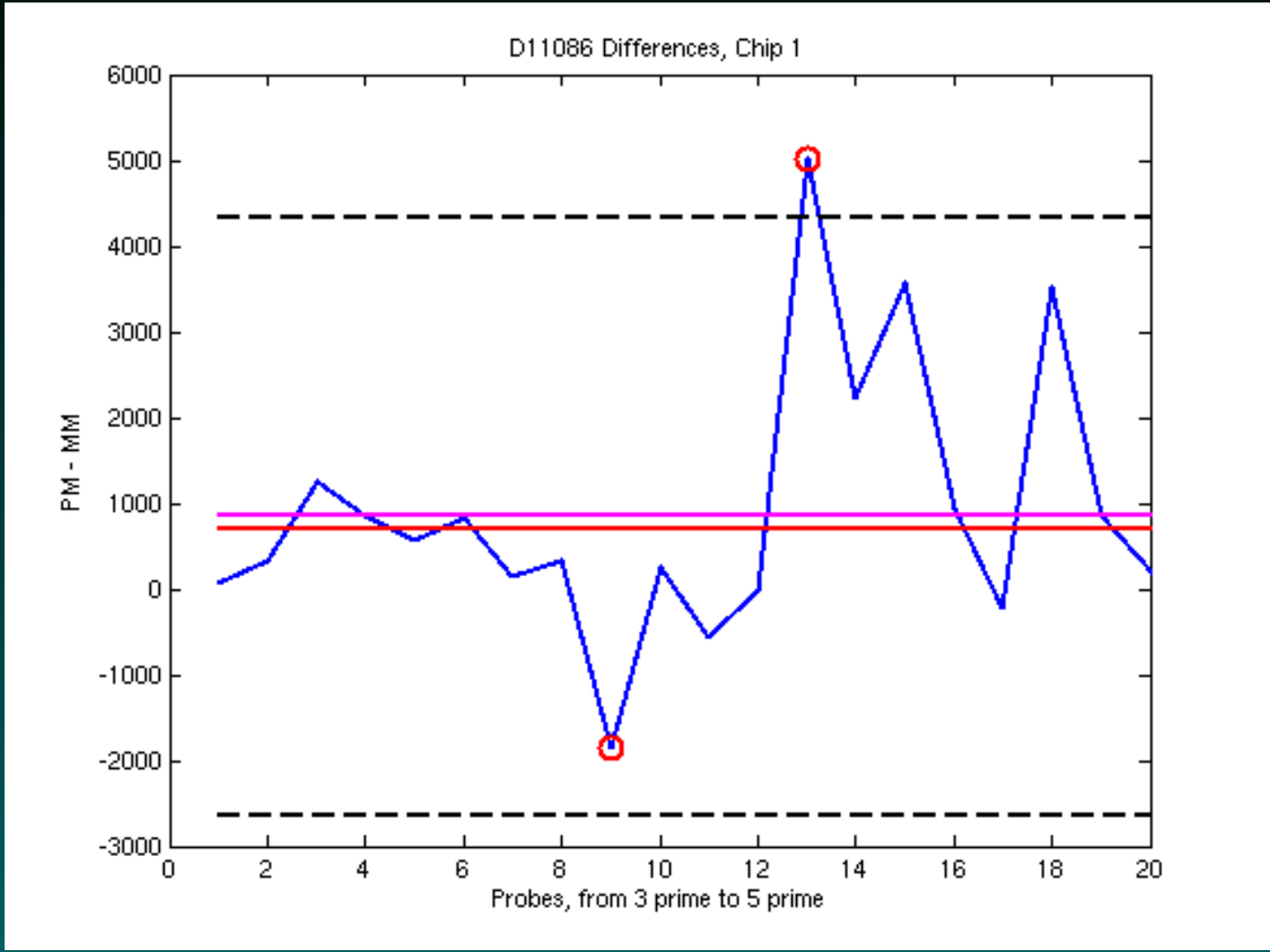


AvDiff Processing 2: Define "ok" Bounds



mean $\pm$ 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.

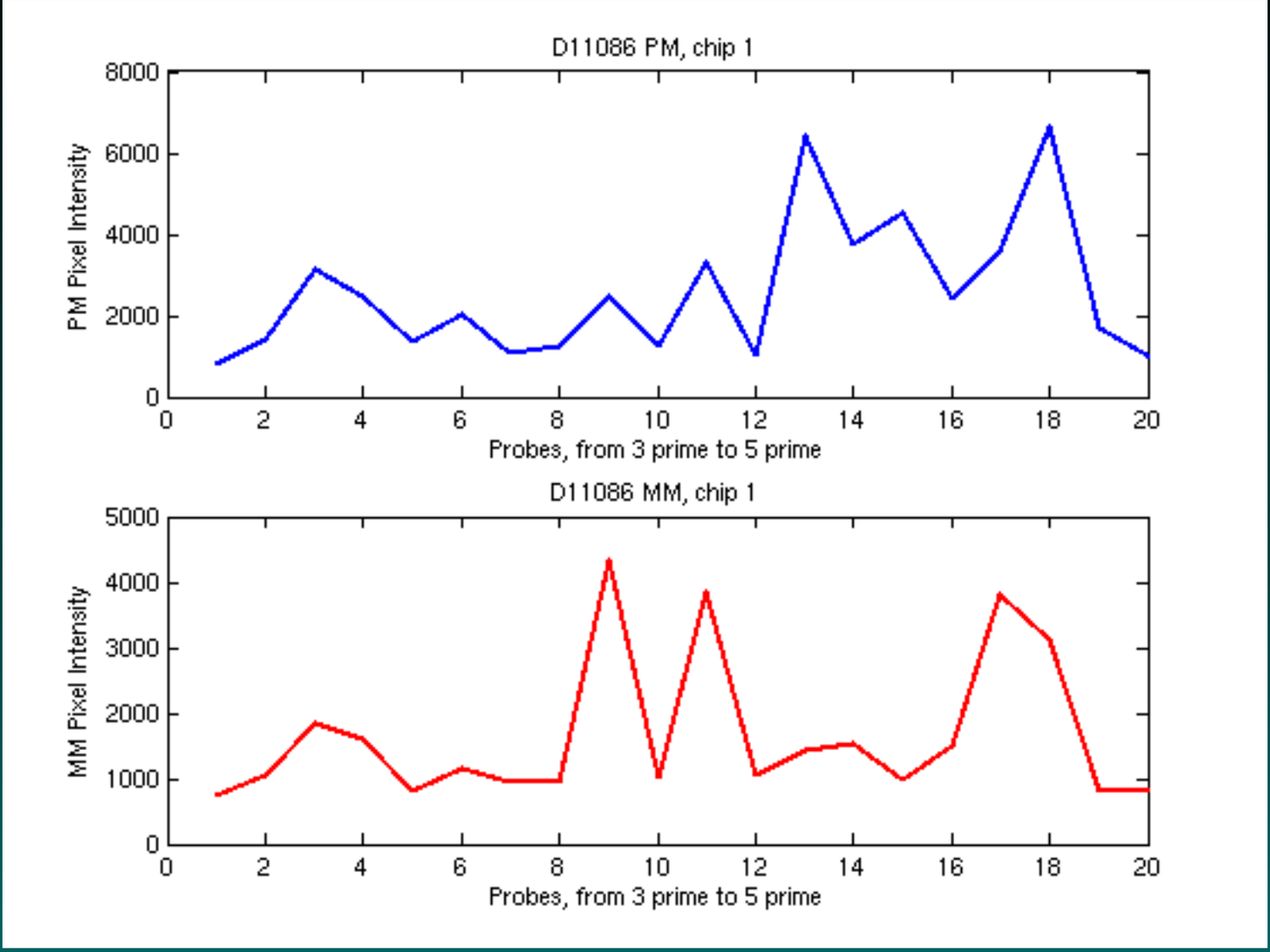
It works one chip at a time.

Using models: dChip

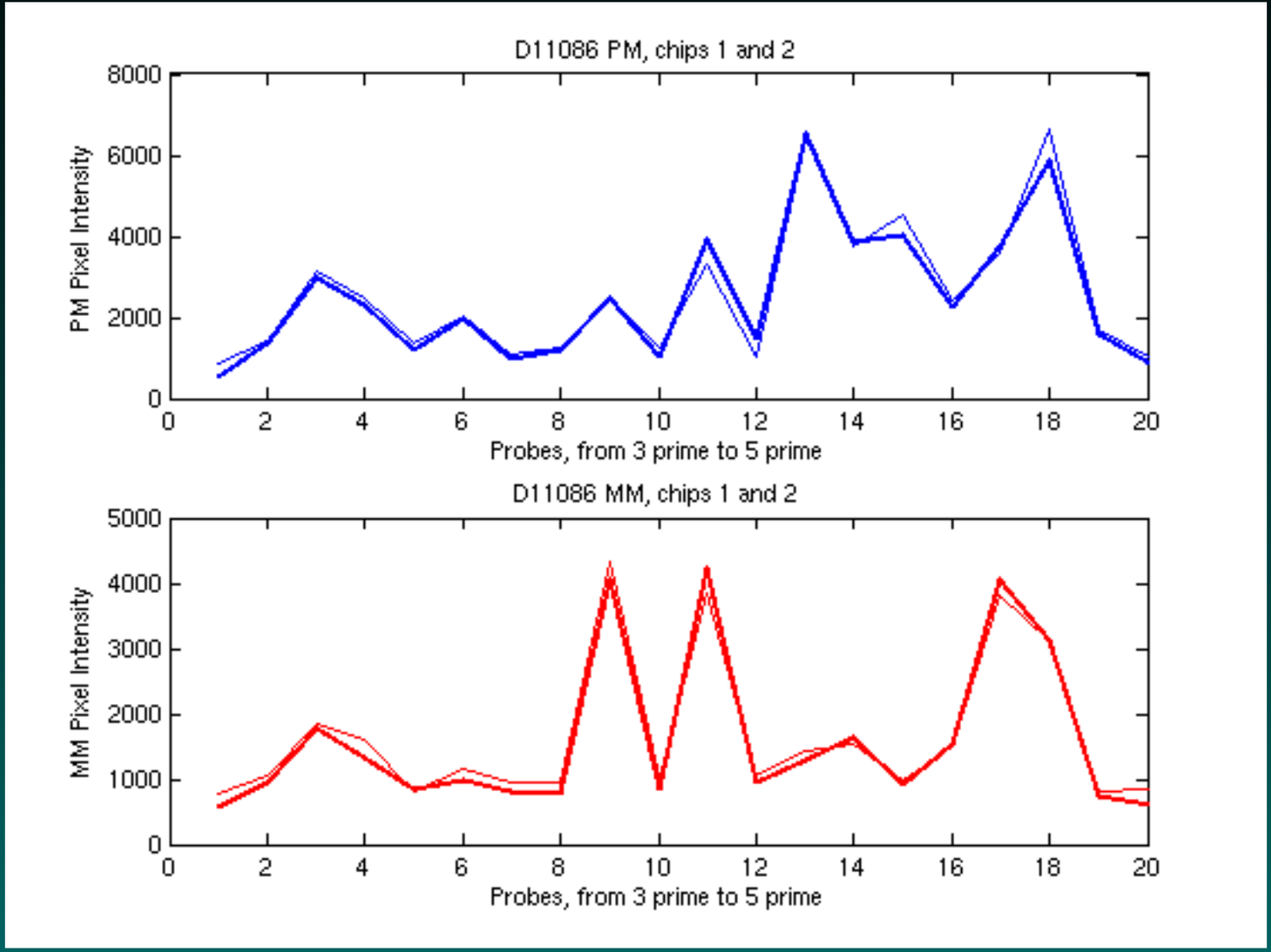
In 2001, Cheng Li and Wing Wong introduced a new method of summarizing probeset intensities, “model-based expression indices”, or MBEI. (PNAS, v.98, p.31-36).

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.

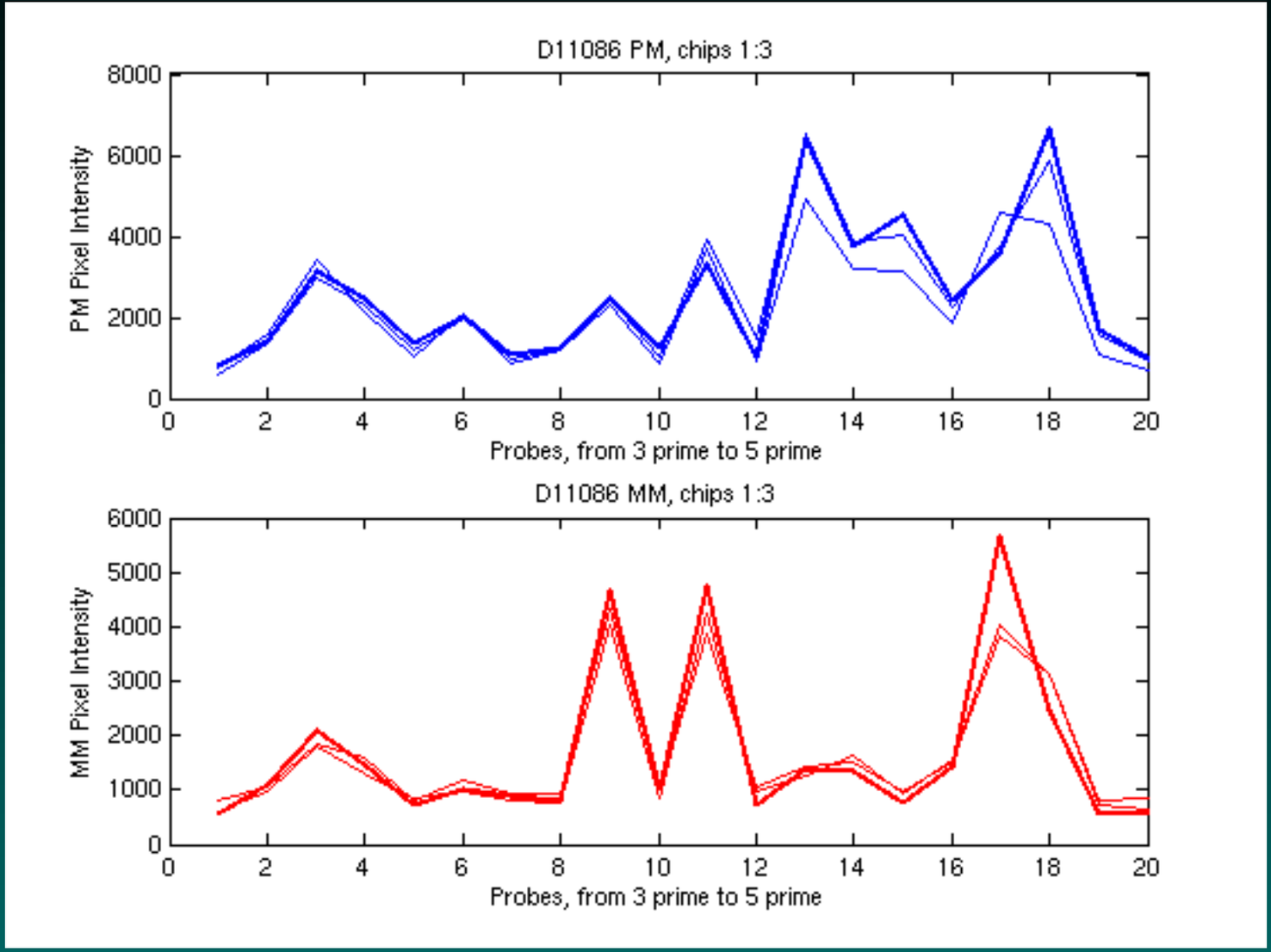
Stability: Our First Chip



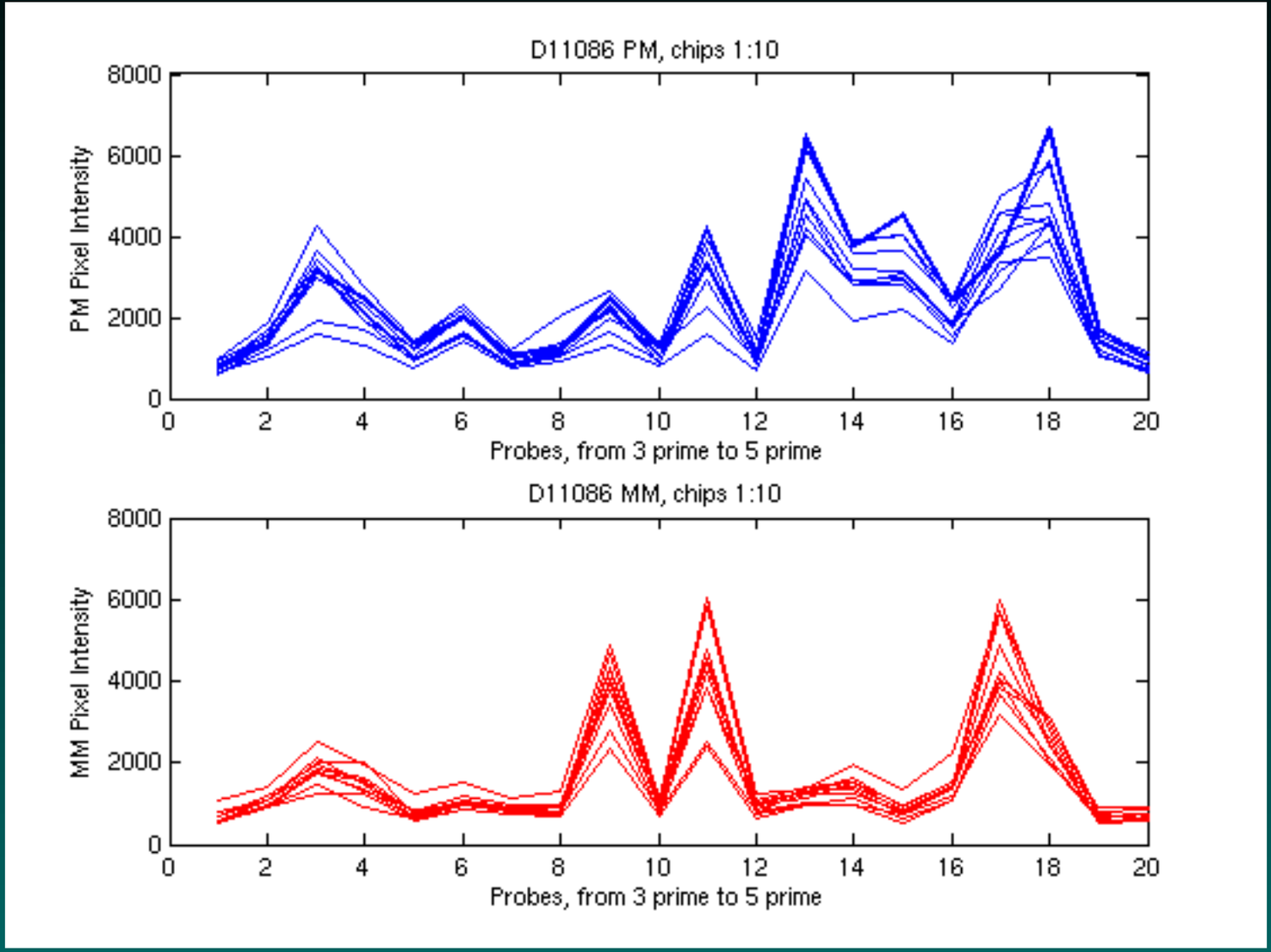
Stability: Two Chips



Stability: Three Chips



Stability: Ten Chips



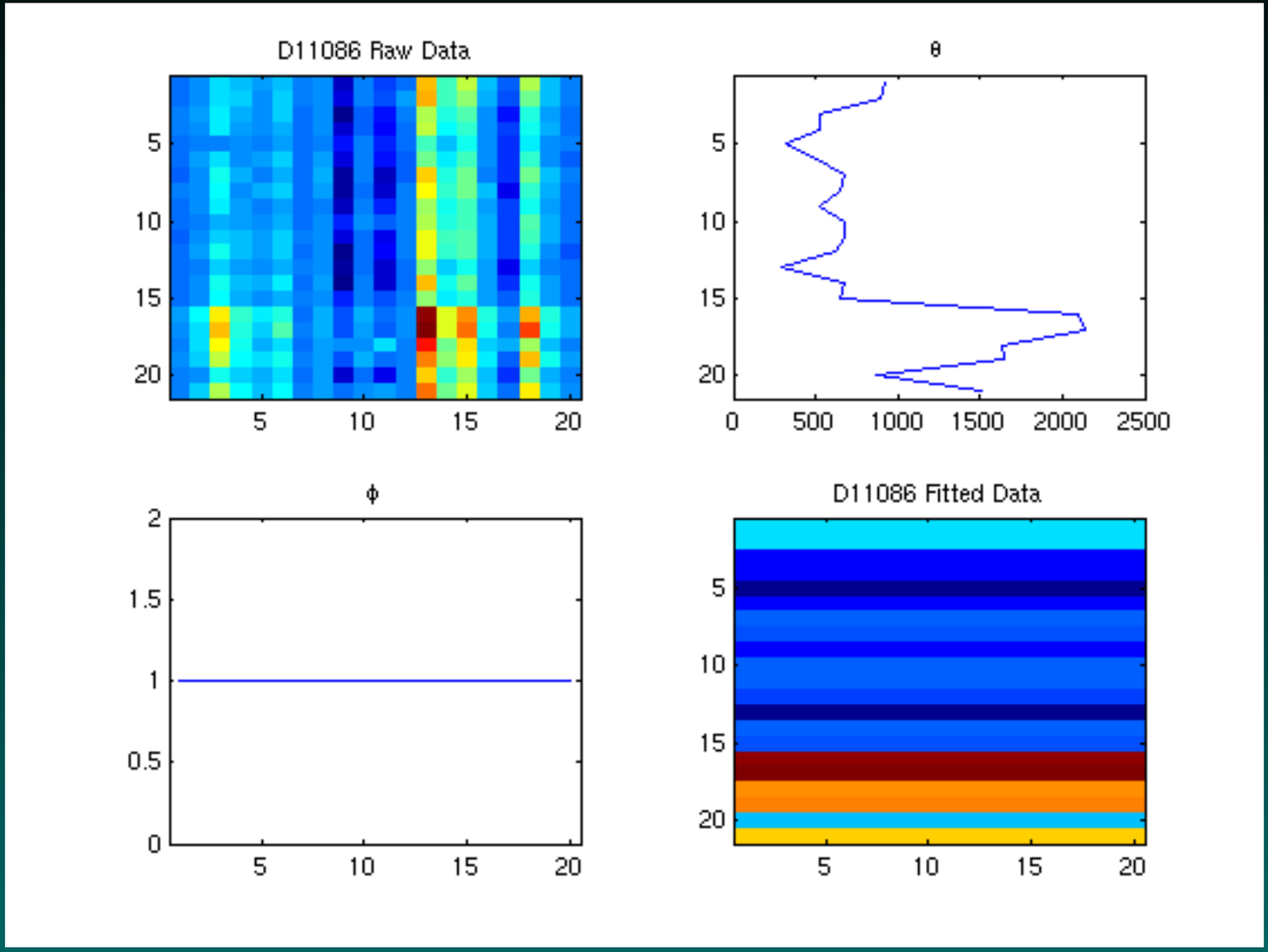
So, how to exploit this?

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

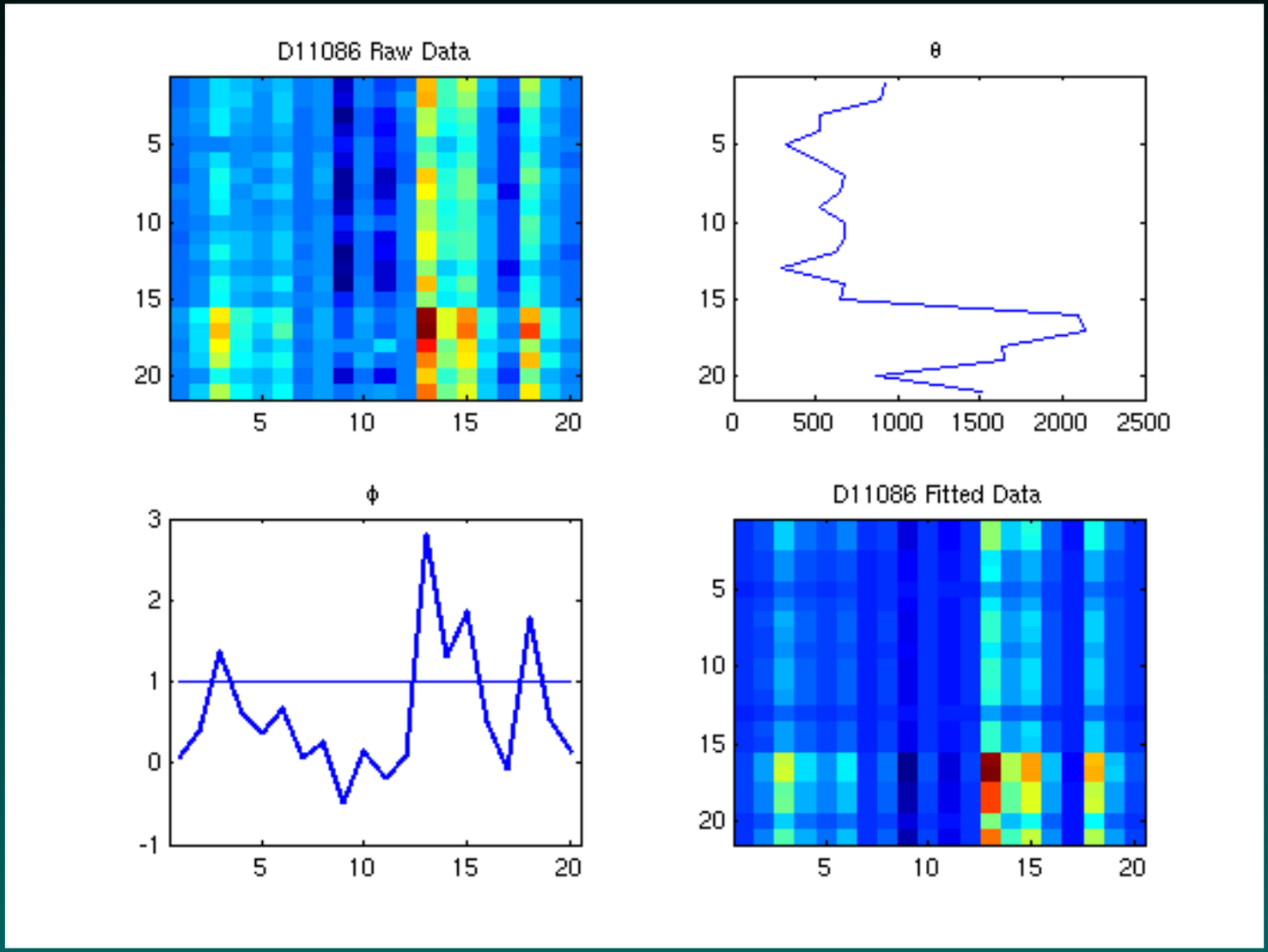
$$MM_{ij} = \nu_j + \theta_i \alpha_j + \epsilon$$
$$PM_{ij} = \nu_j + \theta_i \alpha_j + \theta_i \phi_j + \epsilon$$

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

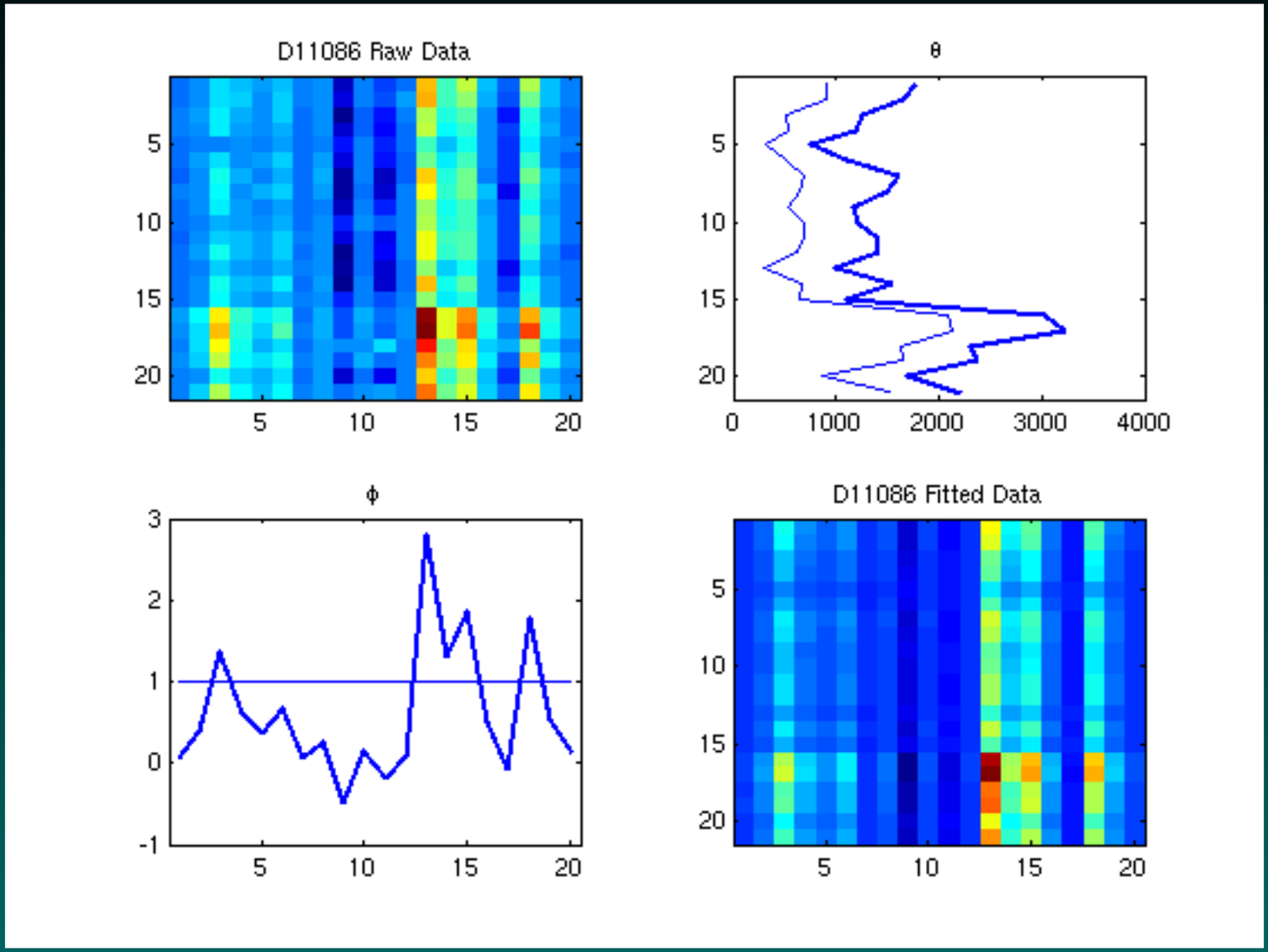
Fit using several chips at once!



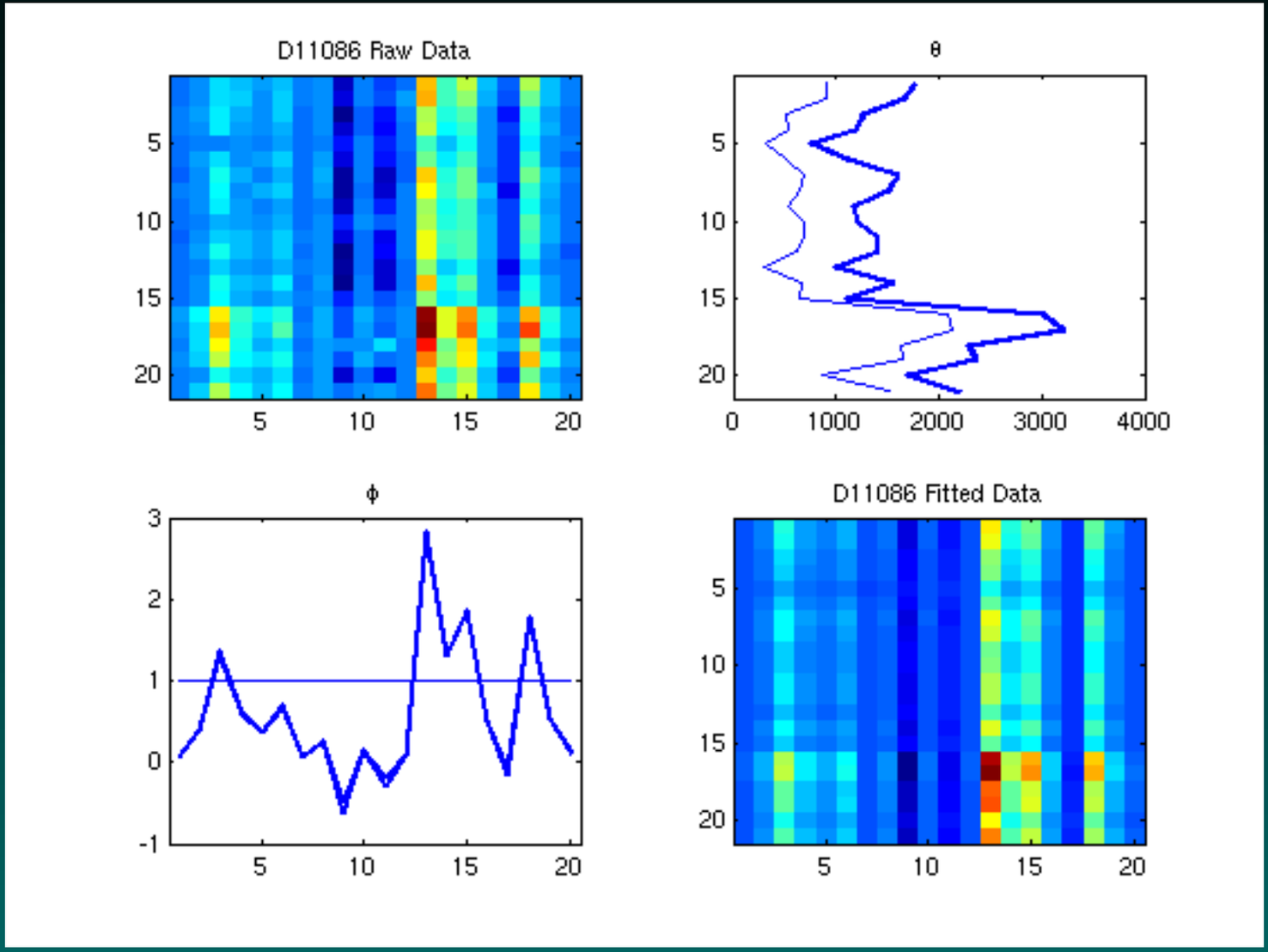
The next step: ϕ



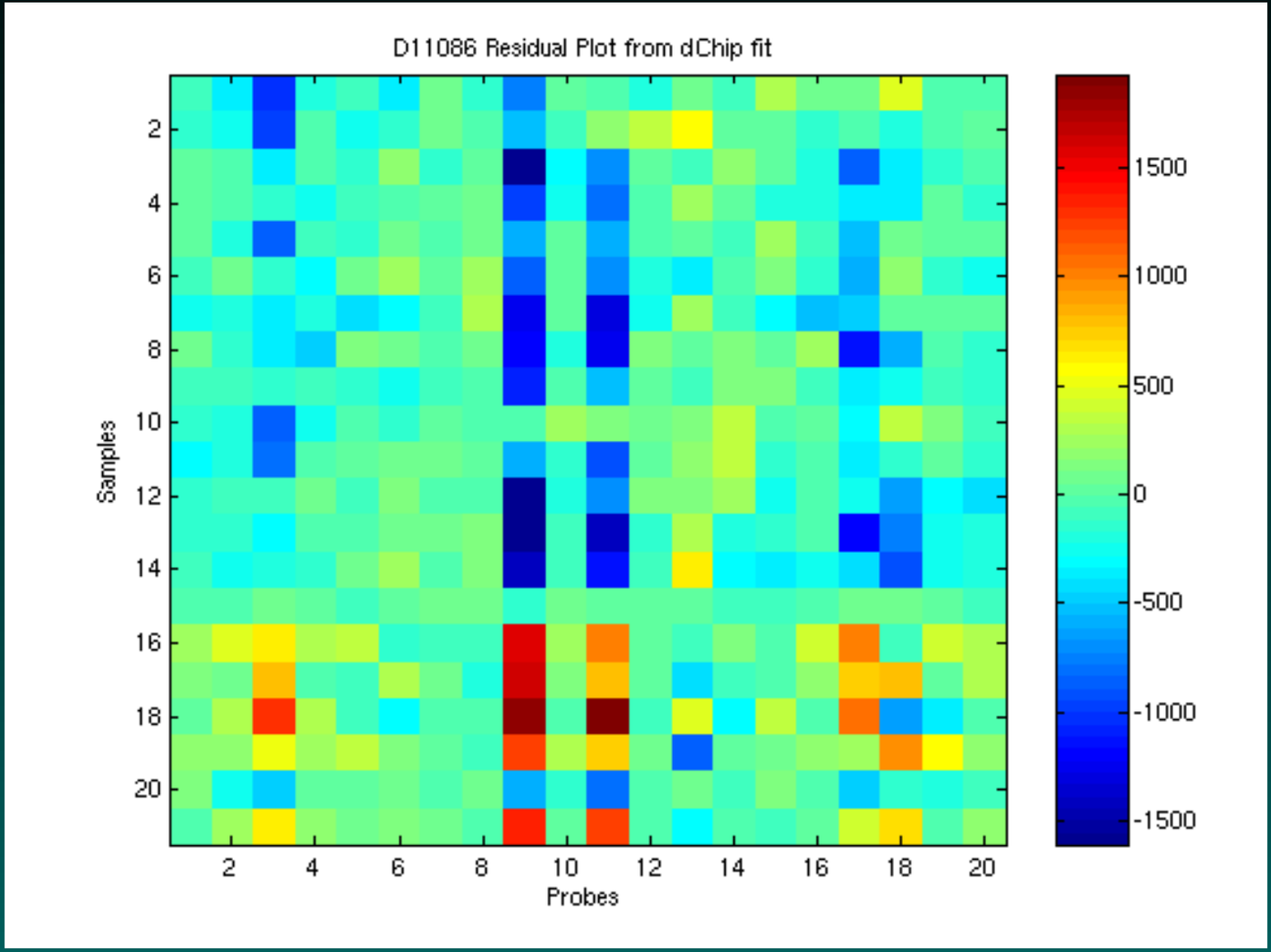
and back to θ



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.

Moving on from there...

Affy did learn some stuff from the modelling process. In particular, it noted the importance of multiplicative adjustments and statistical measures with some means of identifying outliers. They also noted that negative values just weren't well received.

Improving Robustness: MAS 5.0

So, how did they adapt the algorithms?

signal = Tukey Biweight(log($PM_j - CT_j$)).

Improving Robustness: MAS 5.0

So, how did they adapt the algorithms?

signal = Tukey Biweight(log($PM_j - CT_j$)).

Test works on the log scale (capturing multiplicative effects).

Improving Robustness: MAS 5.0

So, how did they adapt the algorithms?

signal = Tukey Biweight(log($PM_j - CT_j$)).

Test works on the log scale (capturing multiplicative effects).

Instead of the straight mismatch, they subtract a "change threshold" which is always lower than the PM value: no negative numbers.

Improving Robustness: MAS 5.0

So, how did they adapt the algorithms?

signal = Tukey Biweight(log($PM_j - CT_j$)).

Test works on the log scale (capturing multiplicative effects).

Instead of the straight mismatch, they subtract a "change threshold" which is always lower than the PM value: no negative numbers.

Use a robust weighting to downweight outliers.

Checking the fit

It was at this stage that 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.

MAS 5.0 vs MAS 4.0

The signal statistic is an improvement on AvDiff.

It tracks nominal fold changes better, and it is less variable.

What it still doesn’t do is use information across chips.

Robust Multichip Analysis: RMA

RMA (Irizarry et al, Biostatistics 2003) tries to take the better aspects of both dChip and MAS 5.0, and to add some further twists.

As with dChip, RMA is built around a model:

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

(array i, probe j).

What's different?

Robust Multichip Analysis: RMA

For starters, it tosses the MM values entirely. They contend that there are too many cases where MM ¿ PM, and hence including the MMs introduces more variability than the correction is worth.

Like dChip, it assumes a model for the data, and the parameters of this model are fit using multiple chips.

Unlike dChip, the random jitter (epsilon) is introduced on the log scale as opposed to the raw scale. This more accurately captures the fact that more intense probes are more variable.

incorporating other information: PDNN

The above methods are all mathematical, in that they focus solely on the observed values without trying to explain those values.

Why should some probes give consistently stronger signals than others?

What governs nonspecific binding?

In general, these will depend on the exact sequence of the probe, and the thermodynamics of the binding.

Fitting the thermodynamics

Recently, Li Zhang introduced the PDNN model. Unlike dChip and RMA, the parameters for the PDNN model can all be estimated from a single chip, in large part because the number of parameters is much smaller.

He posits a scenario where the chance of binding is dictated by the probe sequence, and shifts the mathematical modeling back from the expression values to the sequences.

What parameters drive the model?

The base pair at position k in the sequence.

Interactions with nearest neighbors: knowing k, we must also know what is at $k-1$ and $k+1$.

The key thing here is that redundancy in terms of the model parameters can be supplied by multiple probesets from the same chip.

Which method is best?

Well, all of the above methods are implemented in Bioconductor.

we're going to try a few head to head comparisons later. In this context, it's worth thinking about how we can define a measure of "goodness". Hmm?