

GS01 0163

Analysis of Microarray Data

Keith Baggerly and Bradley Broom
Department of Bioinformatics and Computational Biology
UT MD Anderson Cancer Center
`kabagg@mdanderson.org`
`bmbroom@mdanderson.org`

September 2, 2010

Lecture 2: The Basics of dChip

- So, why are we here?
- Getting the stuff required
- Using dChip
 - Loading Data
 - Looking at Data
 - Normalizing Data
 - Model Fitting
 - Exporting Results
- The Real World...

So, why are we here?

We want to learn about dChip.

The freeware package dChip has become quite widely used for the analysis of Affymetrix gene chip data. We're going to look at using it now.

The main web page for dChip is

`https://sites.google.com/site/dchipsoft/`

where you can download the software, get links to some publicly available data, and browse through the online manual.

Much of this lecture will follow the manual, and the associated “Tutorial using leukemia data by Armstrong et al.” (under Tutorials and courses).

Step 1: Get dChip

This step is fairly trivial; simply download the latest non-beta version (dchip_2010_01.exe as of September 1, 2010) and put the application somewhere (eg, C:Program Files/dChip/). We keep this application on a shared drive at

`/data/bioinfo/affymetrix/00 Affymetrix Info/DChip Files`

The entire application is about 2.1M in size. At present, dChip only runs on Windows platforms. Some success has been reported using windows emulators on the Mac, but there is a performance hit.

A Biological Example

There is a genetic translocation that occurs in ALL, associated with a mixed-lineage leukemia gene (MLL). Patients with this translocation have noticeably worse outcomes. It is thought that this translocation may make the disease qualitatively different, and somewhat closer to AML. If the disease is different, we may want to adjust the therapy as well.

Using Affymetrix gene chips, can we identify differences between ALL, MLL, and AML?

Step 2: Get CEL Files

Armstrong et al (2001), Nat Gen, 30:41-7. The CEL files (ALL,MLL,AML) from Dana Farber:

```
http://www.broad.mit.edu/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=63
```

The CEL files are available as gzipped tar files, which WinZip should be able to uncompress. There are 6 CEL file collections at this site, each about 35-41M in size, or 100-127M in size when uncompressed. These files contain about 10-12 CEL files each.

The suffixes on these files should be .tar.gz. Earlier, for some reason they were tar.tar. This latter suffix needs to be changed so that the file type will be recognized.

Step 2: Get CEL Files (cont)

If you are working with CEL files stored in more than one location, it is often useful to assemble a “data file list” specifying the locations of the files. This file should be a text file (and end in .txt). Every row should contain either a specific file name or a directory. An example from the manual:

```
E:\Affy data\dan\CA-H.cel  
E:\Affy data\dan\CA-HR.cel  
E:\Affy data\dan\zugen  
E:\Affy data\dan\PC-C.cel
```

Here, the AML samples were run later, so we put them in a different directory.

Step 2A: Digression on Folders

Keeping things organized is pretty important. Here's where I put things.

C:/Program Files/dchip/
Z:/dchipExample/InfoFiles
Z:/dchipExample/CELFiles
Z:/dchipExample/AMLCELFiles
Z:/dchipExample/CDFFile
Z:/dchipExample/Output

The data list file went into InfoFiles.

Step 2B: Digression on Other Info

The Dana Farber web site also supplies the quantifications that they used in their analyses, as

expression_data.txt

or

expression_data_plus_APcalls.txt

These data were initially quantified using MAS4.0 (AvDiff). We prefer to work with the CEL files as raw data and to construct our own quantifications.

Step 3: Get Explanatory Files

Also at the above site, there are files describing the sample-to-chip mapping in more detail:

scaling_factors_and_fig_key.txt

and a link to the paper that appeared in Nature Genetics describing the biological context of the problem.

We stored the above file in InfoFiles.

Step 4: Find the CDF file

This requires that we know what type of Affy chip was used. In this case (according to the paper), the chips were U95A.

For this example, a compressed version of the U95A CDF file can be downloaded from the dChip site.

Others can be acquired from the Affymetrix website,

`http://www.affymetrix.com`

Free registration may be required. Acquire CDFs for both U95A and U95Av2.

A warning – the cdf extension is also used for “channel files” by Microsoft, so don’t worry if you see a weird icon.

Step 4A: Digression

Actually, the CDF file for these chips is a bit tricky.

There is a set of U95 chips, U95A,U95B,...,U95E that contain probes for all genes in the genome. The probes were assembled using the 95th build of the Unigene database to define what a “gene” was. However, while these chips surveyed the genome, most of the probes corresponding to “interesting” genes were put on the A chip, so most people just bought those as opposed to the set.

Soon after the U95A release, some mistakes were noted in the probe design, and Affy released the U95Av2, which is the type we have encountered more frequently here at MDA.

Can you tell them apart?

Step 5: Get the Gene Info file(s)

Every chip type has a fixed set of probesets printed on it, but the probeset identifiers are typically not enough to suggest anything (1389_at?). We need more context – is there a common name for the associated gene? Which chromosome is it on, and where? Is the gene known or thought to be part of a functional family (eg, cytoskeleton)? Are there IDs that can let us look up more information in national databases?

The above information for each chip type has been collected and assembled into GeneInfo files available at the dChip website. *These files are tab-delimited text files, but they've had an xls extension placed on them so that Excel is the default program for opening them.*

These info files can change over time!

Step 5: Get the Gene Info file(s) (cont)

Actually, when we download the zip file from the dChip web site, we get 3 files:

HG-U95Av2 gene info2.xls

HG-U95Av2 gene info2 Gene Ontology.xls

HG-U95Av2 gene info2 Protein Domain.xls

We're going to look at each of these in turn, but I want to quickly note that these files are for the U95Av2 chip, as opposed to the U95A chip. In terms of the probesets that were used, the overlap is so large (12600 of 12625) that working with these should be fine.

Again, we've stored these in InfoFiles.

HG-U95Av2 gene info2.xls

The first few entries:

```

Probe Set Name : Identifier : LocusLink :
Name : Gene Ontology.xls : Protein Domain.xls :
Pathway : Chromosome : Description
1000_at : X60188 : 5595 :
mitogen-activated protein kinase 3 : |7165|
7154|6935|42330|9605|6928|8151|4707|4702|4674|
4672|16301|3824|16773|16772|16740|5057|4871| :
|2290|719|3527| : : |16|16p|16p12| :
X60188 /FEATURE=mRNA /DEFINITION=HSERK1 Human ERK1
mRNA for protein serine/threonine kinase
1001_at : X60957 : 7075 :
tyrosine kinase with immunoglobulin and epidermal
growth factor homology domains : |7498|9888|

```

HG-U95Av2 gene info2 Gene Ontology.xls

Term ID	Term Name	Frequency	
3	reproduction	101	
18	regulation of DNA recombination	9	
41	transition metal transport	16	
67	DNA replication and chromosome cycle	10	
70	mitotic chromosome segregation	7	
72	M-phase specific microtubule process	8	
74	regulation of cell cycle	330	
75	cell cycle checkpoint	35	
76	DNA replication checkpoint	8	

HG-U95Av2 gene info2 Protein Domain.xls

Term ID	Term Name	Frequency
1	Kringle 16	
2	Cdc20/Fizzy	4
3	Retinoid X receptor	15
4	Saposin type B	5
5	Helix-turn-helix, AraC type	11
6	Vertebrate metallothionein, family 1	6
7	Tubby 7	
8	C2 domain	84
10	Cysteine proteases inhibitor	18

“what ghastly names they all have...” E. J. (Ernest John) Moncrieff

Step 6: Get the Sample Info file

Most of the files that we have worked with so far have described properties associated with a given chip type, not with the samples we have used. We can also supply and use sample-specific information in a tab-delimited text file. The first few entries here:

scan name	sample_name	type
CL2001011101AA	ALL_1 A	A
CL2001011104AA	ALL_2 A	A
CL2001011105AA	ALL_3 A	A
CL2001011108AA	ALL_4 A	A
CL2001011109AA	ALL_5 A	A

The file at the dChip website is incomplete. ALL samples are type A, MLL type M, and AML are blank. I augmented my sample info to use 3 letter acronyms. *Working Directory, Output.*

Step 6: Get the Sample Info file (cont)

The header row and the first two columns are required, but any columns beyond that are at our discretion. By default, column values are treated as factors, but adding the string “(numeric)” to a column name will override this.

What else could we have included?

- Presence/absence of other translocations
- train/test status
- specimen type (diagnostic, relapse)
- run date...

Step 7: write a README file

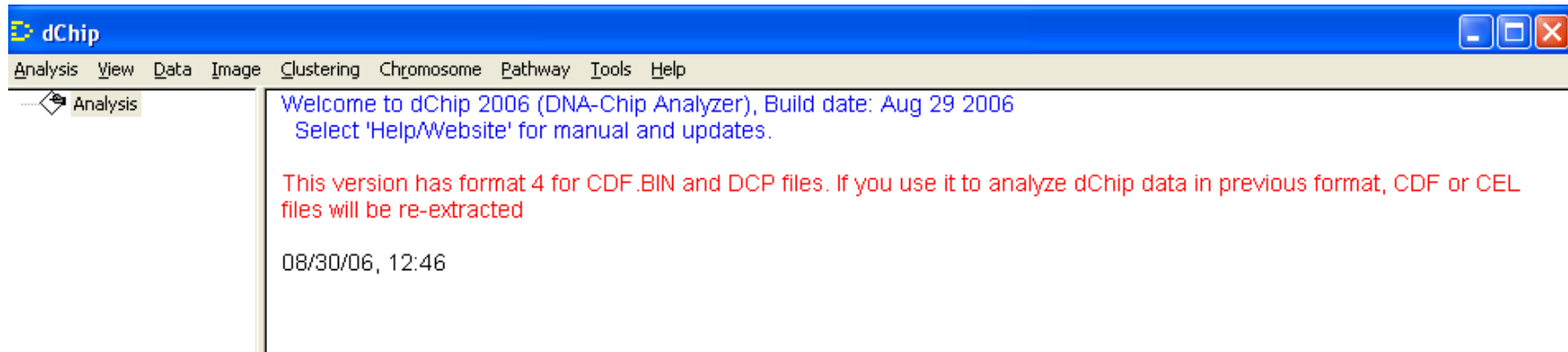
Strictly speaking, this is not mentioned in the Lab or Tutorial, but I'll put it here, right before actually running the program.

What is the biological question you are seeking to address?
What contrasts of data samples will allow you to address this?

Sending a brief description of this type off to the investigator before running the analysis can save some time...

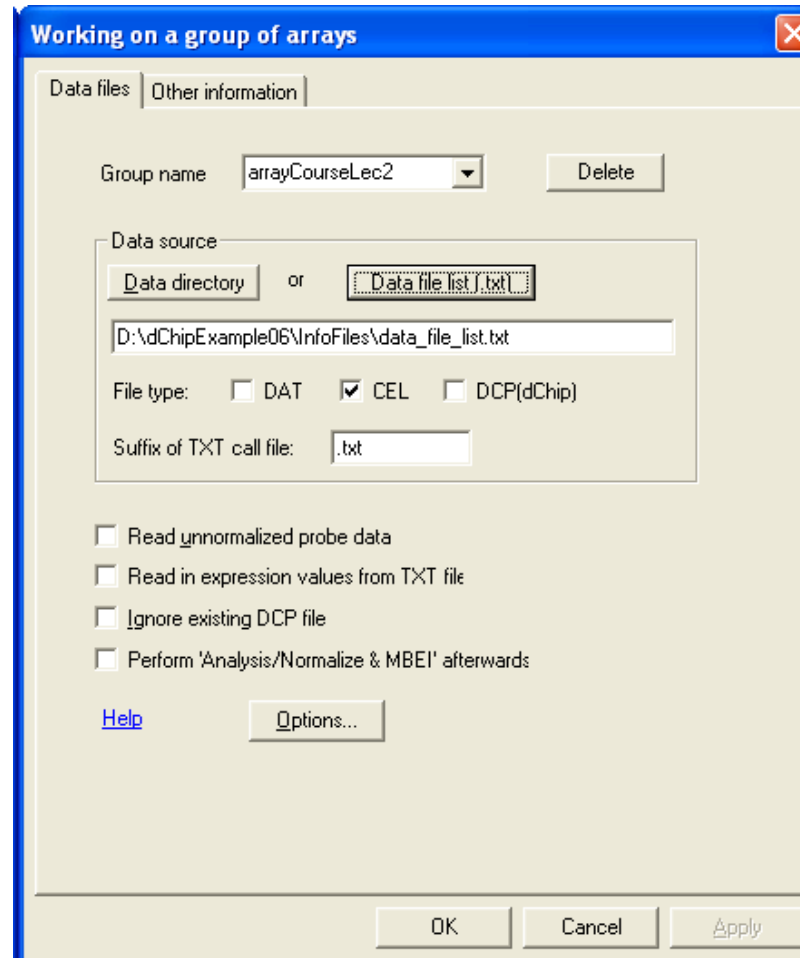
Step 8: run dChip

Nice, friendly, unexciting...



Now, we need to tell it where to find the data for analysis. Go to Analysis/Open Group.

Finding files, part 1



assign a group name, locate data files

Finding files, part 2

Working on a group of arrays

Data files | Other information

CDF file (Chip description file)

Select: [Help](#)

Ignore existing .cdf.bin file

Subarray CDF: [Help](#)

Probe sequence:

Probeset mask file: [Help](#)

Array type:

Information files

Gene or SNP: [Help](#)

(Do not specify genome information file)

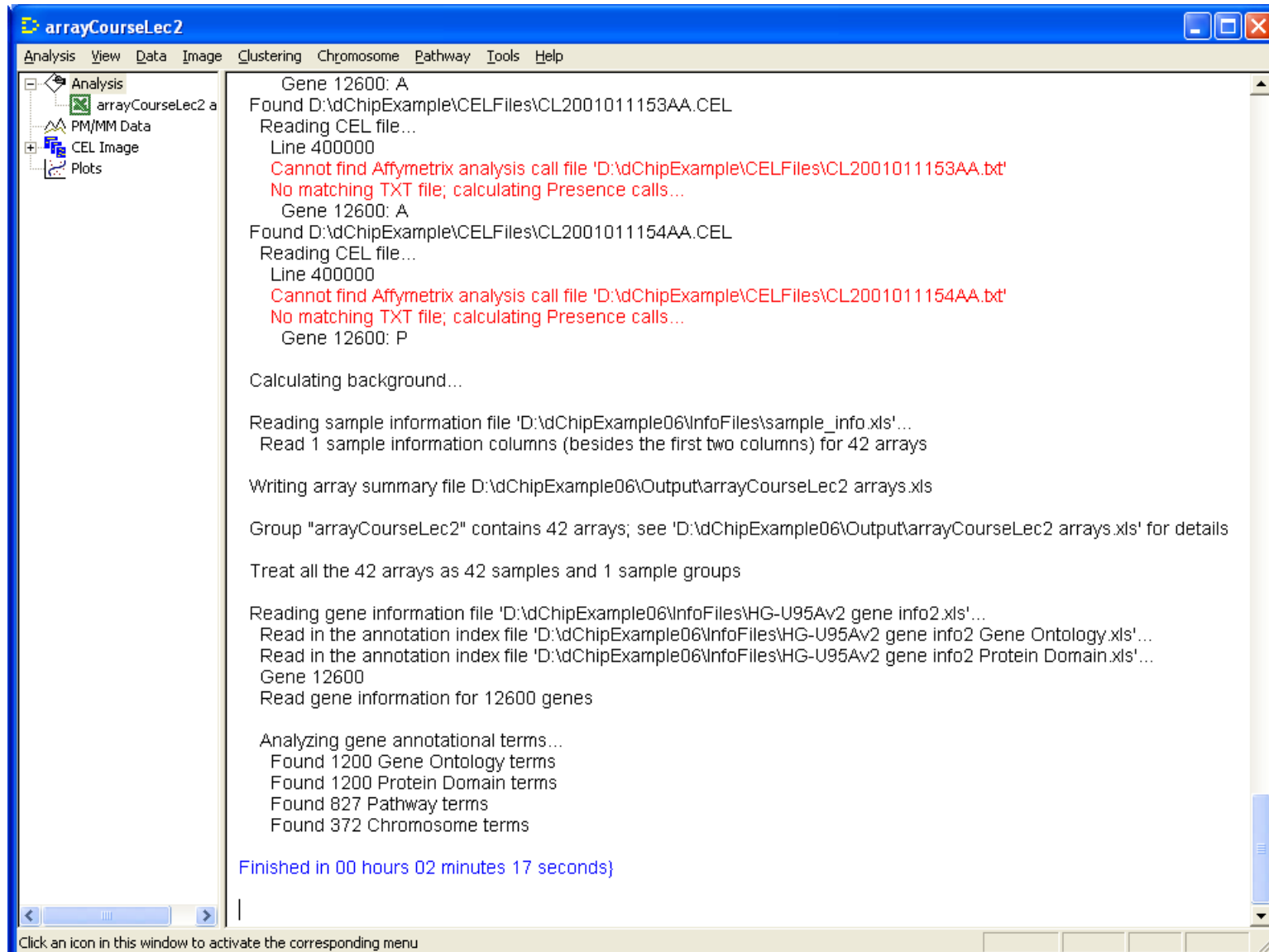
Sample: [Help](#)

(Probe set mask file, gene/SNP and sample information file are optional)

OK Cancel Apply

locate CDF, gene info, and sample info files. Under Options, we can set the working directory (Output) to store results.

Reading files



What has been wrought?

For each CEL file, a binary “dcp” file has been produced (in CELFiles):

CL2001011101AA.CEL	10,393	KB
CL2001011101AA.dcp	1,764	KB
CL2001011102AA.CEL	10,324	KB
CL2001011102AA.dcp	1,764	KB

$$(2 * 640^2) * 2 = 1638400$$

Keep the means as 16-bit integers, and allocate space for 2 CEL equivalents in each dcp file – 1 for the raw data, and 1 for the processed data.

This saves space, and uses an intelligent data structure.

What has been wrought?

A binary version of the CDF file has been produced for quicker processing.

HG_U95A.CDF	29,814	KB
HG_U95A.CDF.bin	7,092	KB

What has been wrought?

2 interim files have been produced:

arrayCourseLec2.ini (ProgramFiles/dChip)

arrayCourseLec2_array_summary.xls (Output)

The first is a configuration file, and is stored with the exe file. The last summarizes some aspects of the files examined, and is stored in the working directory.

The arrayCourseLec2.ini file

arrayCourseLec2.ini

```
CDF_FILE=Z:ArrayCourse09\dChipExample\CDFFile\HG_U
READ_DAT=0
READ_CEL=1
READ_DCP=0
DATA_PATH=Z:\dChipExample\InfoFiles\data_file_list
WORKING_DIR=Z:\dChipExample\Output
GOSURFER_DIR=C:\Program Files\dChip
USE_UNNORM=0
MAS5_SIGNAL=0
```

The arrayCourseLec2 arrays.xls file

arrayCourseLec2 arrays.xls

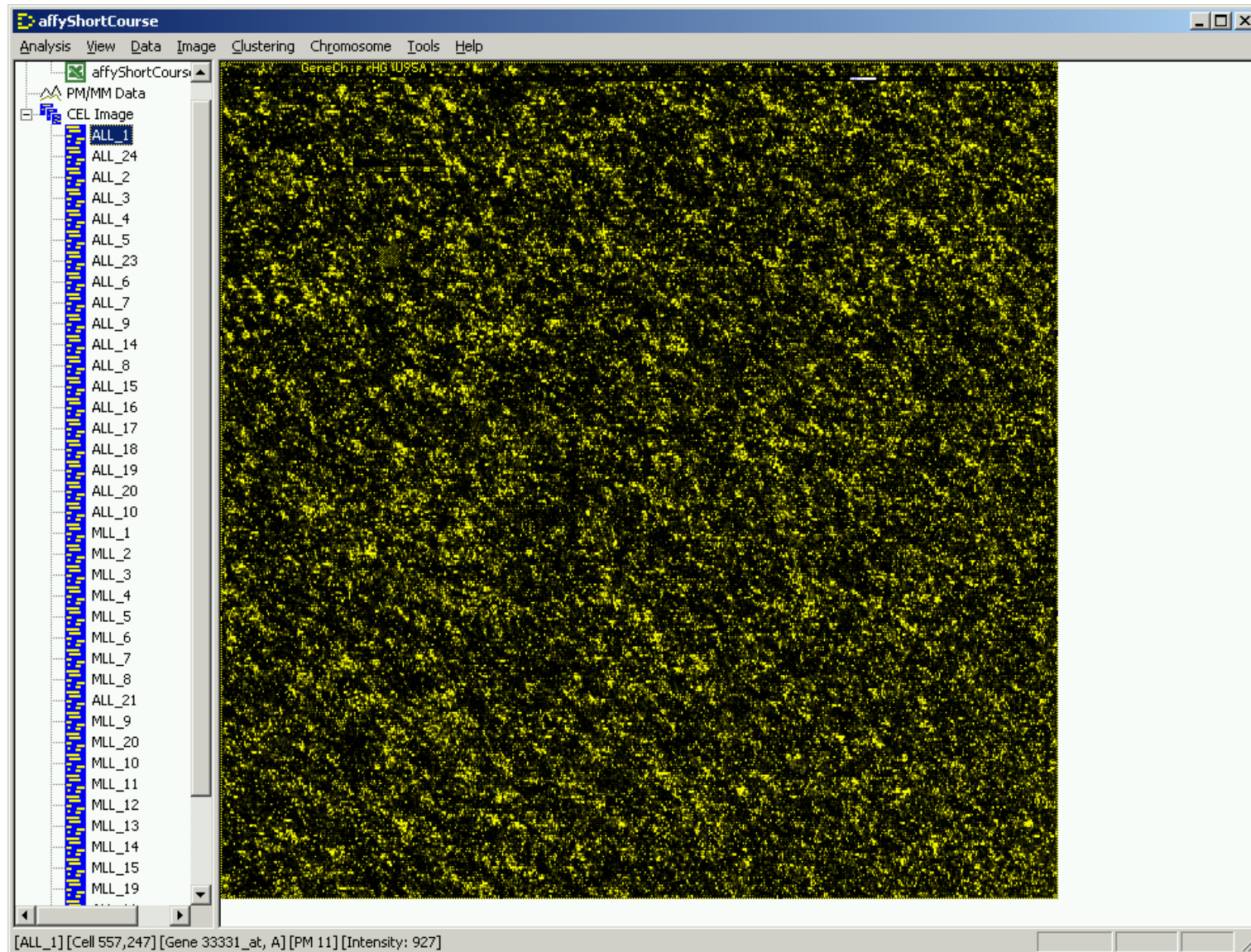
```
Number : Array : File Name : Median Intensity
(unnormalized) : P call %
1 : ALL_1 : Z:\dChipExample\CELFiles\
CL2001011101AA.CEL : 1519 : 48.2
2 : ALL_24 : Z:\dChipExample\CELFiles\
CL2001011102AA.CEL : 1202 : 38.3
3 : ALL_2 : Z:\dChipExample\CELFiles\
CL2001011104AA.CEL : 1795 : 49.5
4 : ALL_3 : Z:\dChipExample\CELFiles\
CL2001011105AA.CEL : 1106 : 36.9
```

Look at the Chips

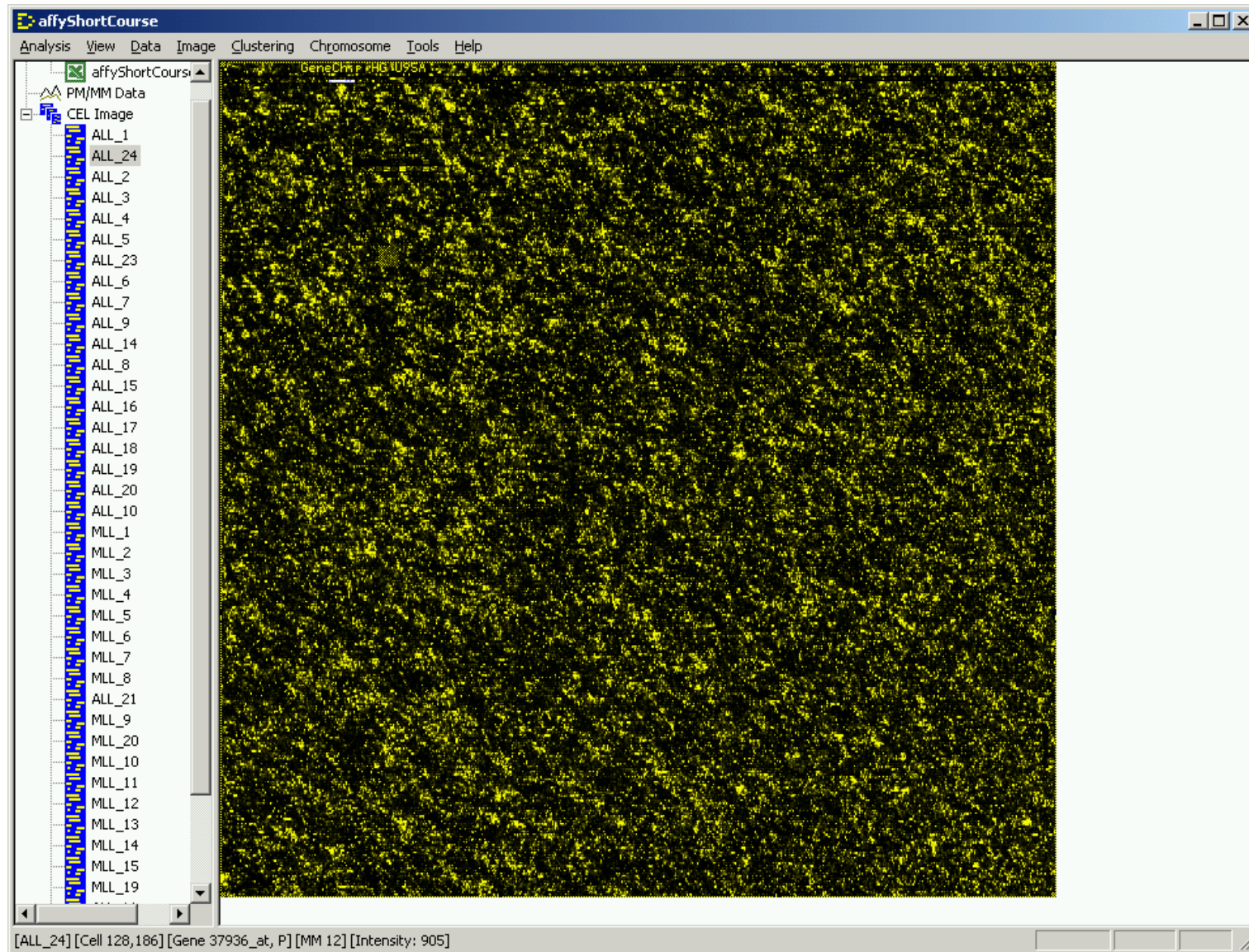
The “Short Tutorial” next suggests going to “View/CEL Image” to look at the data. Unfortunately, this is for an earlier version of dChip, as this pulldown option no longer exists.

So, we click on the “CEL Image” icon at the left of the display and cycle through. If you click on one of the file names, the up and down arrows will let you cycle through them, or Page Up/Page Down also works. The display range covers from the 1st percentile (black) to the 95th (bright yellow).

Look at the First Chip: ALL_1



Look at the Second Chip: ALL_24



Zoom In

If you click on a part of the image, you select the corresponding probe set. The arrow keys will let you zoom in on the image to look at that spots more closely.

Down arrow: zoom in lots

Up arrow: zoom out lots

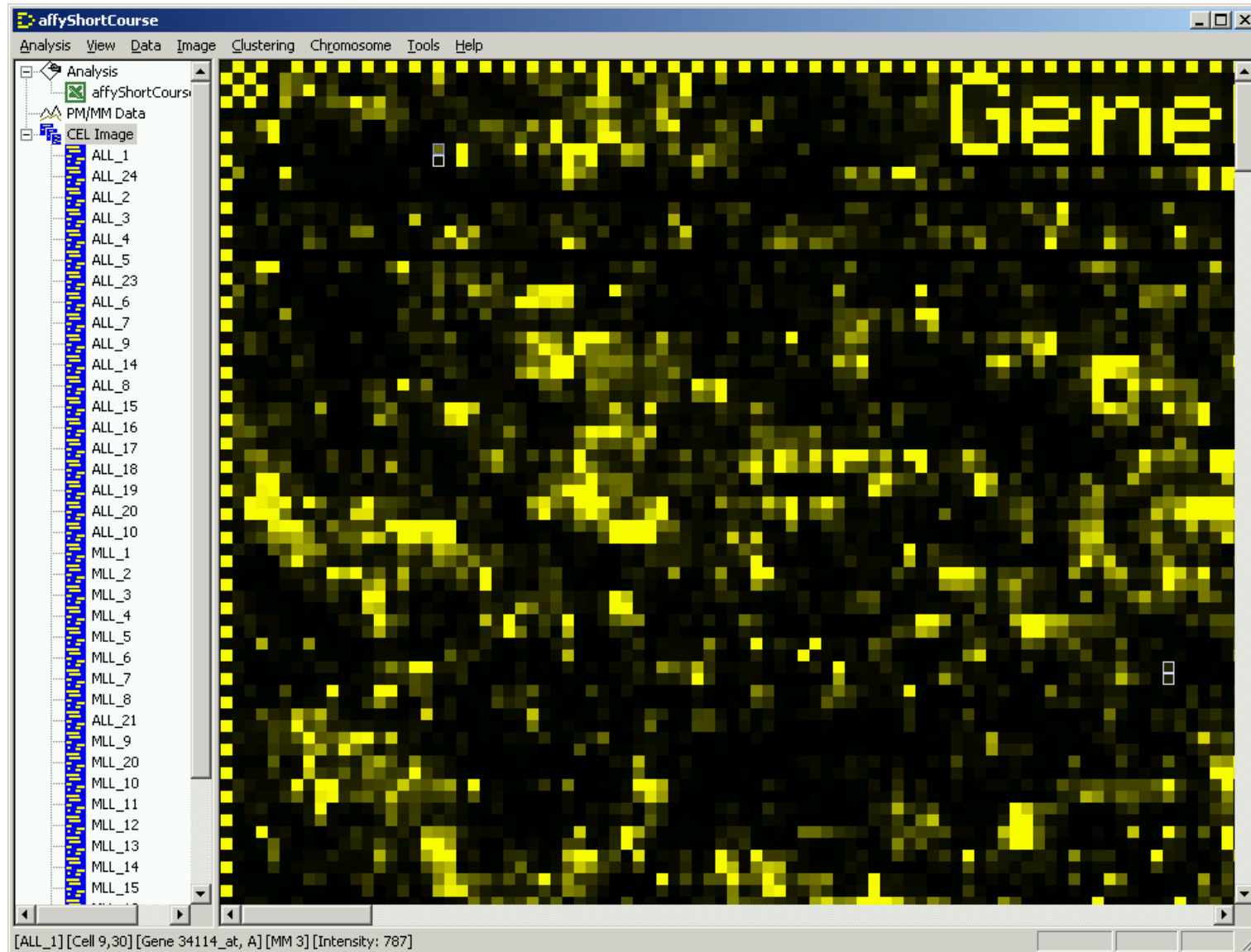
Right arrow: zoom in a little

Left arrow: zoom out a little

Scrollbars move about

Page Up and Page Down cycle you through the set of chips.

Zoom In: ALL_1



Normalize the data

go to Analysis/Normalize & Model

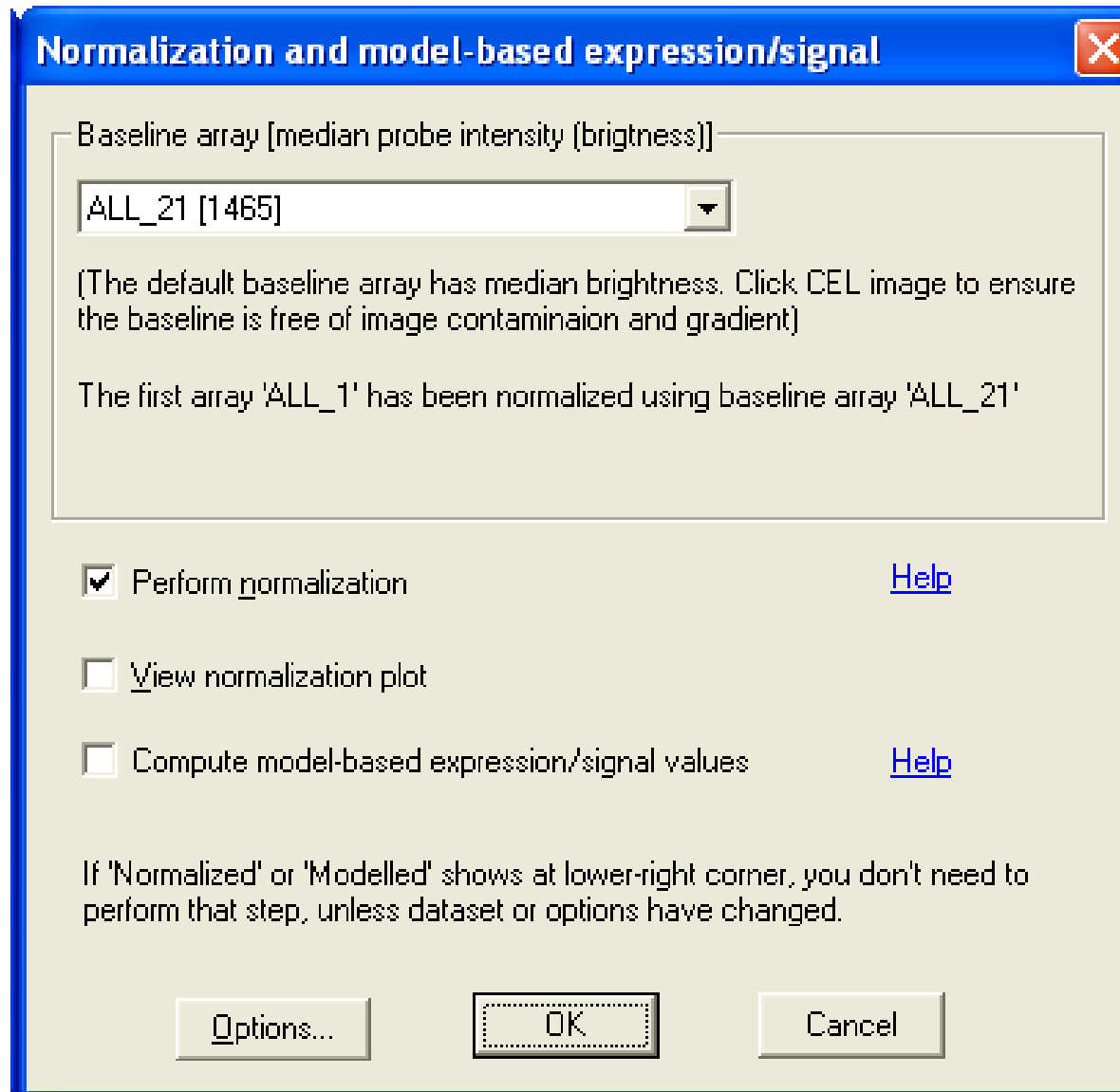
dChip will pick one array in the set to normalize all of the others to; by default it will choose the array with the median overall feature intensity.

(This can make a difference. Trying it with at least two different chips is recommended.)

For each chip, dChip then calculates an “invariant set” of features whose ranks do not change a great deal, and uses those to define a normalization curve.

Functionally, this often works like quantile normalization to the target chip.

Choosing from the menu...



Is it normalized?

The screenshot shows the Affymetrix Structure software window titled "arrayCourseLec2". The interface includes a menu bar (Analysis, View, Data, Image, Clustering, Chromosome, Pathway, Tools, Help) and a tree view on the left under "Analysis" containing "arrayCourseLec2", "PM/MM Data", and "CEL Image". The "CEL Image" folder is expanded, showing a list of files including ALL_1 through ALL_24 and MLL_1 through MLL_14. The main output pane displays the following text:

```

Searching Invariant-set: 11771
Median probe intensity: 1663 -> 1474
MLL_14
Accessing 'D:\dChipExample06\CELFFiles\CL2001011144AA.dcp' (file format 4)
Searching Invariant-set: 10003
Median probe intensity: 1267 -> 1503
MLL_15
Accessing 'D:\dChipExample06\CELFFiles\CL2001011146AA.dcp' (file format 4)
Searching Invariant-set: 12445
Median probe intensity: 1852 -> 1452
MLL_19
Accessing 'D:\dChipExample06\CELFFiles\CL2001011149AA.dcp' (file format 4)
Searching Invariant-set: 11989
Median probe intensity: 1684 -> 1459
ALL_11
Accessing 'D:\dChipExample06\CELFFiles\CL2001011150AA.dcp' (file format 4)
Searching Invariant-set: 12065
Median probe intensity: 1898 -> 1476
ALL_22
Accessing 'D:\dChipExample06\CELFFiles\CL2001011151AA.dcp' (file format 4)
Searching Invariant-set: 11426
Median probe intensity: 1592 -> 1446
MLL_18
Accessing 'D:\dChipExample06\CELFFiles\CL2001011152AA.dcp' (file format 4)
Searching Invariant-set: 11979
Median probe intensity: 1777 -> 1489
ALL_12
Accessing 'D:\dChipExample06\CELFFiles\CL2001011153AA.dcp' (file format 4)
Searching Invariant-set: 12725
Median probe intensity: 1071 -> 1486
ALL_13
Accessing 'D:\dChipExample06\CELFFiles\CL2001011154AA.dcp' (file format 4)
Searching Invariant-set: 13396
Median probe intensity: 1235 -> 1481

Calculating background...

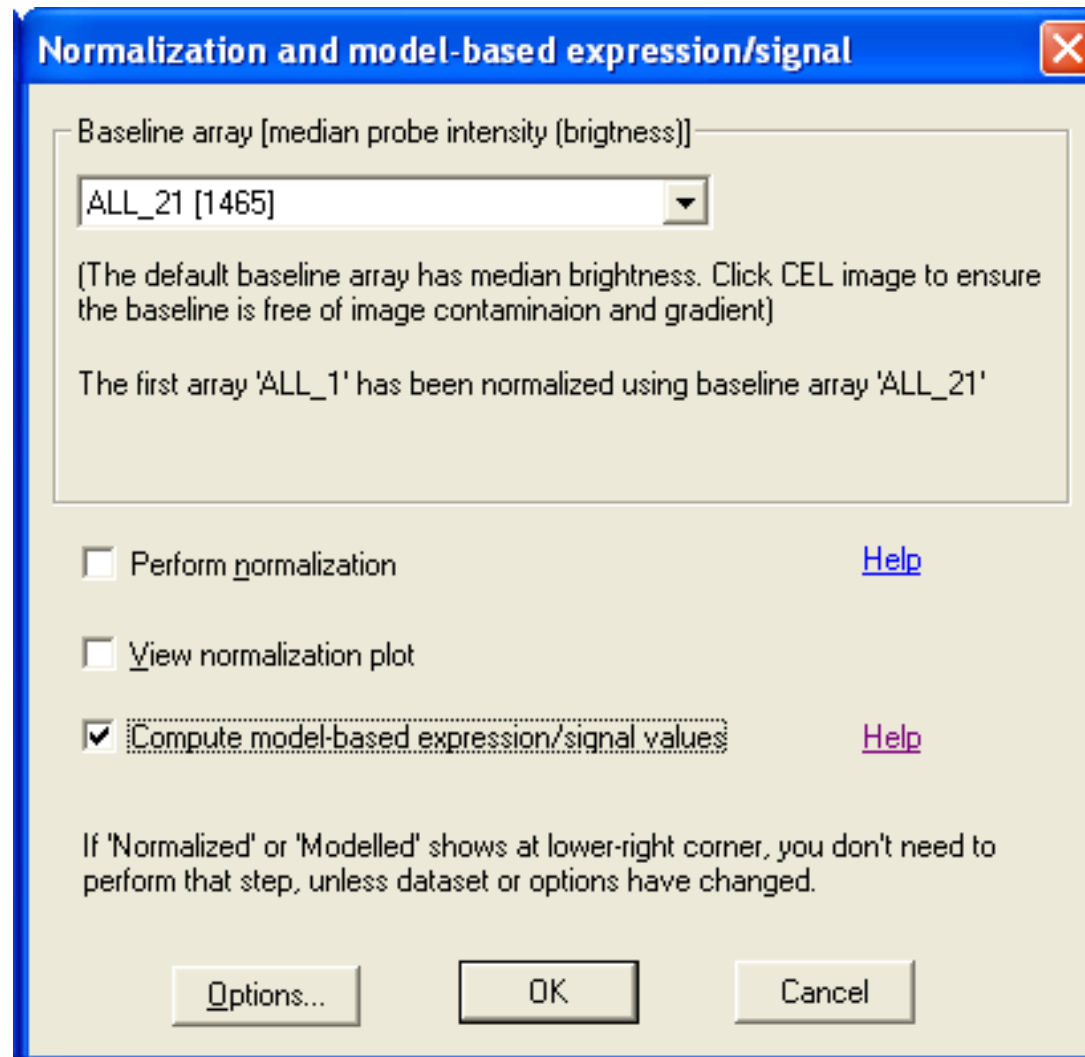
Finished in 00 hours 01 minutes 16 seconds)

```

At the bottom of the window, the status bar indicates "Analysis outputs" and a "Normalized" checkbox is checked.

Fit the Model 1

go to Analysis/Normalize & Model

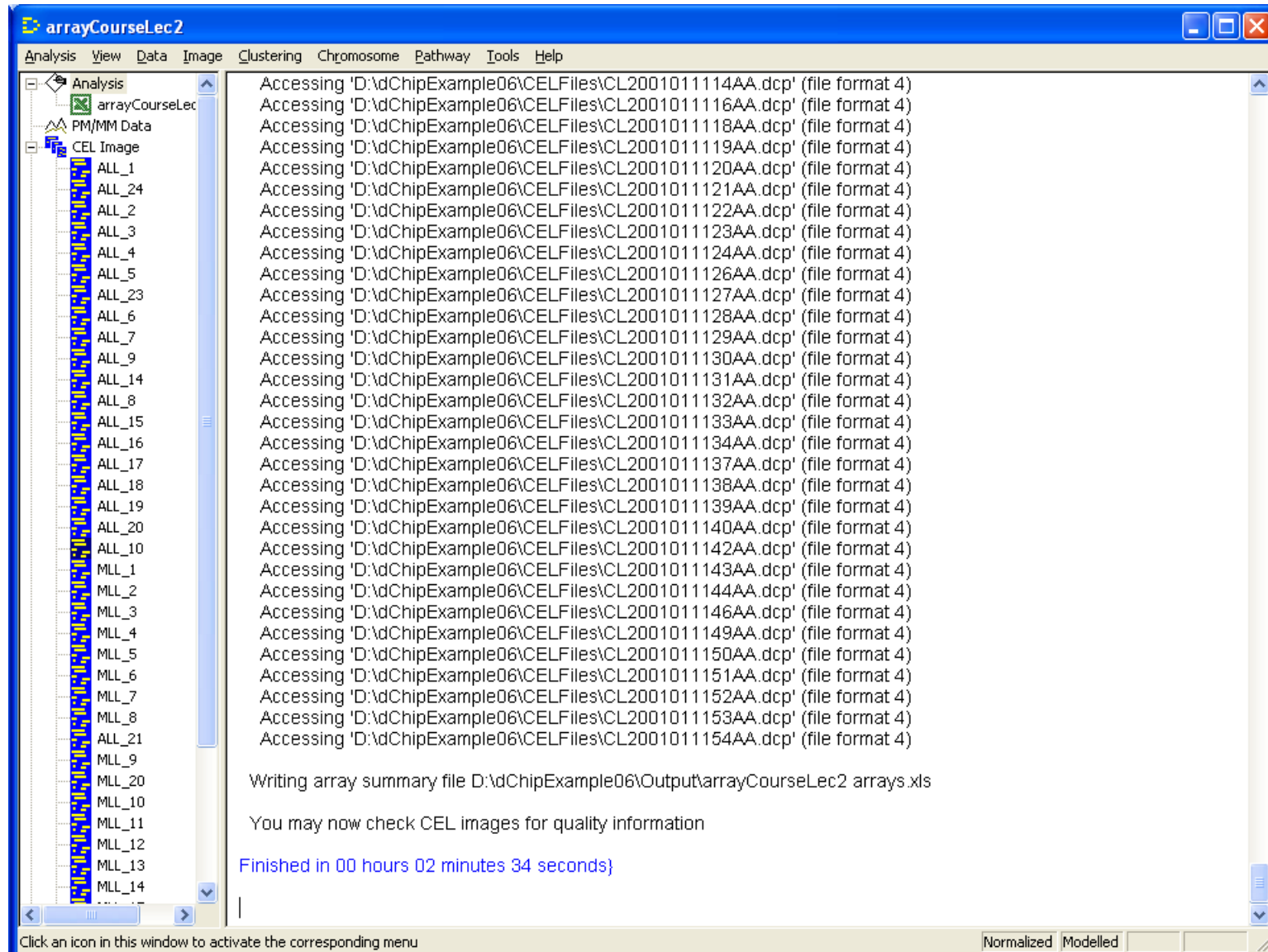


Fit the Model 2

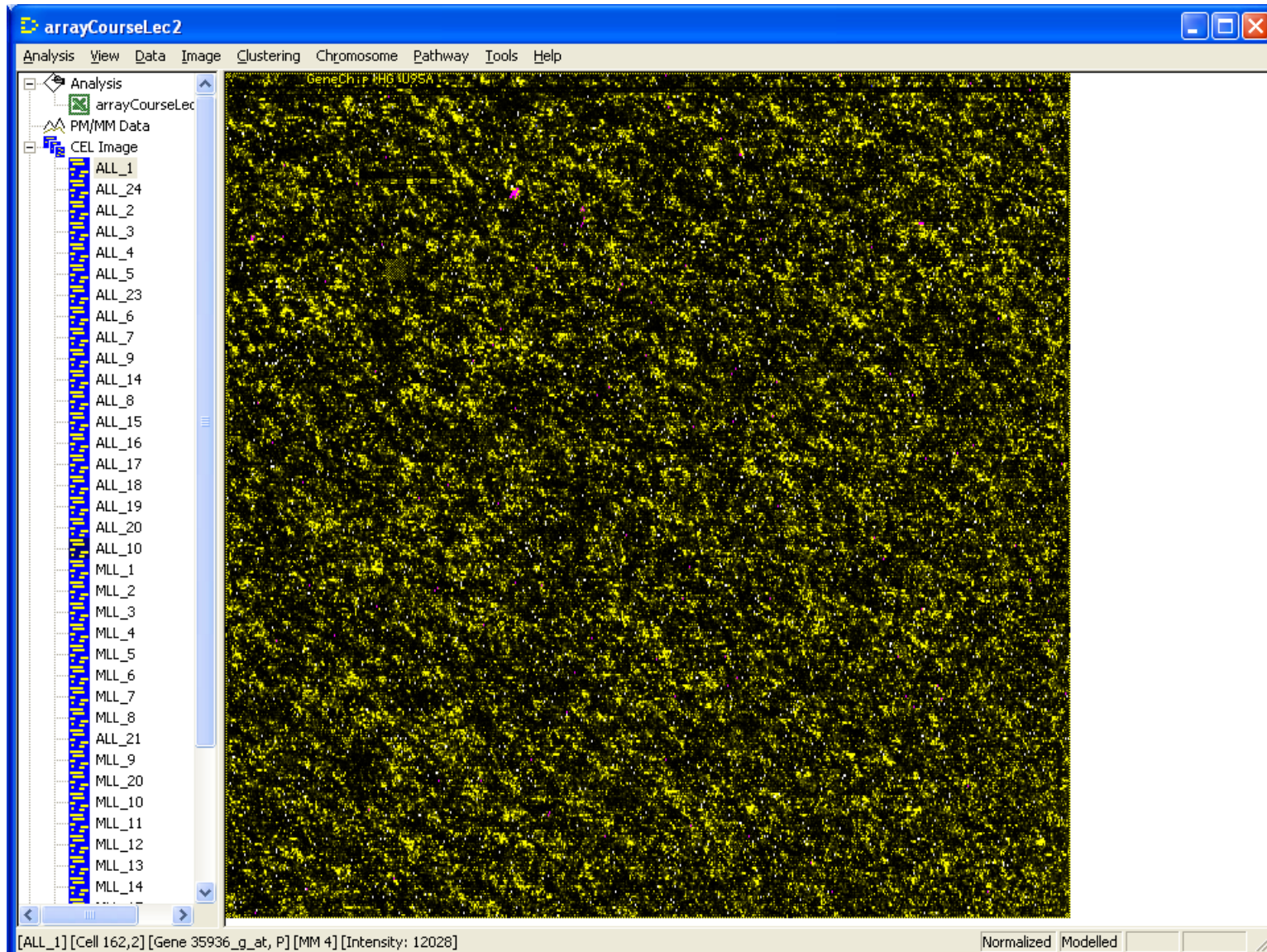
Choose “Options” and select the PM-only model (BG)

The screenshot shows the 'Options' dialog box with the 'Model' tab selected. The 'Model-based expression/signal value' section includes a dropdown for 'Model method' set to 'Model-based expression', a dropdown for 'Background subtraction' set to '5th percentile of region (PM-only)', a checked checkbox for 'Check single, array and probe outliers', an unchecked checkbox for 'Do not call all replicate arrays as array outlier', an 'Exclude' field set to '0' with the text '5' probes (For degraded or two-round amplified)', and an unchecked checkbox for 'Compute signals separately for A and B allele for SNP arra'. The 'Probe sensitivity index (PSI) file' section has a 'Usage' dropdown set to 'Do not use' with a 'Help' link, and a 'File' field containing 'D:\Program Files\vdChip2006\arrayCourseLec2.psi'. The 'Normalization' section has a 'Use selected probes' dropdown set to 'Invariant set', a 'Probe set file' field set to 'None', and a 'Smoothing method' dropdown set to 'Running median'. At the bottom are buttons for 'Reset Default', 'Print Settings', 'OK', 'Cancel', and 'Apply'.

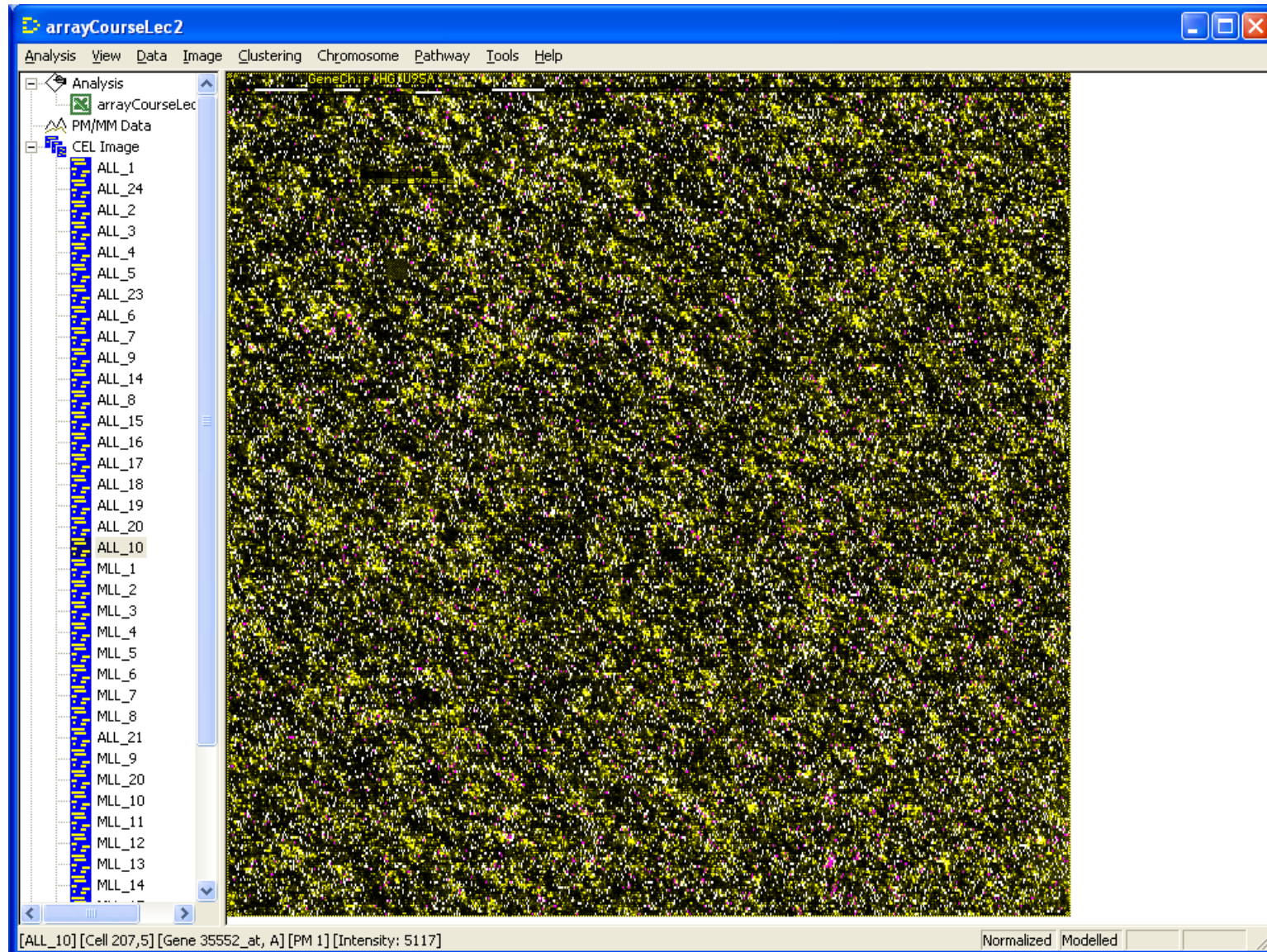
Fit the Model 3



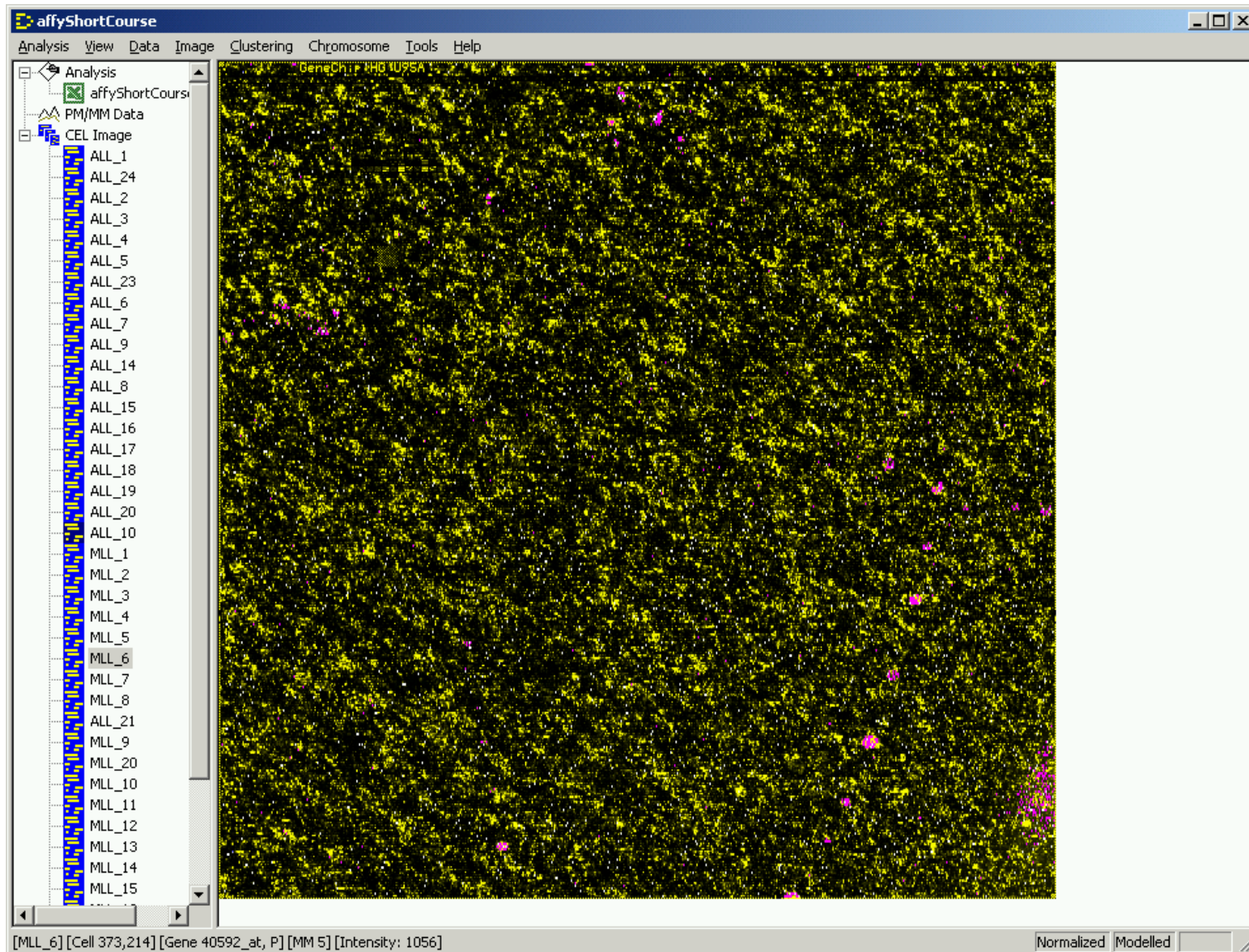
Look at the Chips, with Cues



Look at the Chips, with Cues



Look at the Chips, with Cues



Residual Checking is Useful

Hitting the “o” key toggles the display of outliers, which can let us look at the values underneath to see if we can spot what the model is picking up.

The file

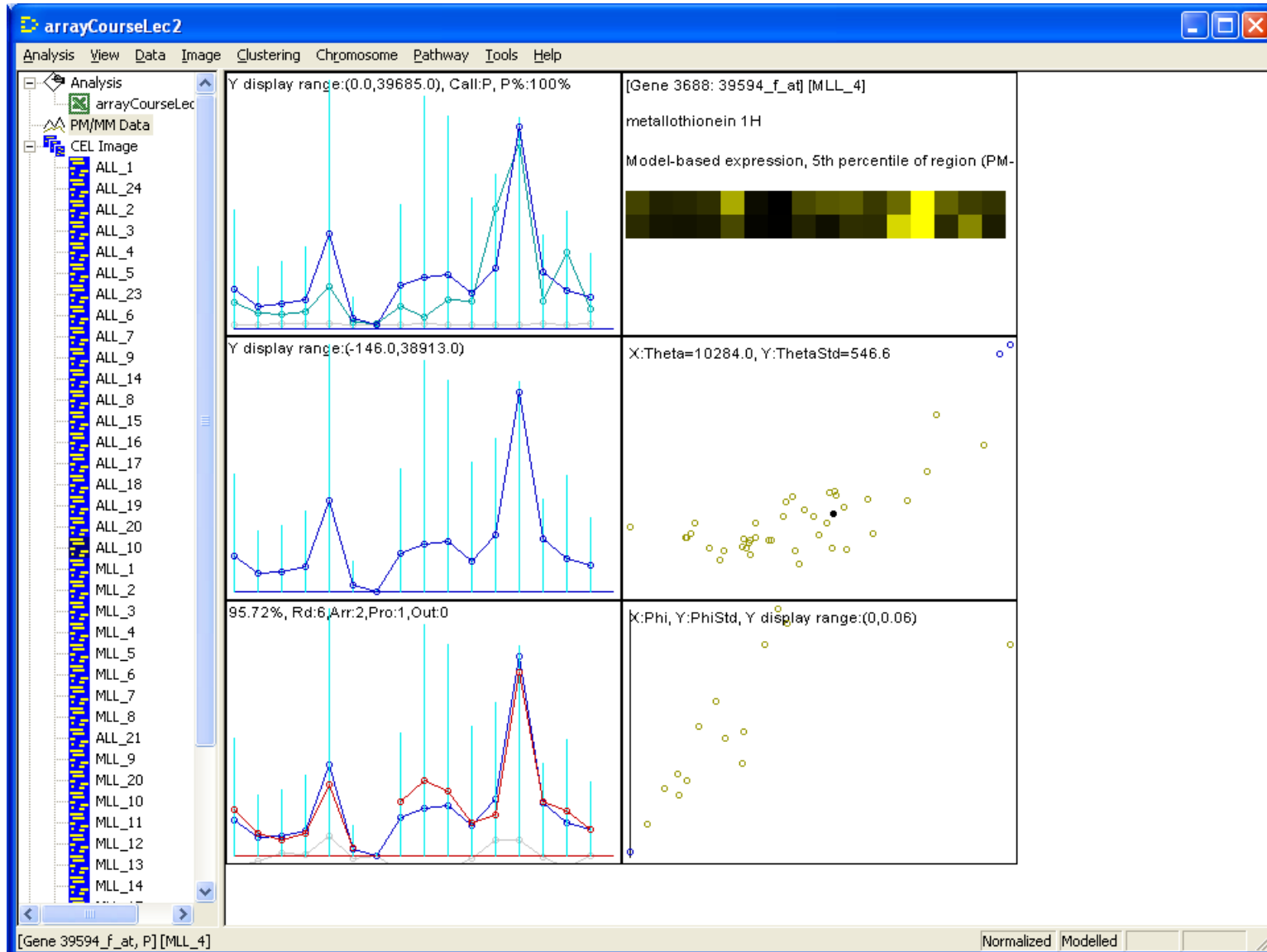
arrayCourseLec2_array_summary.xls

has been updated in the model-fitting process to record the percent of “array outliers” (high standard errors, in white) and “single outliers” (discounted measurements, in purple). Model fitting is performed in a robust fashion. ALL_10 has $> 10\%$ array outliers.



So, what does a probeset look like?

Look at a Probeset



Look at a Probeset

Various panels show

- The PM/MM/BG values for this probeset in this array
- A heatmap view of the same thing
- The target PM-MM values or PM-BG values in this array
- The MBEI values, plotted against their standard errors
- The target values, fitted values, and residuals
- The probe sensitivity indices, plotted against their standard errors

outliers are indicated with colored dots.

Look at a Probeset

Cycling through the different chips can be accomplished using the Page Up or Page Down keys. The arrow keys zoom in and out as before, but this feature is less useful here.

Holding down the Page Down key produces an animation effect, which can also be achieved using Data/Animate.

The samples are sorted in order of increasing MBEI values, so cycling through produces a differential effect.

For the sample in question, there were 2 array outliers, 1 probe outlier, and 0 single outliers. The model explained 95.72% of the variation, and iterative fitting took 6 rounds.



So, which probesets are “interesting”?

Find Interesting Genes

Go to Analysis/Compare Samples

Choose the groups using “Select by Category”; this exploits the information that we supplied in the Sample Info file.

One group is “Baseline”, the other “Experiment”

Filter using the lower bound of fold change

Filter on absolute differences

Find Interesting Genes: Panel 1

Compare Samples

Compare samples | Combine comparisons

Baseline (B)

- MLL_12 [33]
- MLL_13 [34]
- MLL_14 [35]
- MLL_15 [36]
- MLL_19 [37]
- ALL_11 [38]
- ALL_22 [39]
- MLL_18 [40]
- ALL_12 [41]

Experiment (E)

- MLL_10 [31]
- MLL_11 [32]
- MLL_12 [33]
- MLL_13 [34]
- MLL_14 [35]
- MLL_15 [36]
- MLL_19 [37]
- ALL_11 [38]
- ALL_22 [39]

Select by category

Select by category

Comparison criteria

(1) E / B > 1.2 or B / E > 1.2

Use lower 90% confidence bound of fold change

(2) E - B > 100 or B - E > 100

For logged data, use (2) instead of (1) for fold

(3) (P value for testing E = B) <= 0.05

(4) P call of B >= 20 % and P call of E >= 20 %

(5) (P value for paired t-test) <= 0.05

[Help](#)

OK Cancel Apply

Find Interesting Genes

Look at “Combine Comparisons”

See where the comparison results will be sent

Estimate FDR using permutations

Find Interesting Genes: Panel 2

Compare Samples [X]

Compare samples | Combine comparisons

Combine type

And And not
 Or Or not

Combine	Baseline	Experiment	E/...	or B/E>	U...	E-B>	or B
{	1,2,3,4...	20,21,22,23...	1...	1.200	L...	10...	100.
}							

Compare on gene list:

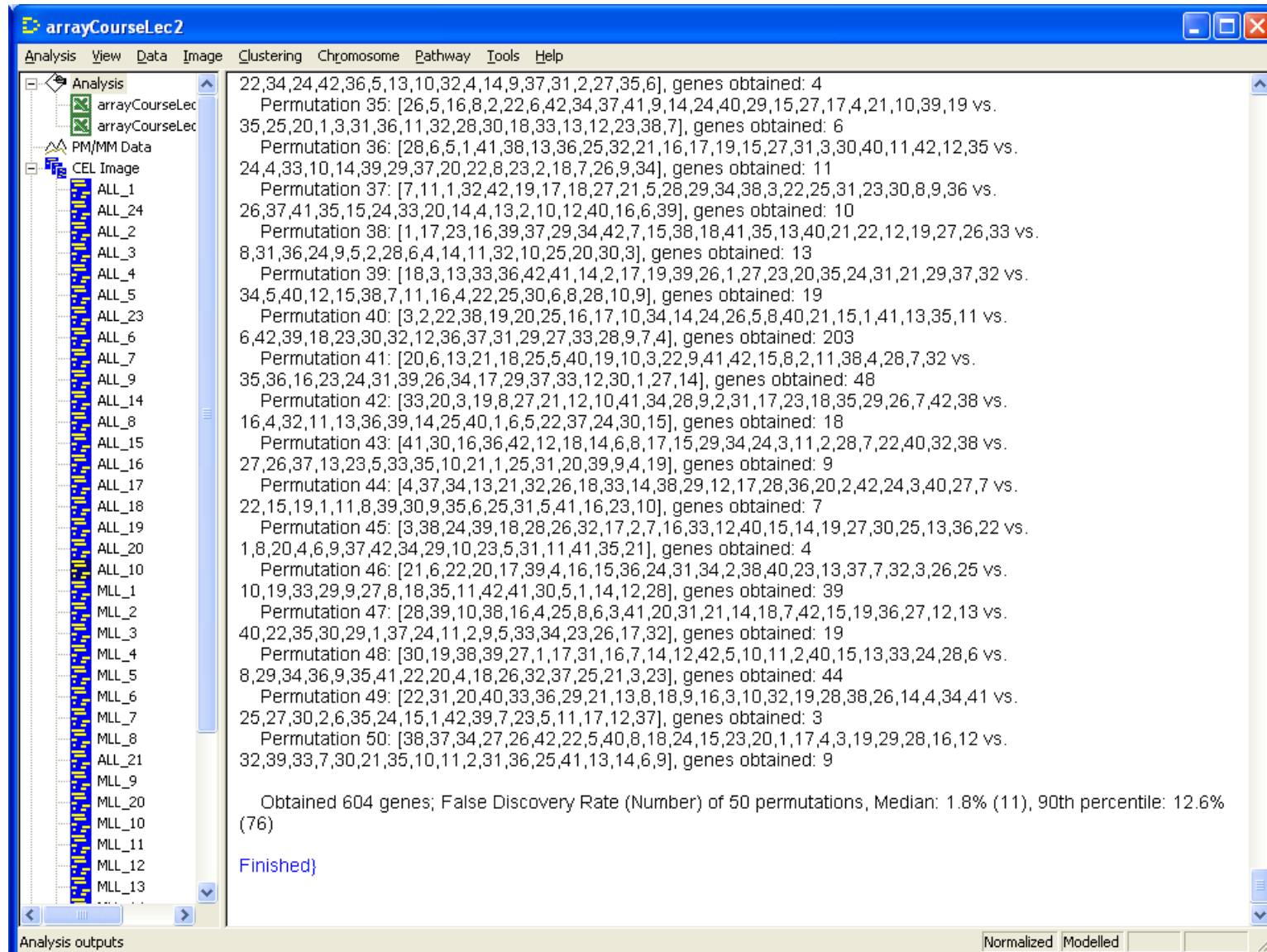
Compare result file

Output all genes Output expression values

Permute samples to assess False Discovery Rate (FDR): times

[Help](#)

Find Interesting Genes – Voila!



Find Interesting Genes

Results are exported to

arrayCourseLec2_compare_result.xls

```
[COMPARE_CRITERIA_V2]  
$NUM_OPTION_LINE=5  
$ARRAY_LIST_FILE=  
$COMPARE_ON_GENE_LIST=  
$COMPARE_ON_USE_LIST=1  
$AVERAGE_USING_STANDARD_ERROR=Yes  
$OMIT_AFFY_CONTROL_GENE=Yes  
$NUM_CRITERION=1
```

More compare result.xls (1)

```

$Parenthesis : Combine : Baseline : Experiment :
  E/B>      : or B/E>      : Use      : E-B>      : or B-E>
  P value <= : P call % of B >= :
  and P call % of E >= : % Pair P value <=
No : and : 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,
    16,17,18,19,28,38,39,41,42 :
    20,21,22,23,24,25,26,27,29,30,31,32,33,34,35,
    36,37,40 :
    1.200 : 1.200 : Lower Bound : 100.000 : 100.000
NA      : NA      : NA      : NA

```

More compare result.xls (2)

[COMPARE_RESULT]

```
probe set      : gene      : Accession      : LocusLink
Description    : ALL_1    ALL_24    ALL_2    ... :
baseline mean :
MLL_1    MLL_2    MLL_3    MLL_4    ... :
experiment mean :
fold change : lower bound of FC : upper bound
of FC : difference of means : filtered
```

More compare result.xls (3)

```
31407_at : protease, serine, 7 (enterokinase) :  
: U09860 : 5651 :  
Cluster Incl. U09860:Human enterokinase mRNA,  
complete cds /cds=(40,3099) /gb=U09860 /gi=  
746412 /ug=Hs.158333 /len=3696 :  
988.74 158.31 296.43 76.82 427.5 ... :  
256.93 :  
100.29 64.72 157.82 111.28 110.88 ... :  
128.5 :  
-2.15 : -1.28 : -3.09 : -148.05 : *
```


Find This Gene

in Probeset View, use View/Find Gene

Find probe set or gene [X]

Change one of the following to search:

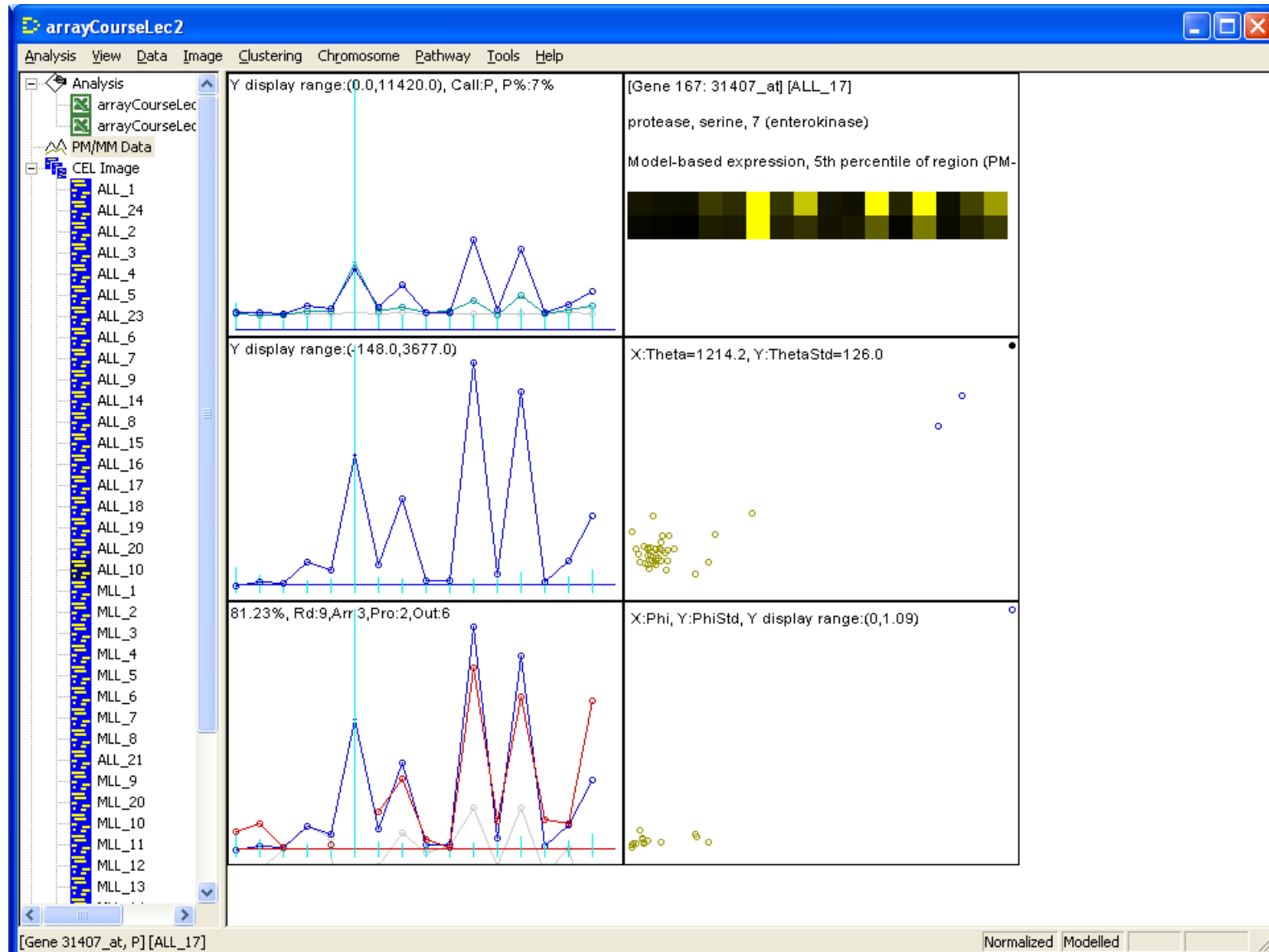
Probe set number:

Probe set name:

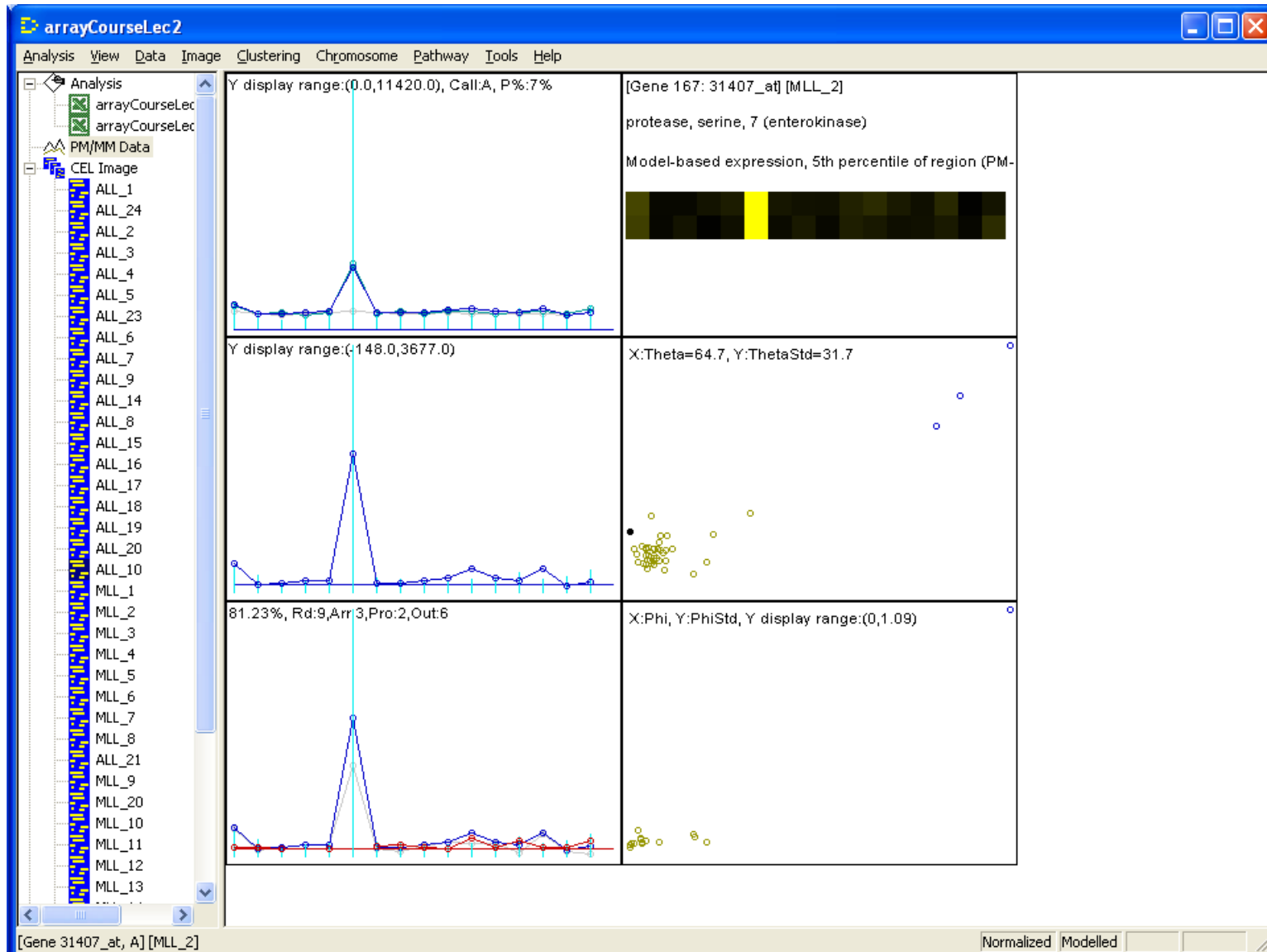
Gene keywords:

(Can use wildcards, e.g. "receptor * kinase" matches with "receptor tyrosine kinase", and both "receptor [1-9]" and "receptor ?" match with "receptor 4")

Find This Gene: ALL_17, High End

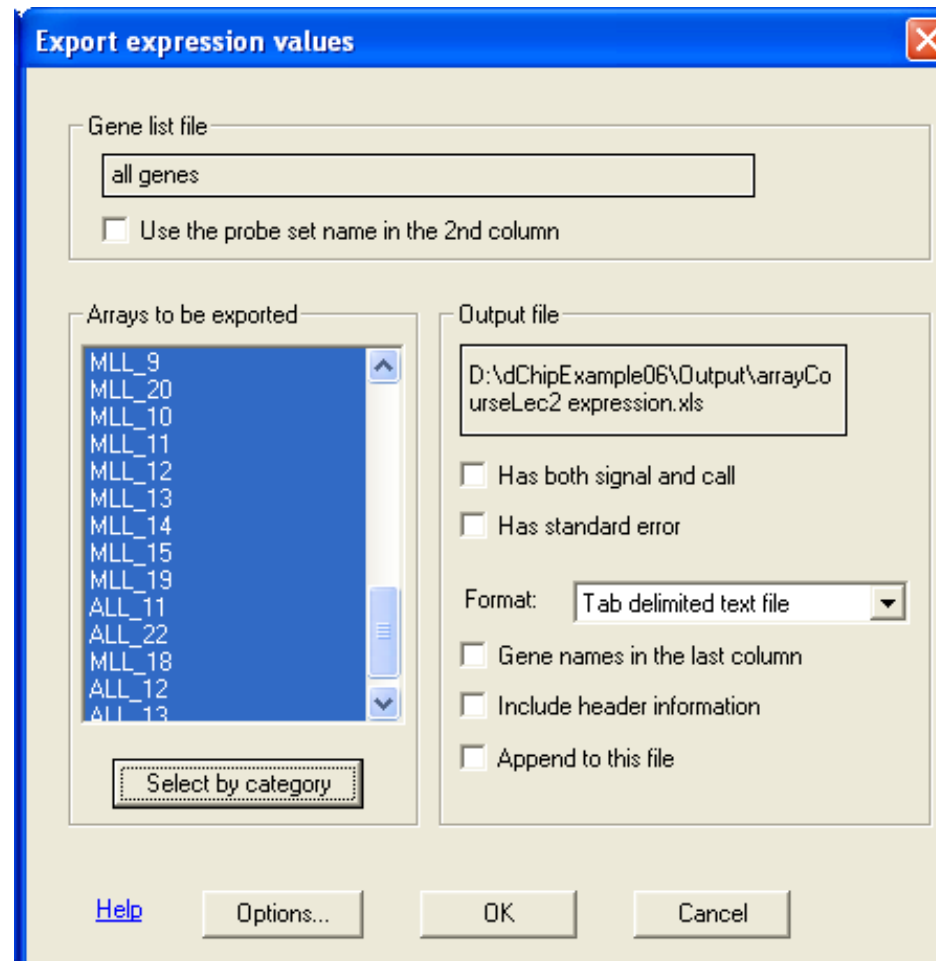


Find This Gene: MLL_2, Low End



Other Exports: Expression Results

Tools/Export Expression Value...



Export all Expression Results (2)

produces arrayCourseLec2_signal.xls

```

probe set      gene      Accession      LocusLink
Description    ALL_1      ALL_24         ALL_2
ALL_3          ALL_4      ALL_5          ALL_23        ALL_6         ALL_7
...
AFFX-MurIL2_at M16762 Mouse interleukin 2 (IL-2)
M16762          M16762 Mouse interleukin 2 (IL-2)
1324.22 1766.49 1562.23 1739.9 1486.82
1624.63 1759.31 1763.18 1558.21 1555.06
...
AFFX-MurIL10_at interleukin 10 M37897 16153
M37897 Mouse interleukin 10 mRNA, complete cds
917.868 1360.26 1067.69 1380.64 1037.5 1074.34
1294.49 1109.37 1181.09 1090.53 1121.5

```

Other Exports: Probe Results

Tools/Export Probe Set...

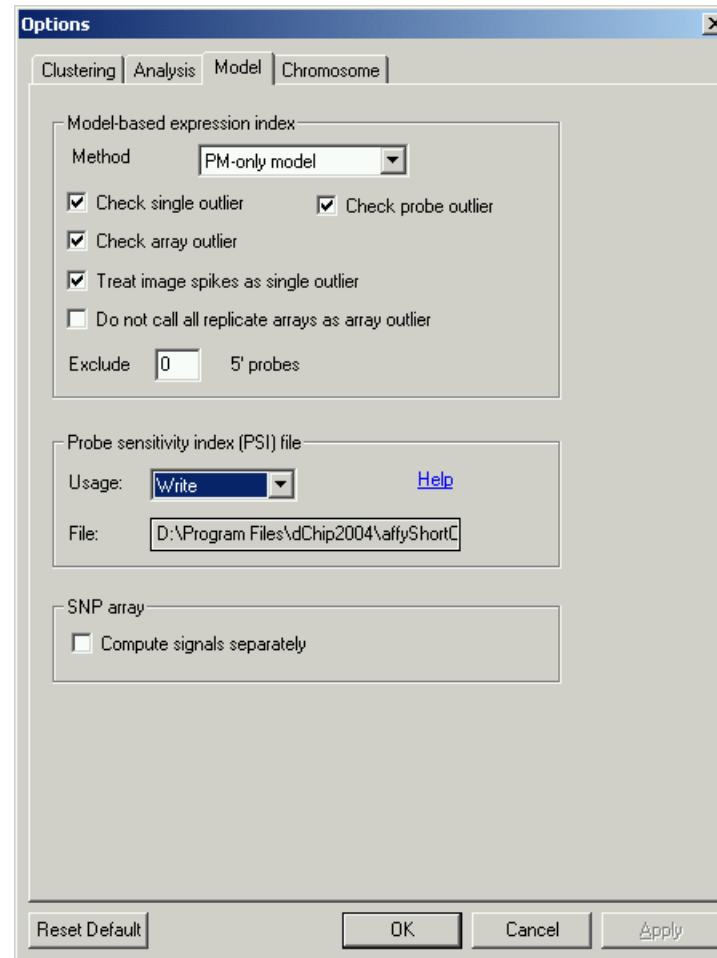
produces

arrayCourseLec2 31407_at probe data.xls

Probeset	Probe	Array	PM	MM	Bkgrd
Theta	Theta_Std	Phi		PhiStd	
31407_at	0	0	985	805	842
988.743	85.1642	0.221123		0.121287	
31407_at	0	1	976	786	812
158.308	29.8064	0.221123		0.121287	

Other Exports: PSIs

Keep the PSIs? Analysis/Normalize & Model, Options, Usage: Write



So, Did We Find What They Did?

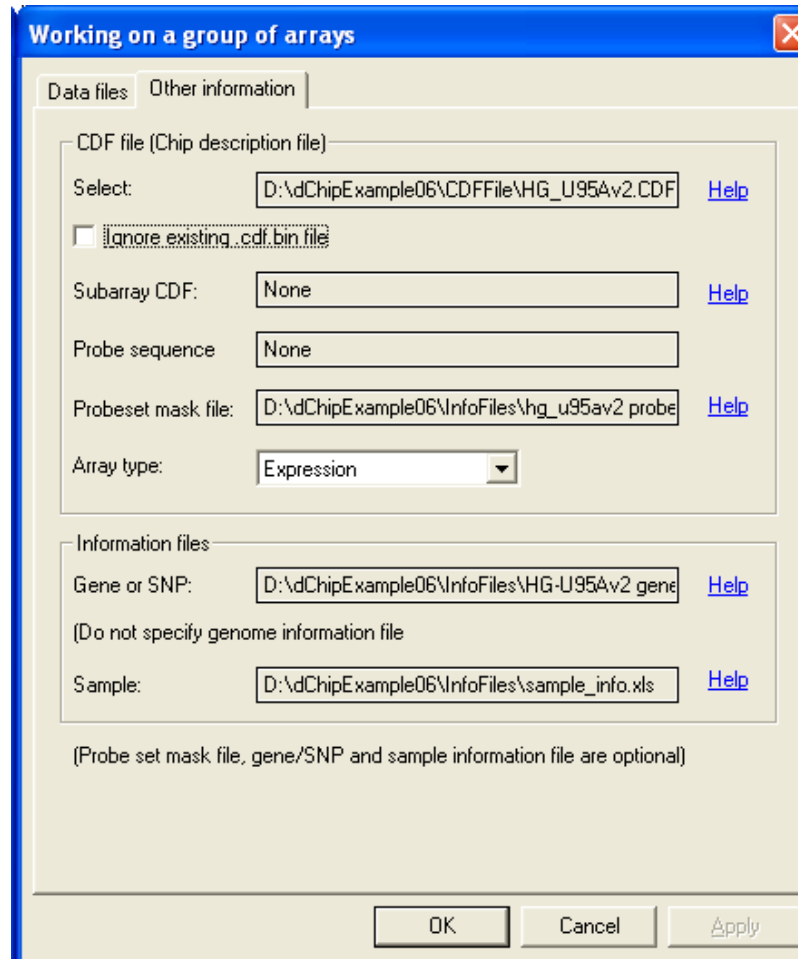
Well...■

It turns out that half of the chips used were U95A, and the rest (including all of the AML samples) were U95Av2. By default, dChip does not combine results from different chip types. However, since the difference is not large (25 probesets out of 12625), we can mask the ones that don't overlap and get it to fit anyway.

try `http://biosun1.harvard.edu/complab/dchip/combine%20chip.htm`

(look at the bottom of the page)

Combine the Chip Types



the mask file is from the dChip web site, and we use the U95Av2 CDF file, recomputing all dcp files.

Do We Find What They Did Now?

Well...■

It turns out that the paper reported gene names and gene symbols, but did not specify the Affymetrix probe ids. Unfortunately, some of the annotation has changed over time.

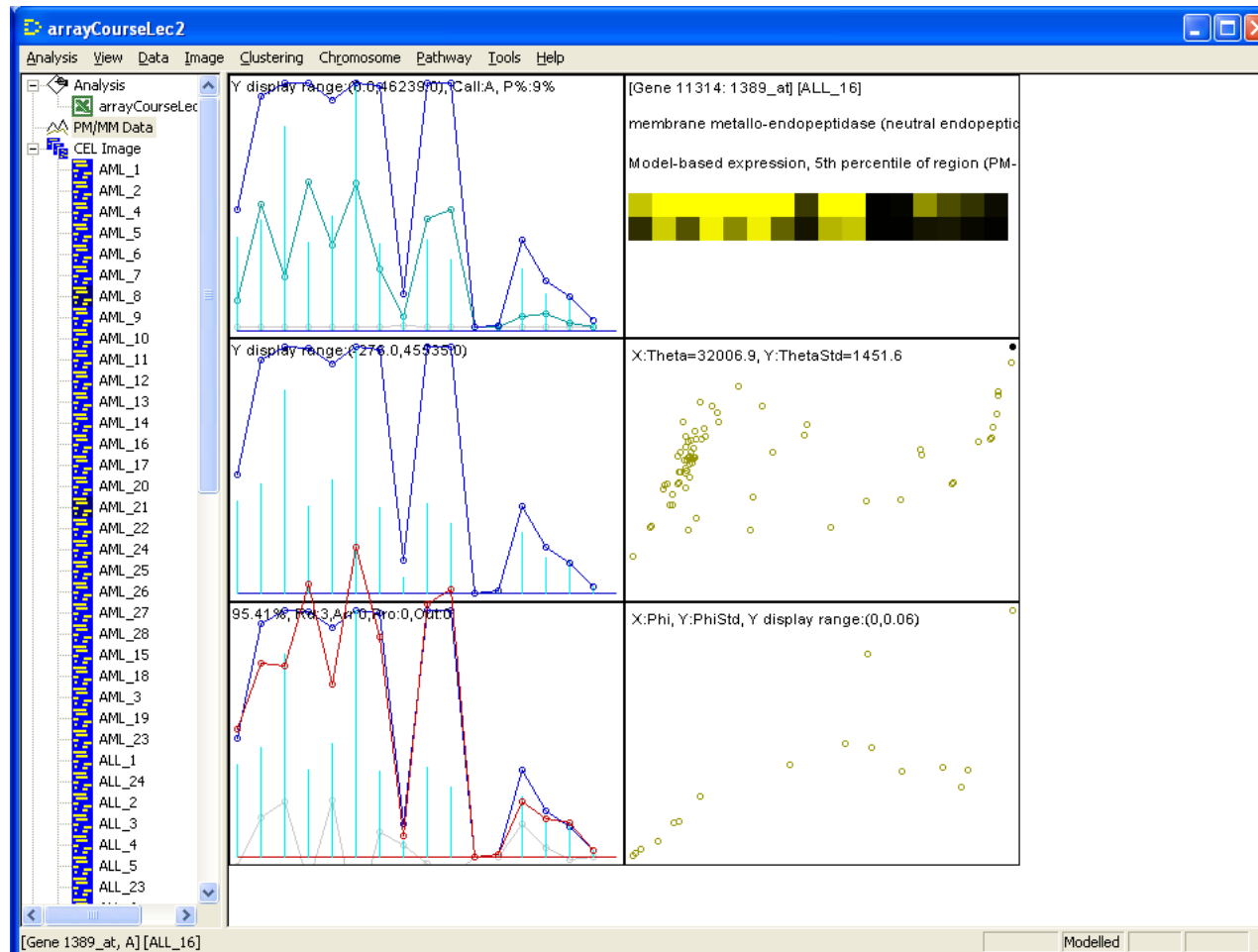
If we look for

J03779 (gene accession number), aka
CD10 (gene symbol)

(high in ALL, low in MLL) in the expression tables supplied with the paper, it's not there. ■ But if we look in the gene info files supplied with dChip, it *is* there (it's 1389_at).

And?

FC: -3.87, CI: (-3.32,-4.46), Diff: -16956.1. Different!



One Last Step

Analysis/Save Log

Summary

We know what files to track down

We know how to load them in for processing

We know how to normalize and fit models

We know how to export results

We've seen how finicky indexing can be.



And we struck biology!

Thus endeth the lesson...