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

18 October 2005

Lecture 13: Microarrays in R: Start to Finish

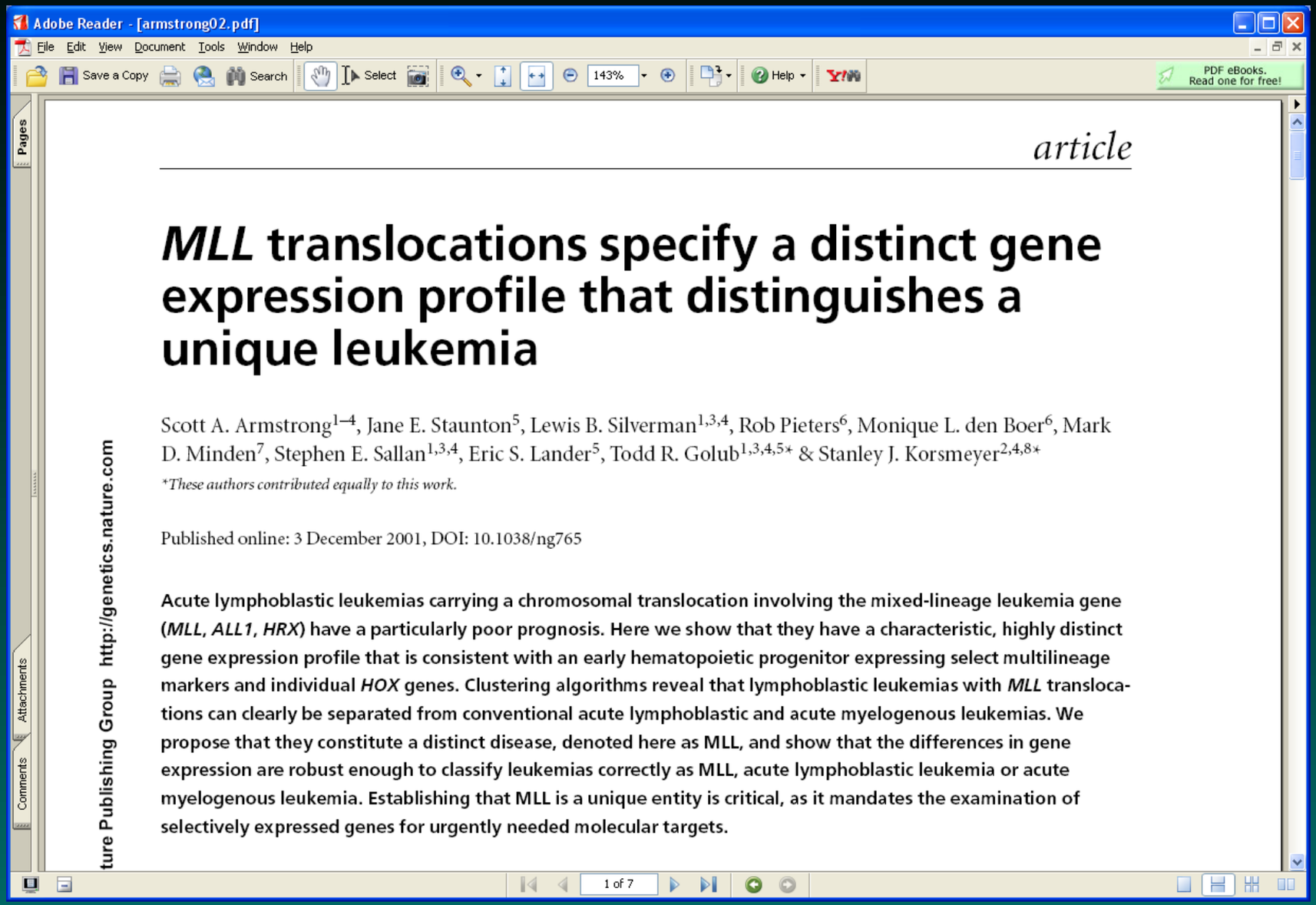
- Source of the Data Set
- Understanding the Sample Information
- First Pass Using dChip
- Starting in R
- Just RMA

Source of the Data Set

In today's lecture, we're going to perform a complete start-to-finish analysis of a microarray data set. We have chosen to use a leukemia data set that we looked at briefly in an earlier lecture. The data set consists of U95A microarray experiments on

1. 24 patients with acute lymphocytic leukemia (ALL)
2. 28 patients with acute myeloid leukemia (AML)
3. 20 patients with mixed lineage leukemia (MLL)

Armstrong, Nat Genet, 2002; 30:41-47



Results reported in the paper

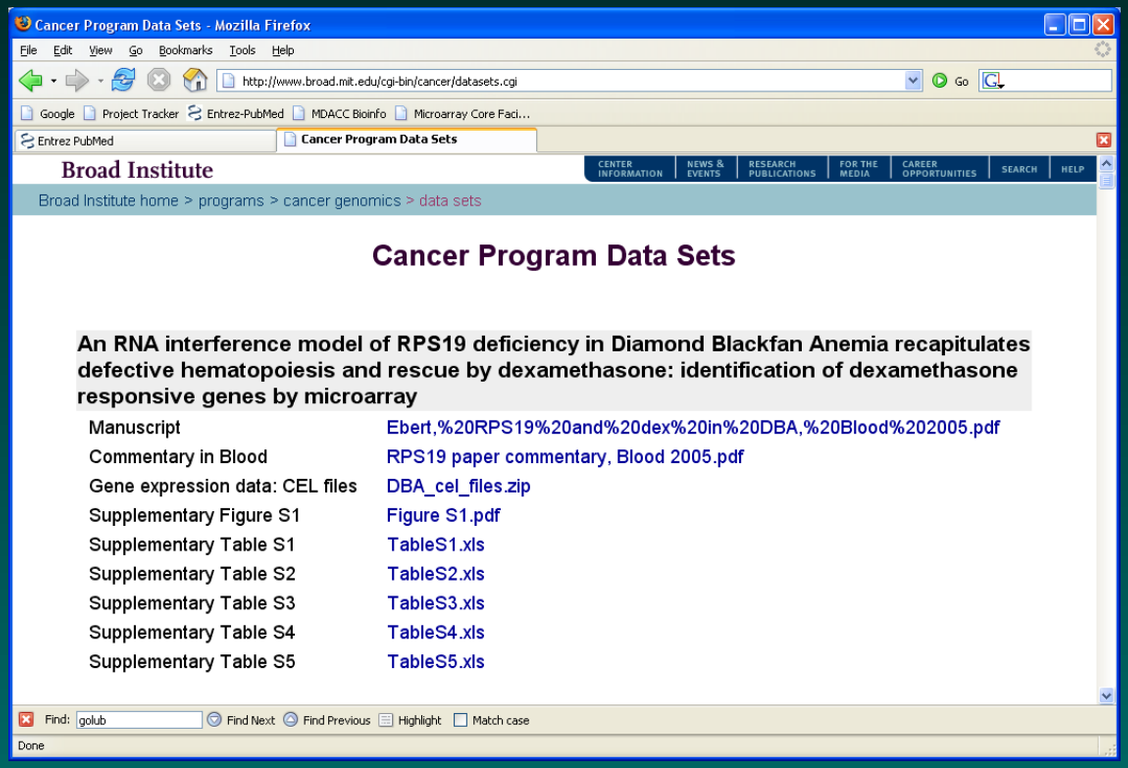
- MLL is distinct from conventional ALL
  - Based on arrays, $\sim$1200 differentially expressed genes
- MLL shows multilineage gene expression
  - Looking at the expressed genes, find some from B cells and some from myeloid cells
- MLL is arrested at an early stage of hematopoiesis
- Some HOX genes are overexpressed in MLL

Results reported in the paper

- MLL is distinct from AML as well as from conventional ALL
  - Principal components analysis
  - Selected genes using one-versus-all comparisons
- Gene expression profiles correctly classify ALL, AML, MLL
  - K-nearest-neighbors (KNN) predictor

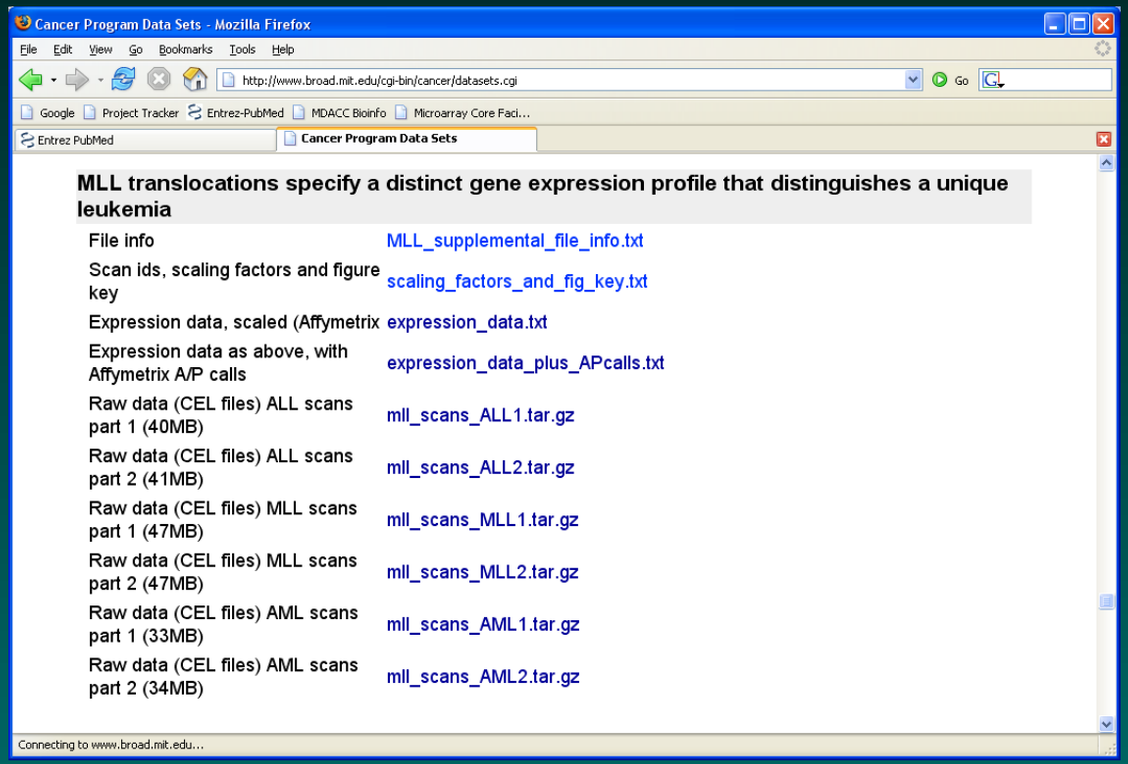
Getting the data

Note: The links to supplementary data listed in the original paper no longer work (as of the morning of 18 October 2005). However, the data is available at: `http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi`



Getting the data

Scroll down (or search for “translocations”) to find the data set. You’ll need all six collections of CEL files along with the “scaling_factors_and_fig_key.txt” that contains the sample information.



Understanding the Sample Information

Microsoft Excel - scaling_factors_and_fig_key.txt

File Edit View Insert Format Tools Data Window RExcel Help

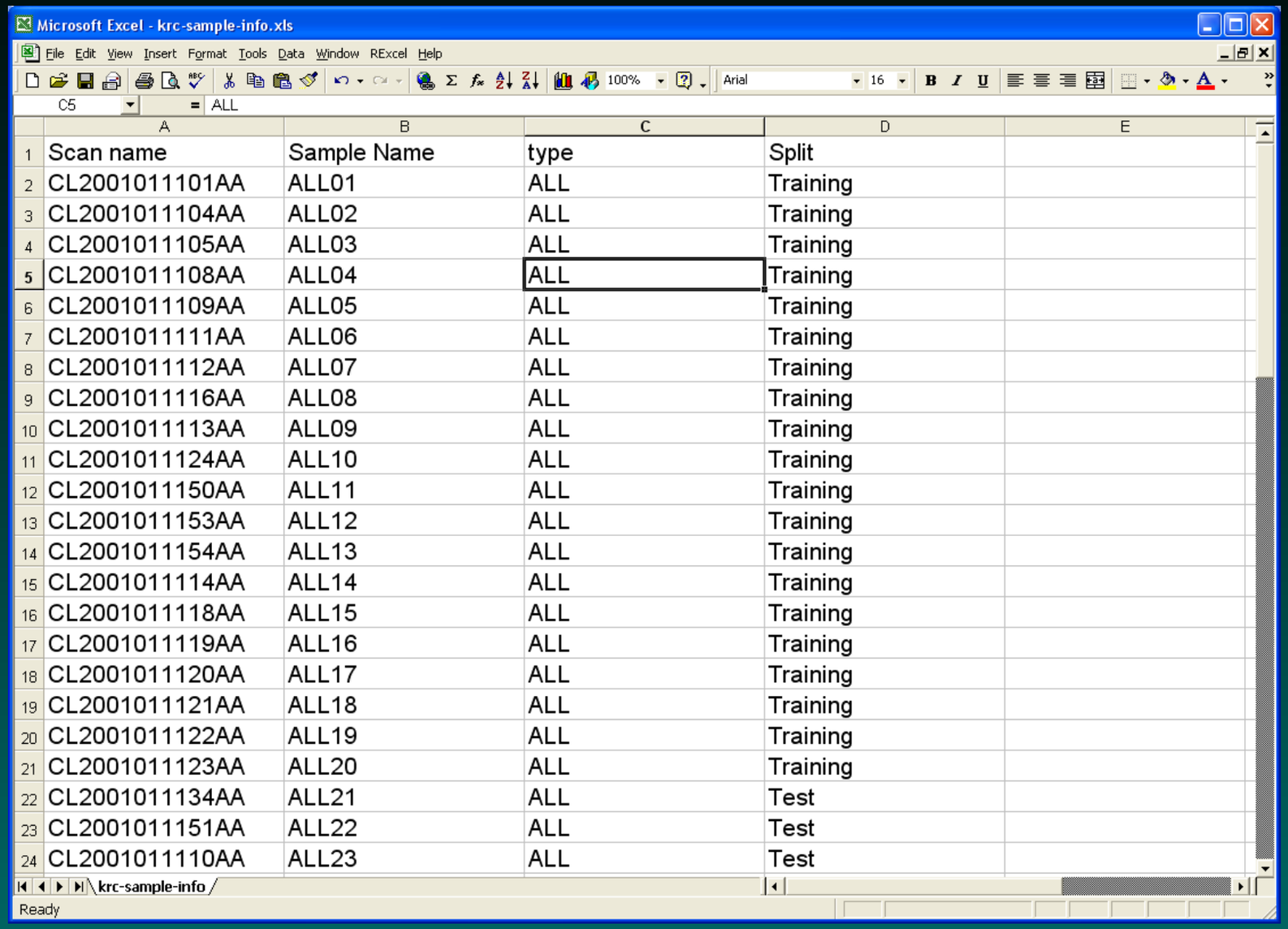
E1 =

	A	B	C	D	E
1	sample_name	scan name	linear scaling factor	paper_figure	
2	ALL_1	CL2001011101AA	1	TRAIN/figs.1,4,5	
3	ALL_2	CL2001011104AA	0.9399	TRAIN/figs.1,4,5	
4	ALL_3	CL2001011105AA	1.6781	TRAIN/figs.1,4,5	
5	ALL_4	CL2001011108AA	1.0635	TRAIN/figs.1,4,5	
6	ALL_5	CL2001011109AA	1.3875	TRAIN/figs.1,4,5	
7	ALL_6	CL2001011111AA	1.1869	TRAIN/figs.1,4,5	
8	ALL_7	CL2001011112AA	1.1951	TRAIN/figs.1,4,5	
9	ALL_8	CL2001011116AA	1.2615	TRAIN/figs.1,4,5	
10	ALL_9	CL2001011113AA	1.5606	TRAIN/figs.1,4,5	
11	ALL_10	CL2001011124AA	1.2855	TRAIN/figs.1,4,5	
12	ALL_11	CL2001011150AA	1.1064	TRAIN/figs.1,4,5	
13	ALL_12	CL2001011153AA	1.2399	TRAIN/figs.1,4,5	
14	ALL_13	CL2001011154AA	1.4928	TRAIN/figs.1,4,5	
15	ALL_14	CL2001011114AA	1.0762	TRAIN/figs.1,4,5	
16	ALL_15	CL2001011118AA	1.3057	TRAIN/figs.1,4,5	
17	ALL_16	CL2001011119AA	1.1453	TRAIN/figs.1,4,5	
18	ALL_17	CL2001011120AA	1.1352	TRAIN/figs.1,4,5	
19	ALL_18	CL2001011121AA	1.1639	TRAIN/figs.1,4,5	
20	ALL_19	CL2001011122AA	1.2322	TRAIN/figs.1,4,5	
21	ALL_20	CL2001011123AA	1.2835	TRAIN/figs.1,4,5	
22	ALL_21	CL2001011134AA	1.1707	test/figs 4,5	
23	ALL_22	CL2001011151AA	1.2464	test/figs 4,5	
24	ALL_23	CL2001011110AA	1.3895	test/figs 4,5	

scaling_factors_and_fig_key

Ready

Including critical factors



Microsoft Excel - krc-sample-info.xls

File Edit View Insert Format Tools Data Window RExcel Help

C5 = ALL

	A	B	C	D	E
1	Scan name	Sample Name	type	Split	
2	CL2001011101AA	ALL01	ALL	Training	
3	CL2001011104AA	ALL02	ALL	Training	
4	CL2001011105AA	ALL03	ALL	Training	
5	CL2001011108AA	ALL04	ALL	Training	
6	CL2001011109AA	ALL05	ALL	Training	
7	CL2001011111AA	ALL06	ALL	Training	
8	CL2001011112AA	ALL07	ALL	Training	
9	CL2001011116AA	ALL08	ALL	Training	
10	CL2001011113AA	ALL09	ALL	Training	
11	CL2001011124AA	ALL10	ALL	Training	
12	CL2001011150AA	ALL11	ALL	Training	
13	CL2001011153AA	ALL12	ALL	Training	
14	CL2001011154AA	ALL13	ALL	Training	
15	CL2001011114AA	ALL14	ALL	Training	
16	CL2001011118AA	ALL15	ALL	Training	
17	CL2001011119AA	ALL16	ALL	Training	
18	CL2001011120AA	ALL17	ALL	Training	
19	CL2001011121AA	ALL18	ALL	Training	
20	CL2001011122AA	ALL19	ALL	Training	
21	CL2001011123AA	ALL20	ALL	Training	
22	CL2001011134AA	ALL21	ALL	Test	
23	CL2001011151AA	ALL22	ALL	Test	
24	CL2001011110AA	ALL23	ALL	Test	

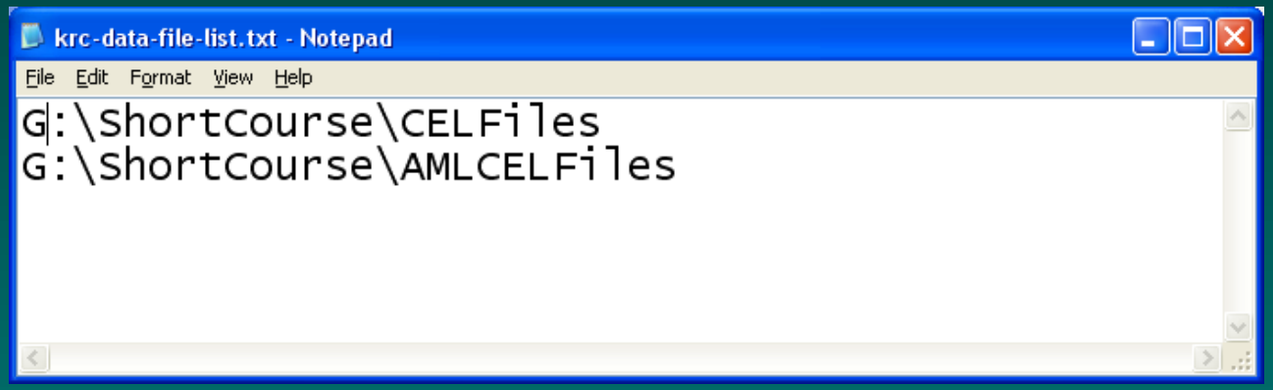
krc-sample-info

Ready

First Pass Using dChip

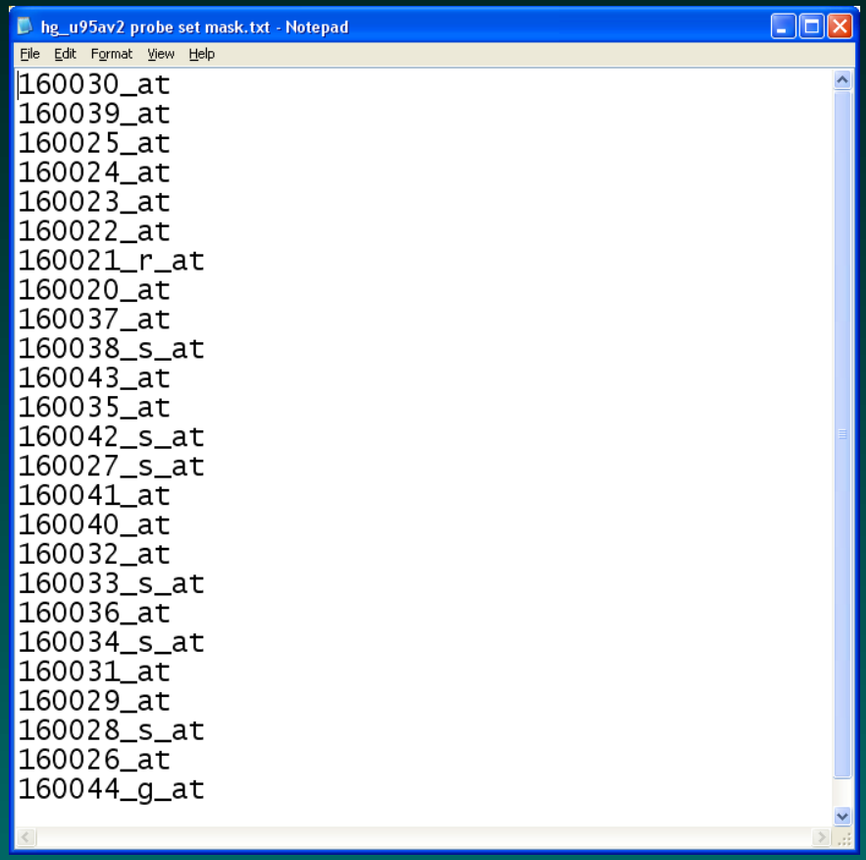
Early in the course, we used this data set in dChip. One of the things we discovered was that the samples were run on two different iterations of the U95A GeneChip. The ALL and MLL samples were run on version 1 of the U95A and the AML samples were run on the U95Av2 chip.

Our first step in dealing with this issue was to install the CEL files into two different directories. In dChip, we handle this by first producing a “data file list” that tells dChip which directories to look into:



Probe Set Mask File

We also have top produce a “probe set mask” file that tells dChip which probe sets to ignore (since they changed from one version of the chip to the next).



dChip analysis

1. Load the files into dChip using
   - The sample information file
   - The data file list
   - The U95Av2 CDF file
   - The probe set mask
2. Normalize to array ALL21 (median brightness)
3. Quantify using the PM-only model

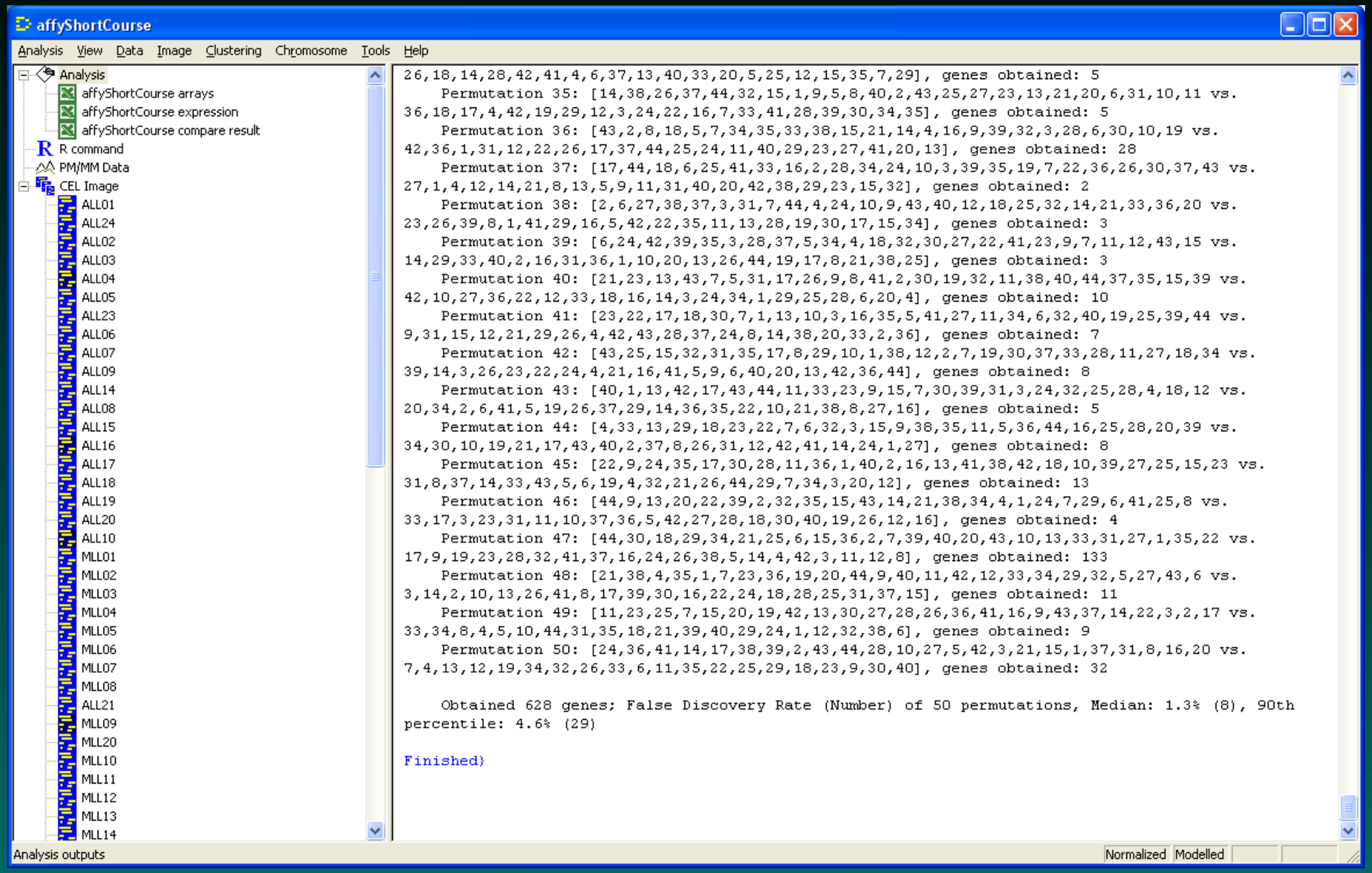
dChip quality check

- dChip flags several arrays as outliers:
  - ALL10, which has the highest overall brightness
  - MLL09, which has the lowest overall broightness
  - ALL04, ALL16, AML08

Microsoft Excel - affyShortCourse arrays.xls

Number	Array	File Name	Median Intensity	P call %	% Array outlier	% Single outlier	Warning
1	ALL01	g:\ShortCourse\CELF	1497	48.2	1.648	0.099	
2	ALL24	g:\ShortCourse\CELF	1196	38.3	3.778	0.432	
3	ALL02	g:\ShortCourse\CELF	1778	49.5	1.450	0.144	
4	ALL03	g:\ShortCourse\CELF	1097	36.8	3.152	0.375	
5	ALL04	g:\ShortCourse\CELF	1489	38.7	5.299	0.216	*
6	ALL05	g:\ShortCourse\CELF	1573	46.6	0.855	0.120	
7	ALL23	g:\ShortCourse\CELF	1436	46.3	0.467	0.064	
8	ALL06	g:\ShortCourse\CELF	1410	47.5	0.554	0.092	
9	ALL07	g:\ShortCourse\CELF	1242	49.8	1.933	0.164	
10	ALL09	g:\ShortCourse\CELF	1122	43.8	1.719	0.121	
11	ALL14	g:\ShortCourse\CELF	1783	45.7	1.703	0.108	
12	ALL08	g:\ShortCourse\CELF	1599	46.0	1.370	0.264	
13	ALL15	g:\ShortCourse\CELF	1499	34.8	1.093	0.228	
14	ALL16	g:\ShortCourse\CELF	1310	42.4	6.384	0.256	*
15	ALL17	g:\ShortCourse\CELF	2280	37.3	1.077	0.202	
16	ALL18	g:\ShortCourse\CELF	1216	43.5	2.210	0.166	
17	ALL19	g:\ShortCourse\CELF	1438	47.4	1.711	0.459	
18	ALL20	g:\ShortCourse\CELF	1632	44.7	0.919	0.558	
19	ALL10	g:\ShortCourse\CELF	3097	36.8	8.729	0.952	*
20	MLL01	g:\ShortCourse\CELF	1280	46.6	0.808	0.219	
21	MLL02	g:\ShortCourse\CELF	1169	44.6	2.503	0.313	
22	MLL03	g:\ShortCourse\CELF	1669	46.5	0.333	0.190	
23	MLL04	g:\ShortCourse\CELF	1173	39.4	0.531	0.118	

dChip Comparison: ALL vs MLL



628 differentially expressed genes

Microsoft Excel - affyShortCourse compare result.xls

File Edit View Insert Format Tools Data Window RExcel Help

BB50 = -3.79

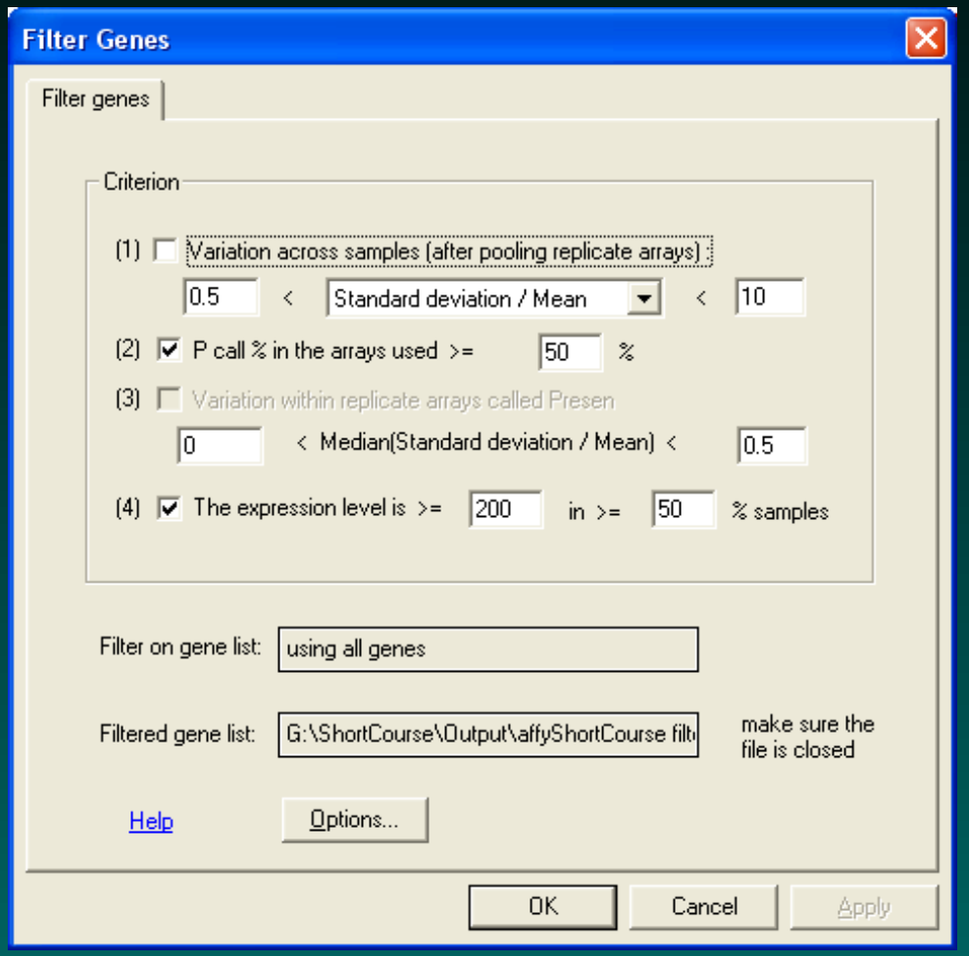
	A	B	AD	AZ	BB	BC	BD	BE
13	probe set	gene	baseline mean	experiment mean	fold change	lower bound	upper bound	difference
14	37680_at	A kinase (PRKA) ar	2991.81	147.65	-20.26	-13.19	-29.88	-2844.16
15	37280_at	MAD, mothers agai	9355.77	697.15	-13.42	-9.62	-17.36	-8658.62
16	1325_at	MAD, mothers agai	7714.68	617.24	-12.50	-8.54	-17.12	-7097.44
17	1488_at	protein tyrosine pho	4121.6	579.89	-7.11	-3.86	-10.50	-3541.71
18	34194_at	Homo sapiens mRN	1420.54	200.61	-7.08	-3.32	-13.62	-1219.93
19	1077_at	recombination activ	6897.9	982.66	-7.02	-4.26	-11.30	-5915.24
20	753_at	nidogen 2 (osteonic	2389.96	342.13	-6.99	-2.78	-11.31	-2047.83
21	37908_at	guanine nucleotide	6909.7	1035.23	-6.67	-4.15	-11.67	-5874.47
22	35614_at	transcription factor-	7521.88	1216.05	-6.19	-4.07	-8.59	-6305.83
23	34800_at	leucine-rich repeats	5226.23	844.98	-6.19	-4.00	-9.80	-4381.25
24	31892_at	protein tyrosine pho	868.91	145.9	-5.96	-2.01	-10.04	-723.00
25	41266_at	integrin, alpha 6	7952.9	1466.32	-5.42	-3.85	-7.47	-6486.57
26	31786_at	KH domain containi	2488.25	472.29	-5.27	-3.29	-9.34	-2015.95
27	38408_at	transmembrane 4 s	6455.32	1264.74	-5.10	-3.47	-7.59	-5190.58
28	36650_at	cyclin D2	10001.23	2021.61	-4.95	-3.47	-7.76	-7979.62
29	38578_at	tumor necrosis fact	4050.73	828.6	-4.89	-3.33	-7.03	-3222.13
30	32778_at	inositol 1,4,5-tripho	2099.86	433.92	-4.84	-3.61	-6.41	-1665.94
31	40570_at	forkhead box O1A (	10307.45	2130.08	-4.84	-3.37	-7.52	-8177.37
32	39878_at	protocadherin 9	12520.62	2670.21	-4.69	-3.07	-7.31	-9850.42
33	31886_at	5'-nucleotidase, ec	2593.49	567.74	-4.57	-2.82	-7.59	-2025.76
34	41690_at	Homo sapiens mRN	5279.81	1166.43	-4.53	-3.01	-6.35	-4113.38
35	33386_at	H1 histone family, r	6216.48	1381.24	-4.50	-2.46	-10.81	-4835.24
36	37780_at	piccolo (presynapti	2867.65	638.28	-4.49	-2.35	-6.71	-2229.37

affyShortCourse compare result

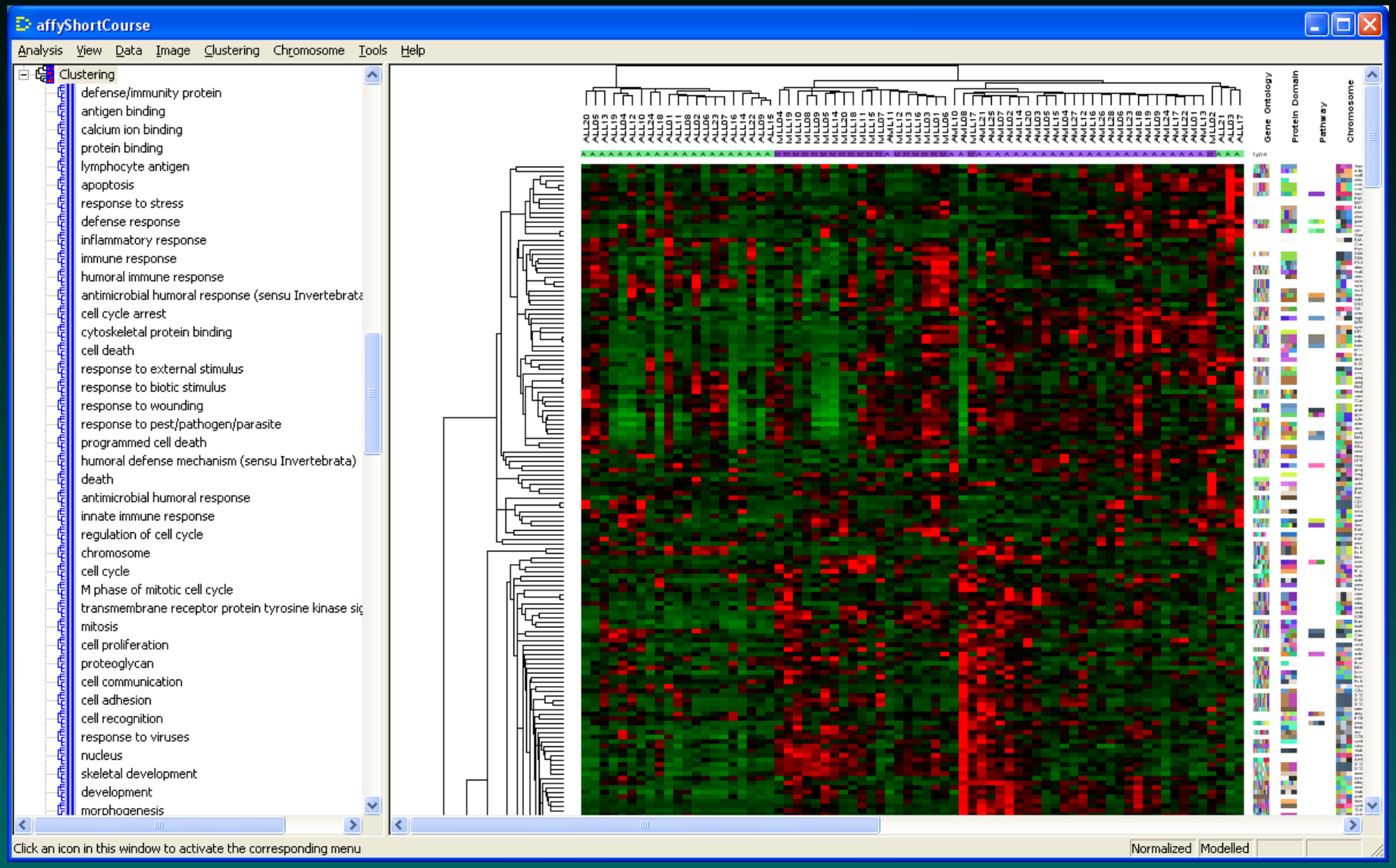
Ready

Filter genes

Filter based on expression level and on percent present calls.

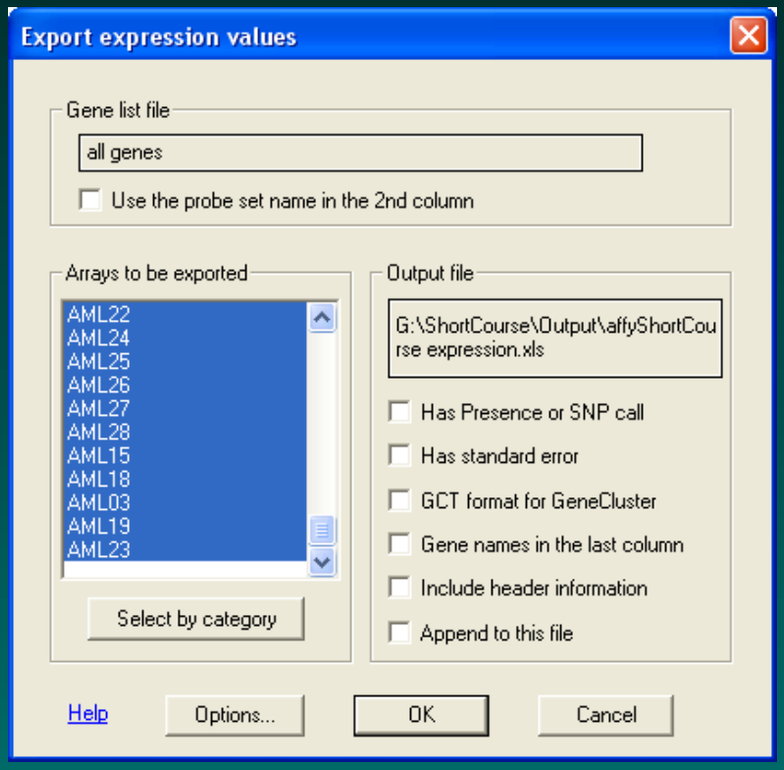


Cluster samples



Exporting all the data from dChip

1. Use menu “Tools” $->$ “Export Expression Value”.
2. Select “all genes”
3. Press “OK”



Starting in R

Load the `affy` package.

```
> require(affy)
Loading required package: affy
Loading required package: Biobase
Loading required package: tools
Welcome to Bioconductor
 Vignettes contain introductory material.  To view,
 simply type: openVignette()
 For details on reading vignettes, see
 the openVignette help page.
Loading required package: reposTools
[1] TRUE
```

Load the Sample Information file

```
> # remember where the data lives
> home <- 'g:/ShortCourse'
> # use the same sample info file we made for dChip
> si <- read.table(file.path(home, 'InfoFiles',
+                            'krc-sample-info.xls'),
+                  header=TRUE, sep='\t')
> si[1:5,]
          Scan.name type    Split
ALL01 CL2001011101AA  ALL Training
ALL02 CL2001011104AA  ALL Training
ALL03 CL2001011105AA  ALL Training
ALL04 CL2001011108AA  ALL Training
ALL05 CL2001011109AA  ALL Training
```

Load dChip's array file

```
> arrays <- read.table(file.path(home, 'Output',
                       'affyShortCourse arrays.xls'),
+                  header=TRUE, as.is=TRUE, sep='\t')
> # Fix the column names!
> dimnames(arrays)[[2]] <-
+     c(dimnames(arrays)[[2]][2:8], 'x')
> # Use the sample name as the row name
> dimnames(arrays)[[1]] <- arrays$Array
> # Only keep useful columns
> arrays <- arrays[, 3:6]
> # Give them sensible names
> dimnames(arrays)[[2]] <- c('MedianIntensity',
+     'PercentPresent', 'ArrayOutlier',
+     'SingleOutlier')
```

Combine sample information

```
> # Merge sample info with dChip info
> si <- merge(si, arrays, by='row.names',
+     sort=FALSE)
> # Sigh. Fix the row names yet again.
> dimnames(si)[[1]] <- si$Row.names
> # Remove redundant columns
> si <- si[, 2:8]
> rm(arrays) # cleanup
```

Note that the order has changed!

```
> si[1:5,]
           Scan.name type    Split MedianIntensity
ALL01 CL2001011101AA  ALL Training             1497
ALL24 CL2001011102AA  ALL     Test             1196
ALL02 CL2001011104AA  ALL Training             1778
ALL03 CL2001011105AA  ALL Training             1097
ALL04 CL2001011108AA  ALL Training             1489
      PercentPresent ArrayOutlier SingleOutlier
ALL01           48.2        1.648         0.099
ALL24           38.3        3.778         0.432
ALL02           49.5        1.450         0.144
ALL03           36.8        3.152         0.375
ALL04           38.7        5.299         0.216
```

Specialized factors

```
> # make a factor to compare ALL vs MLL
> temp <- si$type
> temp[temp=='AML'] <- NA
> si$ALLvMLL <- factor(temp)
>
> temp <- si$type
> temp[temp=='MLL'] <- NA
> si$ALLvAML <- factor(temp)
>
> temp <- si$type
> temp[temp=='ALL'] <- NA
> si$MLLvAML <- factor(temp)
```

```
> temp <- si$type
> temp[temp=='AML'] <- 'Other'
> temp[temp=='MLL'] <- 'Other'
> si$ALLvOther <- factor(temp)
>
> temp <- si$type
> temp[temp=='ALL'] <- 'Other'
> temp[temp=='MLL'] <- 'Other'
> si$AMLvOther <- factor(temp)
>
> temp <- si$type
> temp[temp=='AML'] <- 'Other'
> temp[temp=='ALL'] <- 'Other'
> si$MLLvOther <- factor(temp)
>
> si$type <- factor(si$type)
```

```
> summary(si)
  Scan.name            type        Split
 Length:72          ALL:24   Length:72
 Class :character   AML:28   Class :character
 Mode  :character   MLL:20   Mode  :character

MedianIntensity PercentPresent   ArrayOutlier
Min.   : 804    Min.   :28.30   Min.   : 0.253
1st Qu.:1222    1st Qu.:36.80   1st Qu.: 0.729
Median :1442    Median :41.10   Median : 1.085
Mean   :1483    Mean   :40.46   Mean   : 1.724
3rd Qu.:1727    3rd Qu.:44.83   3rd Qu.: 1.697
Max.   :3097    Max.   :49.80   Max.   :14.337
```

```
SingleOutlier    ALLvMLL    ALLvAML    MLLvAML
Min.   :0.0440   ALL :24    ALL :24    AML :28
1st Qu.:0.1610   MLL :20    AML :28    MLL :20
Median :0.2405   NA's:28    NA's:20    NA's:24
Mean   :0.2741
3rd Qu.:0.3460
Max.   :0.9520

ALLvOther  AMLvOther  MLLvOther
ALL  :24   AML  :28   MLL  :20
Other:48   Other:44   Other:52
```

Create the phenoData object

```
> pd <- new('phenoData', pData=si, varLabels=list(
+     Scan.name='CEL file name',
+     type='Histological classification',
+     Split='Used as training or test,
+     MedianIntensity='Unnormalized median brighness,
+     PercentPresent='Percentage of present calls',
+     ArrayOutlier='Percentage of Array Outliers',
+     SingleOutlier='Percentage of Single Outliers',
+     ALLvMLL='binary classifier',
+     ALLvAML='binary classifier',
+     MLLvAML='binary classifier',
+     ALLvOtherL='binary classifier',
+     AMLvOther='binary classifier',
+     MLLvOther='binary classifier'))
```

Create the MIAME object

MIAME = minimum information about a microarray experiment

Some of the BioConductor routines require a MIAME object, even thought they will let you submit a character string as a description.

```
> miame <- new('MIAME',
+              name='SA Armstrong',
+              lab='Lander-Golub',
+              title='MLL translocations')
```

Read in the data from dChip

```
> temp <- read.table(file.path(home, 'Output',
+             'affyShortCourse expression.xls'),
+           header=TRUE, as.is=TRUE, sep='\t',
+           quote='', comment.char='')
> # expression data in the later columns
> data <- as.matrix(temp[, 6:77])
> # gene identifiers in the first five columns
> gi <- temp[, 1:5]
> # Use probe sets as row names
> dimnames(gi)[[1]]   <- gi$probe.set
> dimnames(data)[[1]] <- gi$probe.set
```

Check that the order agrees

We noticed that the order of entries in the sample info file had changed when we merged it with the dChip array information. Just to be on the safe side, we should make sure that the order of the data columns matches the sample infor rows.

```
> sum(dimnames(si)[[1]] != dimnames(data)[[2]])
[1] 0
> sum(dimnames(si)[[1]] == dimnames(data)[[2]])
[1] 72
```

Turn the dChip data into an exprSet

We can bring the dChip quantifications directly into R and turn them into an `exprSet`. Note that this avoids the memory problems by not brininging in the individual CEL files and not producing an AffyBatch.

```
> dchip <- new('exprSet',
+              exprs=data,
+              phenoData=pd,
+              annotation='hgu95av2',
+              description=miame,
+              notes='processed by KRC in dChip')
> rm(temp, data, si) # cleanup
```

Just RMA

In order to process the data using RMA in BioConductor, we will use the `just.rma` function. This method avoids the memory problems associated with reading all the CEL files into R and keeping them around during processing. Instead, the processing is handed off to a C module that produces an `exprSet` but skips the production of an `AffyBatch` object.

Locating CEL files in multiple directories

The default behavior for the BioConductor routines is to read all the CEL files in the current working directory. If you want to combine files from more than one location (or if you only want to use a subset of the CEL files), then you must first prepare a list of character strings that give the complete names and locations of the files you want.

```
> # CEL file location is a function of type
> cel.location <- list(ALL='CELFiles',
+         MLL='CELFiles', AML='AMLCELFiles')
> # Use the type of each file to find its location
> celdir <- cel.location[as.character(
+     pd@pData$type)]
```

```
> # Paste the '.cel' extension at the end
> celname <- paste(pd@pData$Scan.name,
+    'cel', sep='.')
> # make complete file paths for each file
> all.cel.files <- file.path(home, celdir, celname)
> # peek at the results
> all.cel.files[1:3]
[1] "g:/ShortCourse/CELFiles/CL2001011101AA.cel"
[2] "g:/ShortCourse/CELFiles/CL2001011102AA.cel"
[3] "g:/ShortCourse/CELFiles/CL2001011104AA.cel"
```

Running Just RMA

```
> rmaData <- just.rma(filenames=all.cel.files,
+     phenoData=pd, description=miame)
[1] "Attempting to download hgu95acdf from
  http://www.bioconductor.org/packages/data/
  annotation/stable/bin/windows/contrib/2.1"
[1] "Download complete."
[1] "Installing hgu95acdf"
[1] "Installation complete"
Background correcting
Normalizing
Calculating Expression
```

Unexpected glitches

The two quantified sets have different numbers of genes:

```
> rmaData
Expression Set (exprSet) with
12626 genes
72 samples
 phenoData object with 13 variables and 72 cases

> dchip
Expression Set (exprSet) with
12625 genes
72 samples
 phenoData object with 13 variables and 72 cases
```

Figuring out which probe sets differ

```
> rma.ps <- dimnames(rmaData@exprs)[[1]]
> dchip.ps <- dimnames(dchip@exprs)[[1]]
> setdiff(rma.ps, dchip.ps)
 [1] "119_at"    "1215_at"   "1216_at"   "124_i_at"
 [5] "125_r_at"  "127_at"    "1301_s_at" "1302_s_at"
 [9] "132_at"    "1429_at"   "1502_s_at" "1829_at"
[13] "1864_at"   "1889_s_at" "1982_s_at" "36969_at"
[17] "383_at"    "397_at"    "412_s_at"  "426_at"
[21] "439_at"    "787_at"    "788_s_at"  "972_s_at"
[25] "985_s_at"  "997_at"
```

Selecting the common probes

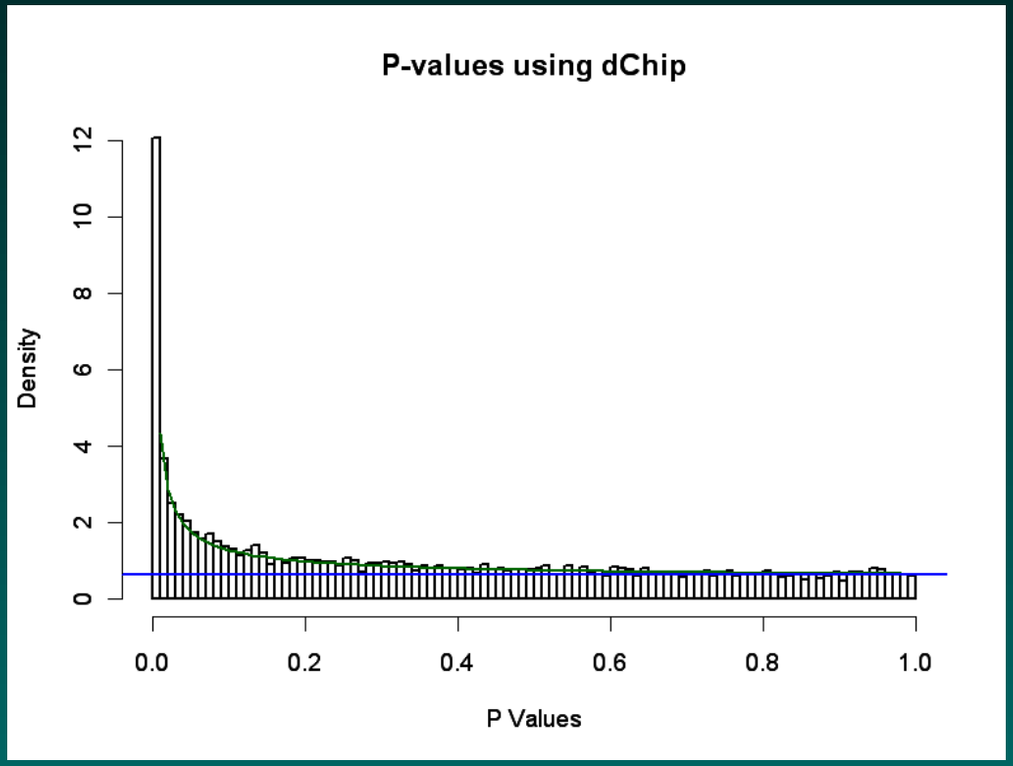
```
> rmaData <- rmaData[is.element(rma.ps, dchip.ps),]
> rmaData
Expression Set (exprSet) with
12600 genes
72 samples
 phenoData object with 13 variables and 72 cases
> rma.names <- dimnames(rmaData@exprs)[[1]]
> dchip@exprs <- dchip@exprs[rma.names,]
> dchip@exprs <- log(dchip@exprs, 2)
> dchip
Expression Set (exprSet) with
12600 genes
72 samples
 phenoData object with 13 variables and 72 cases
```

Loading the ClassComparison package

```
> require(ClassComparison)
Loading required package: ClassComparison
Loading required package: splines
Loading required package: oompaBase
Loading required package: PreProcess
Creating a new generic function for ’plot’ in ’PrePro
Creating a new generic function for ’print’ in ’PreP
Creating a new generic function for ’as.data.frame’ i
[1] TRUE
```

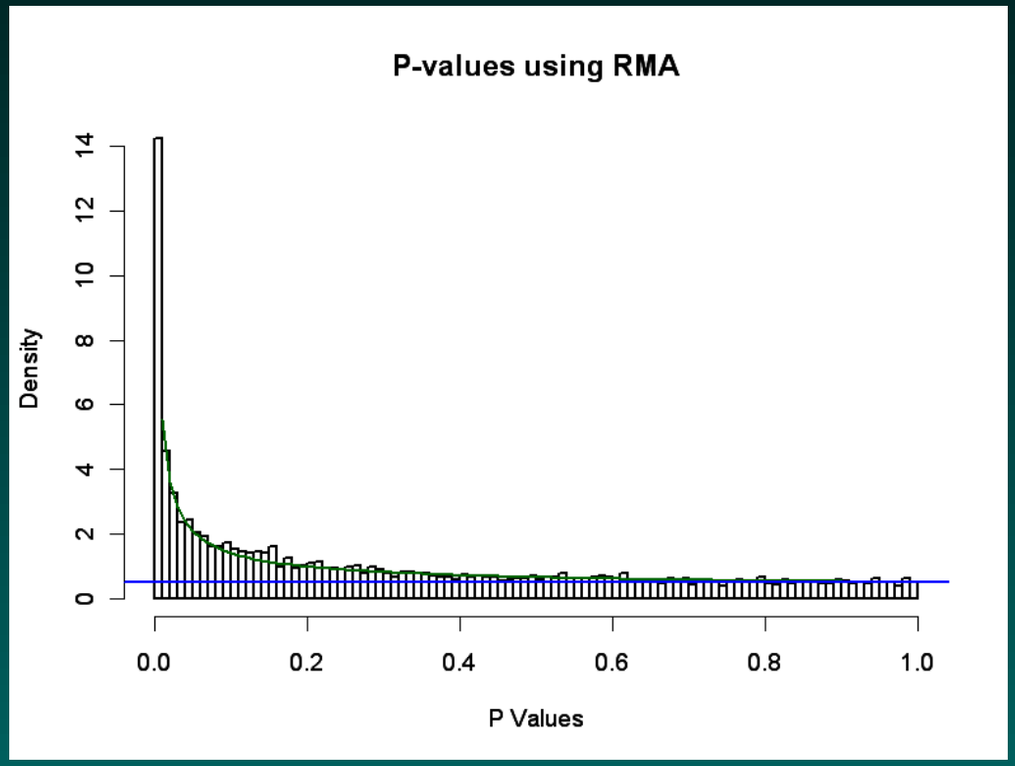
T-test, take one

```
> notAML <- pd@pData$type != 'AML'
> dchip.t <- MultiTtest(dchip[, notAML], 'ALLvMLL')
> dchip.b <- Bum(dchip.t@p.values)
> hist(dchip.b, main='P-values using dChip')
```



T-test, take two

```
> rma.t <- MultiTtest(rmaData[, notAML], 'ALLvMLL')
> rma.b <- Bum(rma.t@p.values)
> hist(rma.b, main='P-values using RMA')
```



DO the two methods agree?

```
> plot(rma.t@t.statistics, dchip.t@t.statistics)
> abline(0,1, col='blue')
```